FCAIH: Swarm Intelligence for Function Optimisation

Abdelhalem Ashraf	20210486	Level 3	Dept AI
Abdelrahman Ramadan	20210505	Level 3	Dept AI
Ali Adel	20210577	Level 3	Dept AI
Amr Khaled	20210640	Level 3	Dept AI
Eid Osama	20210648	Level 3	Dept AI
Mohey Eldeen	20210887	Level 3	Dept AI

May 10, 2024

1 Abstract

Function optimization is a critical task in various fields, from engineering to economics, where finding the optimal solution can significantly impact performance and efficiency. Swarm Intelligence algorithms, such as Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO), have emerged as effective tools for solving optimization problems inspired by the collective behavior of social organisms. This project aims to explore the application of PSO and GWO in function optimization tasks and compare their performance. Through a series of experiments using benchmark functions, we evaluate solution quality of both algorithms. The findings provide insights into the strengths and weaknesses of PSO and GWO, offering valuable guidance for future optimization tasks.

2 Introduction and Overview

Function optimization is a Free Optimization problem involves finding the input values that minimize or maximize a given objective function. This task is prevalent in various domains, including engineering design, machine learning, and financial modeling. Traditional optimization techniques often struggle with high-dimensional and non-linear problems, leading to the development of nature-inspired meta-heuristic algorithms.

Swarm Intelligence is a branch of artificial intelligence that draws inspiration from the collective behavior of natural swarms to solve optimization problems, such as ants, bees, and wolves. These algorithms simulate the collaborative behavior of individuals in a swarm to search for optimal solutions efficiently. Four prominent algorithms in Swarm Intelligence are Particle Swarm Optimization (PSO) ,Ant colony optimization (ACO), Grey Wolf Optimization (GWO) and Artificial Bee Colony (ABC).

2.1 Approaches

1. **Particle Swarm Optimization (PSO):** PSO was introduced by Kennedy and Eberhart in 1995[1], inspired by the social behavior of bird flocks and fish schools. In PSO, a population of particles explores the search space by adjusting their positions based on their own best-known position and the global best-known position found by the swarm.

2. **Grey Wolf Optimizer (GWO):** GWO, proposed by Seyedali Mirjalili and Seyed Mohammad Mirjalili and Andrew Lewis. in 2014[2], models the leadership hierarchy and hunting mechanism of grey wolves.

The algorithm iteratively Updates wolves positions based on alpha, beta, and delta wolves positions, mimicking the social interactions within a wolf pack to converge towards the optimal solution.

2.2 Proposed Solution

This project aims to investigate the effectiveness of PSO and GWO in function optimization tasks. By leveraging the collective intelligence of swarms, we aim to efficiently explore the search space and find near-optimal solutions and comparing their performance on benchmark functions, we seek to identify the strengths and weaknesses of each algorithm.

3 Literature Review

Function optimization is a fundamental problem in various domains, and over the years, numerous optimization techniques have been developed to tackle this challenge. Swarm Intelligence algorithms, inspired by the collective behavior of social organisms, have gained popularity due to their ability to efficiently explore complex search spaces. In this section, we review the relevant literature on Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), and their applications in function optimization.

1. **Particle Swarm Optimization (PSO):** PSO, introduced by Kennedy and Eberhart in 1995, is inspired by the social behavior of bird flocks and fish schools. In PSO, a population of particles represents potential solutions to the optimization problem. Each particle adjusts its position in the search space based on its own experience (personal best) and the collective experience of the swarm (global best). Through iterations, particles dynamically explore the search space, guided by their velocity and position updates.

   PSO has been widely applied to various optimization problems, including engineering design, neural network training, and robotics. Its simplicity, ease of implementation, and ability to handle continuous and discrete search spaces make it a popular choice for optimization tasks. However, PSO's performance may degrade in high-dimensional or multi-modal optimization problems, where it may struggle to maintain diversity and escape local optima.

2. **Grey Wolf Optimization (GWO):** GWO, proposed by Seyedali Mirjalili and Seyed Mohammad Mirjalili and Andrew Lewis. in 2014, is inspired by the social hierarchy and hunting behavior of grey wolves. The algorithm simulates the interactions between alpha, beta, and delta wolves within a wolf pack to update the positions of potential solutions. Alpha wolf represent the best solution found so far, while beta and delta wolves represent the second and third best solutions, the rest of wolves explore the search space under the guidance of alpha, beta and delta wolves.

   GWO has demonstrated competitive performance compared to other meta-heuristic algorithms in various optimization tasks. Its ability to balance exploration and exploitation, along with its simplicity and fewer control parameters, make it an attractive choice for function optimization. However, GWO's performance may vary depending on the problem characteristics, and it may struggle with complex landscapes or large-scale optimization problems.

In summary, both PSO and GWO offer effective solutions for function optimization tasks, each with its strengths and weaknesses.

4 Methodology

4.1 Mathematical Model and Implementation of PSO and GWO for Function Optimization

4.1.1 Particle Swarm Optimization (PSO)

In PSO, a population of particles explores the search space to find the optimal solution. Each particle adjusts its position based on its own best-known position ($p_{i,lb}$) and the global best-known position (p_{gb}) found by the swarm. The PSO algorithm can be summarized as follows:

1. Initialize the population of particles with random positions.
2. Evaluate the fitness of each particle based on the objective function.
3. Update the particle's velocity and position using the following equations:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$$
$$v_i^{(t+1)} = w \cdot v_i^{(t)} + c_1 \cdot r_1 \cdot (p_{i,lb}^{(t)} - x_i^{(t)}) + c_2 \cdot r_2 \cdot (p_{gb}^{(t)} - x_i^{(t)})$$

   where $v_i^{(t)}$ is the velocity of particle i at time t, $x_i^{(t)}$ is the position of particle i at time t, w is the inertia weight, c_1 and c_2 are acceleration coefficients (cognitive weight and social weight), and r_1 and r_2 are random numbers between 0 and 1.
4. Update the best-known positions of each particle and the global best-known position.

$$p_{i,lb}^{(t+1)} = \begin{cases} x_i^{(t+1)} & \textbf{if } f(x_i^{(t+1)}) < f(p_{i,lb}^{(t)}) \\ p_{i,lb}^{(t)} & \textbf{otherwise.} \end{cases}$$

5. Update the best-known positions of each particle and the global best-known position.

$$p_{gb}^{(t)} \in \{x_1^{(t)}, ..., x_N^{(t)}\} | f(p_{gb}^{(t)}) = \min\{f(x_1^{(t)}), ..., f(x_N^{(t)})\}$$

   where N is the numbers of particles in the swarm.
6. Repeat steps 2-5 until a stopping criterion is met (e.g., maximum number of iterations reached).

To implement PSO for function optimization, we followed the standard PSO algorithm. The pseudocode for PSO is as follows:

Algorithm 1 Particle Swarm Optimization (PSO)

```
Initialize swarm particles with random positions
Calculate the fitness of each particle
Initialize personal best positions and global best position
while stopping criterion is not met do
    for each particle in the swarm do
        Update velocity and position of each particle using personal and global best positions
        Calculate the fitness of each particle
        Update personal best position
        Update global best position
    end for
end while
```

4.1.2 Grey Wolf Optimization (GWO)

In GWO, the positions of alpha, beta, and delta wolves are iteratively updated to converge towards the optimal solution.

The GWO algorithm can be summarized as follows:

1. Initialize the population of wolves randomly within the search space.
2. Evaluate the fitness of each wolf based on the objective function.
3. Update the positions of the wolves based on alpha, beta, and delta positions using the following equations:

$$\overrightarrow{A} = 2a.\overrightarrow{r_1} - a$$
$$\overrightarrow{C} = 2.\overrightarrow{r_2}$$

$$\overrightarrow{D_\alpha} = |\overrightarrow{C}_1 \cdot \overrightarrow{X_\alpha} - \overrightarrow{X}|, \qquad \overrightarrow{D_\beta} = |\overrightarrow{C}_2 \cdot \overrightarrow{X_\beta} - \overrightarrow{X}|, \qquad \overrightarrow{D_\delta} = |\overrightarrow{C}_3 \cdot \overrightarrow{X_\delta} - \overrightarrow{X}|$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.(\overrightarrow{D_\alpha}), \qquad \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.(\overrightarrow{D_\beta}), \qquad \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.(\overrightarrow{D_\delta})$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3}$$

where t indicates the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors , $\overrightarrow{X}$ indicates the position vector of a grey wolf. components of a are linearly decreased from 2 to 0 over the course of iterations, and r_1 , r_2 are random vectors in [0,1].

4. Update the positions of alpha, beta, and delta wolves.
5. Repeat steps 2-4 until a stopping criterion is met.

To implement GWO for function optimization, we followed the standard GWO algorithm. The pseudocode for GWO is as follows:

Algorithm 2 Grey Wolf Optimization (GWO)

```
Initialize the grey wolf population X_i (i = 1, 2, ..., N)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α=the best search agent
X_β=the second best search agent
X_δ=the third best search agent
while stopping criterion is not met do
    for each grey wolf do
        Update positions based on alpha, beta, and delta positions
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t = t + 1
end while
return X_α
```

4.2 Benchmark Functions

For evaluation, we used several benchmark functions commonly used in optimization literature. These functions represent different characteristics such as multimodality, nonlinearity, and high-dimensionality. Some of the benchmark functions used in our study include:

- Ackley Function

$$f(x) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e \tag{1}$$

- Schwefel Function

$$f(x) = 418.9829 \cdot d - \sum_{i=1}^{d} x_i \cdot \sin(\sqrt{|x_i|}) \tag{2}$$

- Bohachevsky Functions

$$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \tag{3}$$

These benchmark functions provide a standardized way to assess the performance of optimization algorithms across various problem types.

4.3 Parameter Settings for PSO and GWO

The performance of PSO and GWO can be sensitive to parameter settings such as population size, maximum iterations, inertia weight, and acceleration coefficients. For our experiments, we used the following parameter settings:

- PSO:
  - Population size: 60
  - Maximum iterations: 50
  - Inertia weight: 0.5 or linearly decreased from 0.9 to 0.1 over the course of iterations
  - Acceleration coefficients: c1 , c2 $\in \{0.5, 2.5, 5\}$
- GWO:
  - Population size: 60
  - Maximum iterations: 50

These parameter settings were chosen based on prior literature and empirical testing to ensure a fair comparison between PSO and GWO.

5 Experiments

To evaluate the performance of PSO and GWO, we conducted experiments on standard benchmark functions. The experiments focused on:

1. Calculating the convergence speed of PSO and GWO.

   Let A denote the set of algorithms considered in the experiment, and let F denote the set of benchmark functions (Ackley, Schwefel, Bohachevsky) under investigation.

   For each algorithm $a \in A$ and each function $f \in F$, the algorithm was run 30 times from a fixed random seed = 20. Let c_{af}^i represent the list of best fitness for each iteration of the ith run of algorithm a on function f, where $i = 1, 2, \ldots, 30$.

   The average convergence of algorithm a on function f is calculated as:

   $$\text{AvgConv}_{af} = \frac{1}{30} \sum_{i=1}^{30} c_{af}^i$$

2. Calculating the accuracy of PSO and GWO.

   Furthermore, let $\min_{af}^i$ represent the ith minimum obtained by algorithm a on function f, where $i = 1, 2, \ldots, 30$. The mean and standard deviation of the minima obtained by algorithm a on function f are calculated as:

   Mean:

   $$\text{MeanMin}_{af} = \frac{1}{30} \sum_{i=1}^{30} \min_{af}^i$$

   Standard Deviation:

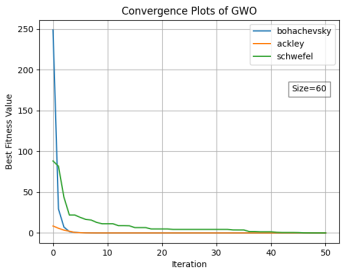
   $$\text{StdDevMin}_{af} = \sqrt{\frac{1}{30} \sum_{i=1}^{30} \left(\min_{af}^i - \text{MeanMin}_{af}\right)^2}$$

6 Results & Comparison

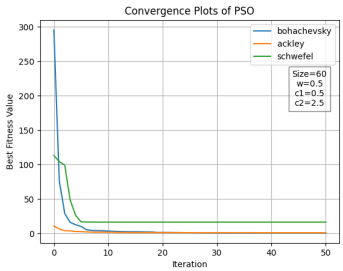
We compared the performance of PSO and GWO in terms of:

1. Convergence Speed: How quickly the algorithms converge to the optimal solution.
2. Accuracy: The quality of the solutions obtained by PSO and GWO.

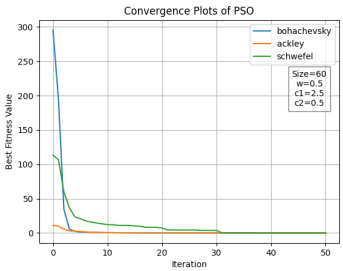
6.1 Parameter Settings and Results



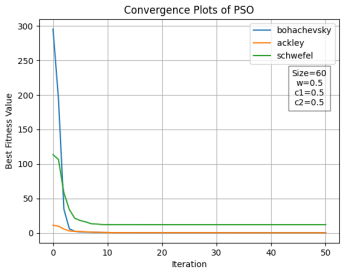
(a) Plot of GWO



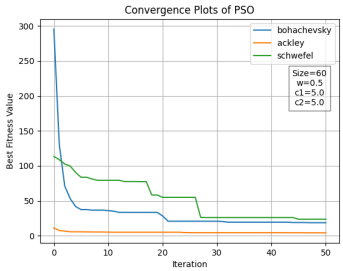
(b) PSO $w = 0.5$, $c_1 < c_2$



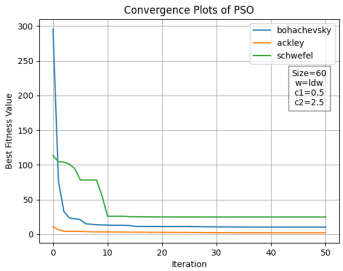
(c) PSO $w = 0.5$, $c_1 > c_2$



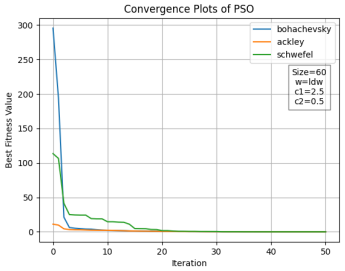
(d) PSO $w = 0.5$, low c_1, c_2



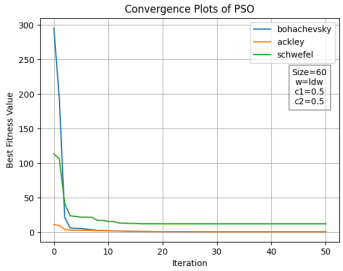
(e) PSO $w = 0.5$, large c_1, c_2



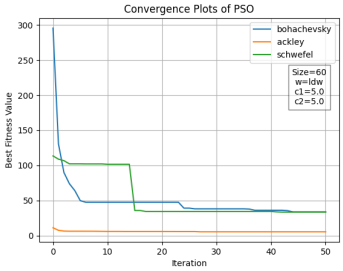
(f) PSO w is LDW, $c_1 < c_2$



(g) PSO w is LDW, $c_1 > c_2$



(h) PSO w is LDW, low c_1, c_2



(i) PSO w is LDW, large c_1, c_2

1. Convergence Speed:

   After doing multiple iterations on the algorithms for optimization, we realized that the GWO convergence was faster than PSO, no matter how many iterations we tested our solution.

   The following Figure 1a shows the GWO convergence and Figure 1c shows the PSO convergence.

2. Accuracy:

   After doing multiple iterations on the algorithms for optimization, we calculated the mean and standard deviation of the minima obtained by the algorithms.

   The following Table 1 shows the results of benchmark functions.

w	c1	c2	Bohachevsky Mean	Bohachevsky Std	Ackley Mean	Ackley Std	Schwefel Mean
0.5	2.5	0.5	0.0000043	0.0000155	0.0000102	0.0000150	0.0010305
0.5	0.5	2.5	0.9499576	2.2522140	1.2721871	1.4263876	16.6817742
0.5	0.5	0.5	0.1891661	0.3401589	0.1348055	0.2673065	11.9433357
0.5	5.0	5.0	18.6243934	15.6108100	4.1949378	1.8374616	23.6264266
LDW	5.0	5.0	33.6970080	28.7025015	5.3069644	2.2040599	33.3660169
LDW	2.5	0.5	0.0261773	0.1312340	0.0002694	0.0003302	0.0000468
LDW	0.5	2.5	10.3400970	15.3709389	2.1554190	1.7298618	24.9039954
LDW	0.5	0.5	0.6258210	0.8573283	0.6681844	1.0526012	12.1970534

Table 1: Function Results

7 Analysis, Discussion, and Future Work

7.1 Algorithm Efficiency

- We analyzed the efficiency of PSO and GWO in terms of computational complexity and convergence characteristics. Both PSO and GWO have their strengths and weaknesses in different scenarios.
- PSO is known for its simplicity and ease of implementation. It requires few parameters to tune. However, PSO may struggle with premature convergence and can get stuck in local optima if not properly configured.
- On the other hand, GWO is inspired by the hunting behavior of grey wolves and exhibits powerful exploration capabilities. It tends to have a faster convergence rate, especially in multimodal optimization problems.
- We discussed how variations in algorithm parameters, such as the number of particles or wolves, inertia weight, and acceleration coefficients, can significantly affect the performance of both PSO and GWO. Fine-tuning these parameters is crucial to achieving optimal results.

7.2 Advantages / Disadvantages

We discussed the advantages and disadvantages of PSO and GWO compared to other optimization algorithms:

- **Advantages of PSO:**
  - Its simplicity and intuitive nature make it easy to understand and implement.
  - PSO is robust to noisy environments, making it suitable for optimization problems with uncertain or noisy objective functions.
  - It can handle both continuous and discrete optimization problems effectively.
- **Disadvantages of PSO:**
  - PSO may struggle with high-dimensional problems, where the search space is vast and complex.
  - It can be sensitive to parameter settings, requiring careful tuning for optimal performance.
- **Advantages of GWO:**
  - GWO offers powerful exploration capabilities, thanks to its nature-inspired search mechanism based on grey wolf behavior.
  - It is particularly effective in solving multimodal optimization problems, where multiple local optima exist, and can quickly converge to global optima.
- **Disadvantages of GWO:**
  - It may struggle with optimization problems that have discontinuous or irregular search spaces, as the search process relies on smooth transitions between solutions inspired by grey wolf behavior.

- **In summary,** the choice between PSO and GWO depends on the specific characteristics of the optimization problem at hand, including its dimensionality, multimodality, and noise level. PSO may be preferred for its simplicity and robustness to noise, while GWO may excel in scenarios requiring powerful exploration capabilities and fast convergence to global optima.

7.3 Future Work

Future research could focus on

- Experiment with additional optimization techniques.
- Enhancing the performance and robustness of PSO and GWO algorithms in various ways:

7.3.1 Parallelization

Investigating parallel and distributed versions of PSO and GWO algorithms could enable efficient utilization of computational resources and scalability to handle large-scale optimization problems. Parallelization techniques, such as parallel particle or wolf evaluations, could significantly reduce the optimization time and enhance the scalability of these algorithms.

7.3.2 Adaptation to Dynamic Environments

Developing adaptive PSO and GWO variants capable of dynamically adjusting their search behavior in response to changes in the optimization landscape or problem constraints could improve their applicability to dynamic optimization problems. Adaptive algorithms that can efficiently track and adapt to changes in the environment could lead to more robust and versatile optimization solutions.

7.3.3 Benchmarking and Comparison

Conducting comprehensive benchmarking studies and comparative analyses of PSO and GWO algorithms against state-of-the-art optimization techniques across a wide range of benchmark functions and real-world applications could provide valuable insights into their performance characteristics, strengths, and weaknesses. Benchmarking studies could help identify the most suitable algorithm for different types of optimization problems and guide the development of future optimization algorithms.

8 Conclusion

In this study, we investigated the effectiveness of Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) in function optimization tasks. Through a series of experiments on standard benchmark functions, we compared the convergence speed and solution quality of both algorithms.

Our findings suggest the following:

- GWO generally exhibits faster convergence compared to PSO across various benchmark functions.
- GWO tends to provide more accurate solutions, particularly in multimodal optimization problems.
- PSO demonstrates advantages in terms of simplicity, ease of implementation, and robustness to noise.

The choice between PSO and GWO depends on the specific characteristics of the optimization problem, including its dimensionality, multimodality, and noise level. While PSO may be preferred for its simplicity and robustness in certain scenarios, GWO excels in tasks requiring powerful exploration capabilities and fast convergence to global optima.

Future research could explore several directions to enhance the performance and applicability of PSO and GWO algorithms:

- Investigating parallelization techniques for efficient resource utilization.
- Developing adaptive variants to handle dynamic optimization environments.

- Conducting comprehensive benchmarking studies against state-of-the-art optimization techniques.

By addressing these challenges and further refining PSO and GWO algorithms, we can advance the field of swarm intelligence and contribute to the development of more effective optimization tools for solving complex real-world problems.

The complete source code details and history of development is detailed on the git repository:
https://github.com/Abdelrhman-Ramadan/Swarm-Intelligence-for-Function-Optimisation

References

[1] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995. doi: 10.1109/ICNN.1995.488968.

[2] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014. ISSN 0965-9978. doi: https://doi.org/10.1016/j.advengsoft.2013.12.007. URL https://www.sciencedirect.com/science/article/pii/S0965997813001853.

[3] Haiping Ma, Sengang Ye, Dan Simon, and Minrui Fei. Conceptual and numerical comparisons of swarm intelligence optimization algorithms. *Soft Computing*, 21, 06 2017. doi: 10.1007/s00500-015-1993-x.

[4] Jun Tang, Gang Liu, and Qingtao Pan. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA Journal of Automatica Sinica*, 8(10):1627–1643, 2021. doi: 10.1109/JAS.2021.1004129.

[5] Yan He, Wei Jin Ma, and Ji Ping Zhang. The parameters selection of pso algorithm influencing on performance of fault diagnosis. In *MATEC Web of conferences*, volume 63, page 02019. EDP Sciences, 2016.