Adaptive Hybridization of Particle Swarm Optimization and Grey Wolf Optimization

Aakash Sharma
Department of MCA
Ajay Kumar Garg Engineering College
Ghaziabad, India
aakashsharma.as34@gmail.com

Aryan Garg
Department of MCA
Ajay Kumar Garg Engineering College
Ghaziabad, India
aryangarg104@gmail.com

Saroj Bala
Department of MCA
Ajay Kumar Garg Engineering College
Ghaziabad, India
saroj.chhokar@gmail.com

***Abstract*—An Adaptive Hybridization of Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) (AHPSOGWO), is presented. This hybridization leverages PSO's exploration ability and GWO's exploitation capacity through adaptive mechanisms to address limitations such as premature convergence and local optima stagnation. Even though the original hybrid approach gives better performance, it is appropriate only for problems with a continuous search space. However, feature selection and premature convergence speed and easy to trap into local optimum on some problems and therefore, Adaptive Hybridization of PSO and GWO is proposed and implemented. 7 standard benchmark functions were utilized to validate the algorithm's efficacy, demonstrating superior convergence speed, stability, and potential to find the global optimum compared to standalone PSO and GWO. Pseudo-code, mathematical models, and experimental evaluations are discussed in detail.**

***Keywords*—Particle Swarm Optimization, Grey Wolf Optimization, Adaptive Hybridization, Swarm Intelligence, Optimization, Benchmark Functions.**

I. INTRODUCTION

Swarms basically refer to how many simple organisms, such as ants, bees, birds, and fish, work together to solve complex problems. In nature, the concept of swarm intelligence does not involve its application in a technological or computer science scenario. Upon their adoption in the processes, the achievements of the engineers and the scientists have managed to answer different sorts of questions, including issues in the robotic domain, data analysis and optimization. An example for the potential applications of swarm intelligence-related algorithms could include the coordination of big autonomous-robot networks or be used for the resolution of mathematical challenging problems.

The application of swarm intelligence helps achieve decentralized, not centralized, solutions in problem-solving, with the collective power of simple action rather than emulating how insects or animal's work.

Particle Swarm Optimization [5] is a population-based algorithm that seeks for the best answer by imitating the social behaviour of fish schooling or flocking birds. Kennedy and Eberhart realized that there are some animal groups, such as flocks of birds or schools of fish, which benefit from information sharing. For example, if a bird finds its food, then indirectly the whole flock benefits since other birds will learn about this discovery and modulate their search efforts correspondingly. In PSO, simple computational entities known as particles are made to search the space of an optimization problem. Each particle position is a candidate solution and changes its velocity according to definite rules that imitate how a flock of birds will move together. The particles are always updating their positions as they try to improve solutions based on both their best discoveries and the best discoveries of other particles in the swarm. Particle Swarm Optimization [5] (PSO) is a metaheuristic optimization algorithm that was developed to optimize complex nonlinear functions.

Grey Wolf Optimization [1] is a metaheuristic algorithm that is inspired from the social living behaviour and hunting pattern of wolves living in natural environments. A pack has a status called alpha with four distinct rank categorizations such as alpha, beta, delta, and omega. The alphas are acting as leaders and receive full support from the beta type. The delta wolf is the one, which has a role in maintaining order, and the omega wolf is the most subordinate one. In terms of hunting, the pack wolves are all in sight seeking and wooing and finally encircling their quarry. Out of this a GWO algorithm was developed to mean that the various 'wolves' have designated roles when identifying the best solution to a problem, which in real life is the hunting roles. As such, this algorithm known as GWO is highly desirable in the accomplishment of tasks related to optimization in many of the engineering, data science, and machine learning fields studied in this paper.

Stand-alone PSO and GWO are limited in their benefits by an exceedingly slow convergence [9] and poor exploration or exploitation abilities. To repair this, this study proposes the Adaptive Hybridization of PSO and GWO algorithm, AHPSOGWO, which adaptively balances exploration and exploitation for enhanced optimization efficiency. In this review, AHPSOGWO along with its implementation is descried in details, which is then compared with stand-alone PSO and GWO providing an efficient, at indicative balance within the terms of optimization process.

In this study, we propose a newly hybrid algorithm called AHPSOGWO, which incorporates Particle Swarm Optimization [5] (PSO) and Grey Wolf Optimizer (GWO). We test the performance of this hybrid variant against standard PSO and GWO using seven different unimodal, multimodal, and fixed-dimension multimodal benchmark functions. The paper is organized as follows: Sections I contains the introduction, Section II describes the related work and Section III provide an overview of the PSO and GWO algorithms, respectively and discusses the implementation, mathematical model and pseudocode of AHPSOGWO. Section IV presents the benchmark functions used for testing, while Section V and Section VI comprises the experimental results and discussion. Finally, the conclusion and future work is given in Section VII.

II. RELATED WORK

TABLE I: Key Findings of the Research Papers

S. No	Name of the Research Paper	Results of the Research Paper
1.	Mirijalili, S., Mirijalili, S. M., & Lewis, A. (2014). "Grey Wolf Optimizer." Advances in Engineering Software, 69, 46-61	Grey Wolf Optimizer as a new metaheuristic inspired by the social hierarchy and hunting strategies of grey wolves. In this paper, the authors demonstrated how GWO mimics the roles of alpha, beta, delta, and omega wolves [1] to find an optimal solution. The authors conducted some tests on different benchmark functions that demonstrated competitive performance of GWO with established algorithms.
2.	E. Emary and Hossam M. Zawbaa & Aboul Ella Hassanien (2016). "Binary Grey Wolf Optimization Approaches for Feature Selection." Neurocomputing, 172, 371-381.	The GREY WOLF OPTIMIZATION (GWO) was made binary to improve selection of features used in classification tasks by Emary, Zawbaa, and Hassanien in 2016. The binary approaches have been applied in two ways. The first one incorporates stochastic crossover among the best solutions, and the second one is performed by using the Sigmoid function and then thresholding to determine the binary position. These methods have been tested on 18 datasets from the UCI repository in comparison with other algorithms like the Particle Swarm Optimization [5] and Genetic Algorithms. The results confirmed that the binary approach can exploit the feature space efficiently, always leading to the discovery of optimal feature subsets despite initial conditions and stochastic variations.
3.	N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization	In this paper the authors present a promising improved version of the Grey Wolf Optimization [1] (GWO) technique involving application to complex engineering optimization problems. Exploring the modified GWO model, it is also observed that it ensures an equilibrium in between exploration and exploitation that facilitates more rapid and accurate convergence. This improvement enables one to solve high dimensional problems without much danger of stagnation in local optimal solutions. However, the high computation requirement may often be a cost which would limit its ability to perform under real-time situations, and oriented features of the model towards a class of problems may keep the general applicability limited.
4.	Kennedy, J., & Eberhart, R. (1995) - "Particle Swarm Optimization"	It was the first work from which PSO emerged as an optimization algorithm inspired from the natural behaviour of flocks of birds and swarms of fish. In it, authors demonstrated how a single "particle" could find an optimum solution by informing other particles in complex search spaces through cooperation. This paper was able to show that PSO is indeed effective in optimizing nonlinear functions and it has since been used in many application areas to further extend it.
5.	Clerc, M., & Kennedy, J. (2002) - "The Particle Swarm – Explosion, Stability, and Convergence in a Multi-dimensional Complex Space"	In this paper, Clerc and Kennedy proposed a modification called the constriction factor, which further improved the stability of the motion of particles in the search space. The constriction factor made sure that particles did not diverge too rapidly. PSO became a much more stable optimization algorithm with this modification and thus found its place as a strong tool for engineering and scientific applications.
6.	Eberhart RC, Shi Y (2000), July Comparing inertia weights and constriction factors in particle	Eberhart and Shi (2000) focused their study on the inertia weights and constriction factors as they related to the performance of Particle Swarm Optimization [5] (PSO). The use of inertia

	swarm optimization. In Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512) (Vol. 1, pp. 84–88)	weights, has been considered for balancing the two views of exploration and exploitation, enabling a proper exploration motion of the particles in the solution space. Further, constriction factors proved to have direct influence as far as convergence is concerned because it resulted in controlling the particle velocity and made this convergence more stable and reliable. It could, however, not accommodate a wide range of variables because it concentrated on few parameter values that would make little or no outside meaning out of more complex or dynamic optimization, where the parameter landscapes keep changing.
7.	Title: Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance Author: Singh, Narinder Journal: Journal of Applied Mathematics Year: 2017	HPSOGWO is hybrid nature-inspired optimization algorithm and it is introduced in this paper. HPSOGWO is a combination of Particle Swarm Optimization [5] (PSO) and Grey Wolf Optimizer [1] (GWO). The motivation of algorithm is to strengthen exploitation capabilities of PSO with exploration potentialities of GWO with using advantages of both methods. To evaluate HPSOGWO performance and solution quality, then some unimodal, multimodal, and fixed-dimensional multimodal test functions have been applied. The hybrid algorithm outperforms the stand-alone PSO and GWO algorithms in terms of solution accuracy and stability, the convergence, and even the ability to locate the global optimum by various numerical and statistical analyses.
8.	Economic Load Dispatch Using Hybrid GWO-PSO Algorithm. *International Journal of Research in Advent Technology*	The study, which focuses on the hybridization of two swarm intelligence techniques for the Economic Load Dispatch (ELD) problem in power systems. This hybrid approach has been successfully used for generating cost savings while satisfying generation constraints to achieve minimum fuel cost. The algorithm ensures faster convergence and offers better accuracy as compared to traditional methods such as lambda iteration by having the PSO advantage as a local search with that of GWO's global search capability. The results indicated evidence of effectiveness and the possibility for this method to be applied for an optimization of existing real-world power system operations toward better economic and reliability outcomes.
9.	S. Yu, Z. Wu, H. Wang, Z. Chen, and H. Zhong, “A hybrid particle swarm optimization algorithm based on space transformation search and a modified velocity model”	S. Yu et al. have recommended a hybrid PSO that combines space transformation search and a modified velocity model to boost overall optimization performance. The space transformation method provides room for exploration because of the expansion of exploration space; the modified velocity model enhances convergence and exploitation. Test results have indicated its advantages over traditional PSO in terms of giving better solution quality, with faster convergence and greater stability. This hybrid approach resolves issues like premature convergence [9] in traditional PSO by balancing exploration and exploitation.
10.	An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning.	This research integrates Particle Swarm Optimization [5] and Grey Wolf Optimization [1] with chaos theory inside a hybrid algorithm. The approach addresses optimization issues related to local optima and untimely convergence, well combining global search with local refinement. The algorithm does perform well in terms of robot path planning since it has made great contributions toward making sure that distances are shorter. Paths are much smoother and safer and more effective than those planned classically by PSO, GWO, or hybrids themselves. Moreover, chaos theory inclusion promotes diversity in solutions while the adaptiveness offers flexibility in dealing with difficult environments.

III. METHODOLOGY

A. PARTICLE SWARM OPTIMIZATION ALGORITHM

A heuristic optimisation technique called Particle Swarm Optimisation [5] (PSO) is inspired by the cooperative behaviour of fish schools and flocks of birds. The algorithm searches within a search space using a population of particles to obtain the best answer to an objective function.

The following steps are required for PSO implementation:

1. Set the particle population's initial placements and speeds at random locations throughout the search space.

2. Assess each particle's fitness in light of the objective function.

3. Based on each particle's current position and fitness, update its personal best position and fitness.

4. Based on the individual best positions and fitness of each particle, update the population's best global position and fitness.

5. Each particle's velocity is updated in accordance with its present velocity, unique best position, and overall best position.

6. Based on the position and velocity of each particle, adjust their positions.

7. The stopping requirement, like reaching to maximum number of iterations or a desirable fitness value, must be satisfied by repeating steps 2 through 6 until it is.

8. As the best answer to the goal function, return the overall best position.

PSO can be implemented with a programming language such as Python or Matlab. By setting parameters such as the number of particles, inertia weight, and acceleration coefficients, the algorithm can be tuned according to the needs of the application. The modifications and combinations with such algorithms have demonstrated better results. For instance, PSO is combined with Genetic Algorithms, Differential Evolution, and so on, while making use of the time-variant inertia weight.

B. GREY WOLF OPTIMIZATION ALGORITHM

Another heuristic optimisation technique that draws inspiration from the social behaviour of grey wolves is called Grey Wolf Optimisation [1] (GWO). To resolve optimisation issues, the algorithm imitates the hunting and pack leadership behaviour of grey wolves.

The following actions are needed for GWO implementation:

1. Set up the wolf population at random starting locations inside the search area.

2. Based on the objective function, determine each wolf's level of fitness.

3. The population's top three fittest wolves should be given the titles of alpha, beta, and delta wolves [1].

4. Based on each wolf's present location as well as the locations of the alpha, beta, and delta wolves [1], each wolf's position should be updated.

5. Adapt each wolf's fitness based on the objective function.

6. Up until the halting requirement is reached, repeat steps 3 through 5.

7. As the best answer to the objective function, give the position of the alpha wolf [1].

It can be practically implemented through existing programming language like python or Matlab with adjustment of parameter. It is easy to adjust the convergence curve and the number of wolves to meet certain requirements. The convergence of GWO can be increased through hybridization or adjustment of the algorithm to other algorithms selectively such as adaptive or chaotic search processes or with combining GWO with other metaheuristic methods such as Particle Swarm Optimisation and Ant Colony Optimisation.

*C. **IMPLEMENTATION OF PROPOSED ADAPTIVE HYBRIDIZATION OF PARTICLE SWARM OPTIMIZATION AND GREY WOLF OPTIMIZATION***

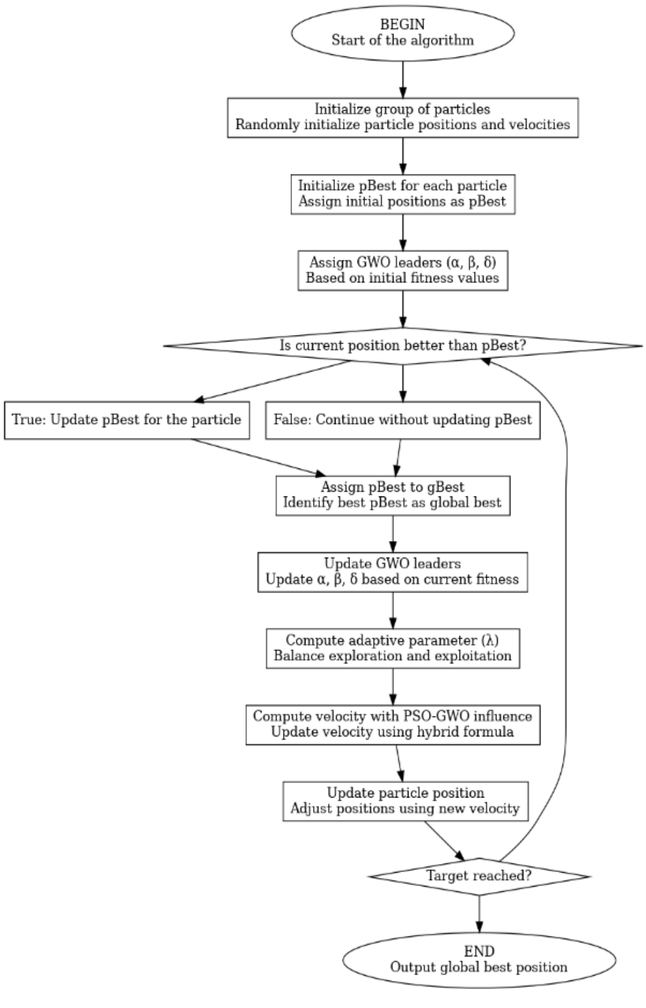


Fig. 1. Flowchart of Proposed Adaptive Hybrid Approach

The proposed Adaptive Hybrid PSO-GWO (AHPSOGWO) combines the swarm intelligence from PSO and the leadership mechanism from GWO [2]. This approach, being a blend, is capable to control the influence of all the above said components, more effectively in the interest of making efficient and accurate convergence in the least possible time, but always with exploration at its initial stages and exploitation towards the later period.

Below are the necessary steps to implement AHPSOGWO:

1. Initialize Particle Positions and Velocities:

- Start by randomly allocating initial positions and velocities to each particle within the search space limits [5].
- Choose the top three particles based on their fitness values to act as the initial GWO leaders: α (best fitness), β (second-best fitness), and δ (third-best fitness) [1].

2. Evaluate Fitness Values:

- Evaluate the fitness of each particle based on the given objective function [1], [5].

3. Update Individual and Global Best Positions:

- For each particle, check if its current fitness value is better than its personal best fitness. If so, update the particle's personal best position and fitness [5].
- Determine the best-performing particle in the population and update this as the global best [5].

4. Assign and Update GWO Leaders:

- Based on the fitness values calculated, assign the top three particles to the roles of α, β and δ. These three leaders will guide the rest of the population using GWO's influence [1].

5. Set the Adaptive Parameter:

- Define the adaptive parameter, λ, which gradually decreases from an initial value of 1 to around 0.5 over time. This parameter enables a shift from exploration to exploitation as the algorithm progresses [14].

6. Update Particle Velocities with the Hybrid Velocity Formula:

- Adjust each particle's velocity by combining PSO's individual and social components with GWO's leader influence [9]. The hybrid velocity formula is as follows:

$$\boldsymbol{v_i(t+1) = \omega \cdot v_i(t) + \lambda\left(c_1 \cdot r_1 \cdot \left(p_i^{best} - x_i(t)\right) + c_2 \cdot r_2 \cdot \left(X_{\alpha\beta\delta} - x_i(t)\right)\right)}$$

- Explanation of terms:
- ω : The inertia weight factor, which helps control how much a particle's current velocity influences its new velocity [14].
- c_1 *and* c_2: Coefficients for the cognitive and social terms, which pull particles toward their personal and global bests, respectively [5].
- λ: The adaptive parameter, which initially emphasizes exploration (values closer to 1) and gradually allows more local searching (values closer to 0.5).
- $X_{\alpha\beta\delta}$: A weighted combination of the three GWO leaders, calculated as:

$$\boldsymbol{X_{\alpha\beta\delta} = \frac{(\omega_\alpha \cdot X_\alpha + \omega_\beta \cdot X_\beta + \omega_\delta \cdot X_\delta)}{\omega_\alpha + \omega_\beta + \omega_\delta}}$$

- Weights ω_j for each leader ($j \in \{\alpha, \beta, \delta\}$) are set based on fitness, where $\omega_j = \frac{1}{(1+f_j)}$. Leaders with better fitness have a larger influence.

7. Update Positions Using the Hybrid Position Formula**:**

- After updating the velocity, adjust each particle's position using the following equation:

$$\boldsymbol{x_i(t+1) = x_i(t) + v_i(t+1) + (1 - \lambda) \cdot \Delta X_{GWO}}$$

- Explanation of terms:
- $\boldsymbol{\Delta X_{GWO} = \frac{(X_\alpha + X_\beta + X_\delta)}{3} - x_i(t)}$: This term represents the average guidance effect from GWO leaders. As λ decreases, GWO's influence grows, pulling particles toward promising regions in the search space.

8. Check Convergence Criteria**:**

- Repeat steps 2 through 7 until a predefined stopping condition is met, like maximum number of iterations or a satisfactory fitness value.

9. Return the Best Solution **: -** Output the global best position as the optimal solution found by the algorithm.

Pseudocode for the Proposed AHPSOGWO

Start by initializing A, c_1, c_2, ω and λ.

Randomly set initial positions and velocities for *n* particles within the defined search space.

Use the fitness function to determine the best three solutions, designated as α, β, and δ.

Calculate the fitness value for each particle based on the objective function.

While the stopping condition (*t<MaxIter*) is not met:

- For each particle:
    - Update its velocity using the combined formula:

$$v_i(t+1) = w \cdot v_i(t) + \lambda\left(c_1 \cdot r_1 \cdot \left(p_i^{best} - x_i(t)\right) + c_2 \cdot r_2 \cdot \left(X_{\alpha\beta\delta} - x_i(t)\right)\right)$$

- Modify its position using the formula: $x_i(t+1) = x_i(t) + v_i(t+1) + (1-\lambda) \cdot \Delta X_{GWO}$
- Ensure that the particle's position and velocity remain within the allowed boundaries.

- Update the leaders α, β, and δ based on the current fitness evaluations.
- Adjust λ adaptively over time:

$$\lambda = \lambda - \left(\frac{1}{MaxIter}\right) \cdot 0.5$$

- Recompute the fitness of all particles to assess their performance.
- Record the global best position and its corresponding fitness value.
- Increment t by 1.

End While

Output the global best position as the solution.

IV. TESTING FUNCTIONS

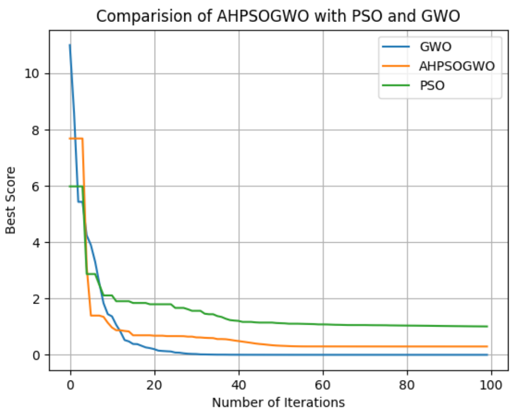
The performance of Particle Swarm Optimisation [5] (PSO), Grey Wolf Optimisation [1] (GWO) and Adaptive Hybridization of PSO and GWO is being tested against benchmark functions. Mathematics benchmark functions that are frequently used to test the effectiveness of optimisation algorithms were used in this study. The capacity of these functions to represent various optimisation issues, including single- or multi-modal problems, continuous or discrete problems, and problems with or without constraints, is the basis for their selection. The tests will be carried out on benchmark functions such the Sphere, Ackley, Penalized, Schwefel, Michalewicz, RosenBrock and Levy functions [9]. The algorithms will be devised and implemented using the Python programming language. In the tests, the algorithms will be applied to these benchmark functions and their performance will be evaluated using a variety of measures, including convergence rate, fitness value, and execution time. or number of iterations. The results of the empirical investigation will shed light on how these hybridizations and algorithms work and can be applied to real-world issues. The study will pinpoint the best algorithm or hybridization strategy for tackling optimisation challenges, assisting decision-makers in choosing the best algorithm for their own requirements.

BENCHMARK FUNCTIONS

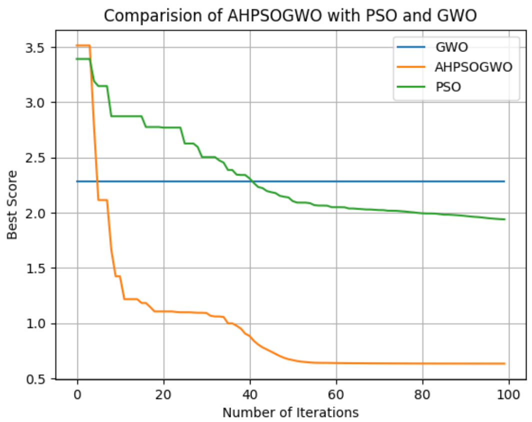
Benchmark Function Name	Formula
Sphere Function (F1)	$f(x) = \sum_{i=1}^{n} x_i^2$
Ackley Function (F2)	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$
Penalized Function (F3)	$f(x) = 0.1\left[\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2(1+\sin^2(3\pi x_{i+1})) + (x_n-1)^2(1+\sin^2(2\pi x_n))\right] + \sum_{i=1}^{n} U(x_i, 5, 100, 4)$
Schwefel Function (F4)	(f(x) = 418.9829 n - \sum_{i=1}^n x_i \sin(\sqrt{
Michalewicz Function (F5)	$f(x) = -\sum_{i=1}^{n}\sin(x_i)\cdot\sin^m\left(\frac{ix_i^2}{\pi}\right)$ (where m is a steepness parameter)
Rosenbrock Function (F6)	$f(x) = \sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2 + (x_i-1)^2\right]$
Levy Function (F7)	$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i-1)^2[1+10\sin^2(\pi w_{i+1})] + (w_n-1)^2[1+\sin^2(2\pi w_n)]$, where $w_i = 1 + \frac{x_i - 1}{4}$

Fig. 2. Benchmark Functions

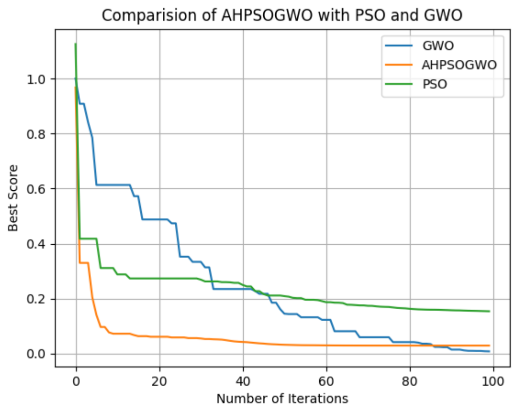
V. EXPERIMENTAL RESULTS AND ANALYSIS



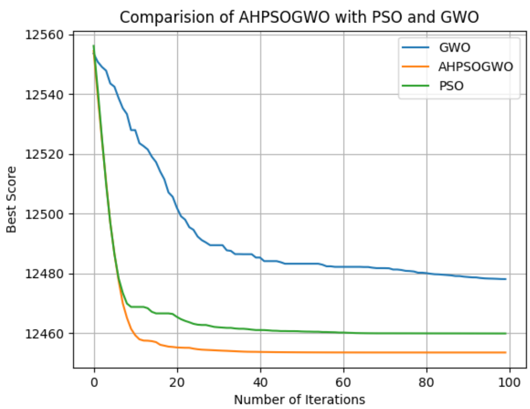
Sphere Function i.e. F1 is finely working and signifies that GWO outperforms both AHPSOGWO and PSO by reaching the global minimum more efficiently and with greater accuracy.



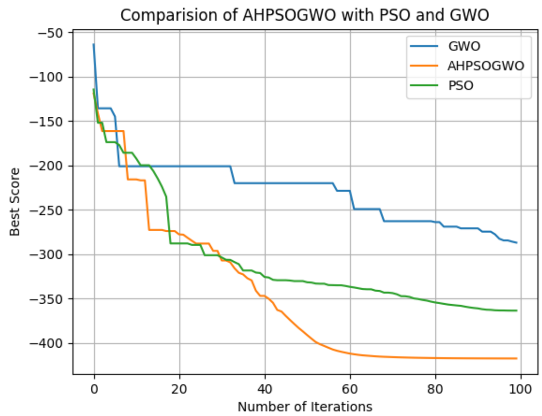
Ackley Function i.e. F2 is finely working and depicts that AHPSOGWO surpass both PSO and GWO by achieving the global minimum more efficiently and with better accuracy.



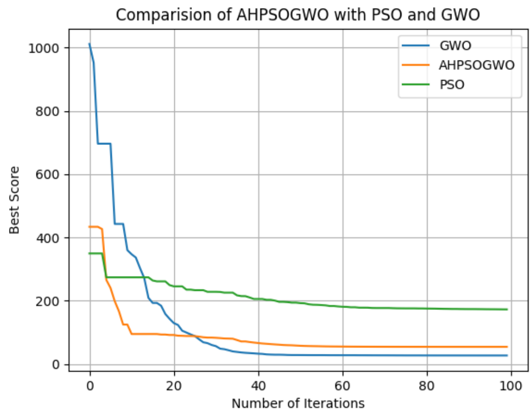
Penalized Function i.e. F3 is working correctly and indicates that AHPSOGWO outperforms both PSO and GWO by achieving the global minimum more efficiently and with greater accuracy.



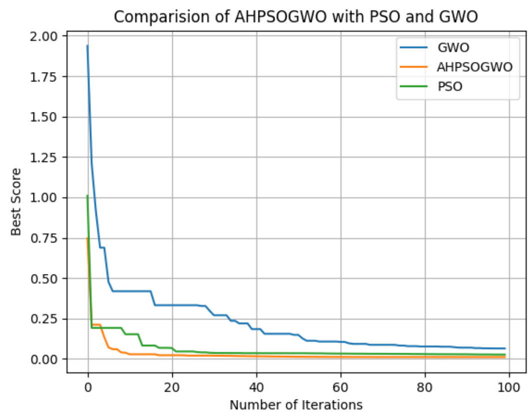
Schwefel Function i.e. F4 is working correctly and depicts that AHPSOGWO outclass both PSO and GWO by attaining the global minimum more efficiently and with greater accuracy.



Michalewicz Function i.e. F5 is finely working and indicates that AHPSOGWO outclass both PSO and GWO by reaching the global minimum more efficiently and with greater accuracy.



Rosenbrock Function i.e. F6 is finely working and demonstrates that GWO outclass both AHPSOGWO and PSO by reaching the global minimum more efficiently and with greater accuracy.



Levy Function i.e. F7 is finely working and indicates that AHPSOGWO surpass both PSO and GWO by achieving the global minimum more efficiently and with better accuracy.

VI. DISCUSSION

The analysis that took place in this project considered the PSO, GWO, and AHPSOGWO metaheuristic optimization algorithms. From the study done on a set of benchmark functions, it can be stated that the AHPSOGWO algorithm converged faster, was more accurate, and generally stable. The empirical results showed that AHPSOGWO was superior over PSO and GWO in all benchmark functions. In the case of the Schwefel function, for instance, this optimization algorithm reached the global optimum in 30 iterations, PSO reached it in 55 iterations, and GWO in 100 iterations, while in the Michalewicz function, AHPSOGWO reached this level in 63 iterations, PSO did it in 94 iterations, and GWO in 100 iterations. Similarly, AHPSOGWO performed well in multimodal functions with its capability to escape local optima. The success of AHPSOGWO lies in its adaptive mechanism in which the balance put into the exploration and exploitation is dynamically modified. Due to such property, coupling the power of PSO and GWO saves the algorithm from early convergence [9] and leads to efficient exploration of search space. The results provided a clear view of such an algorithmic success in solving many optimization problems ranging from engineering design to machine learning.

VII. CONCLUSION AND FUTURE SCOPE

Thus, the results of this study indicated the proposed AHPSOGWO algorithm is more efficient than the PSO and GWO since the AHPSOGWO can bring the advantage of PSO and GWO while avoiding the negative aspect of the two techniques through a better exploitation and exploration. The method can potentially optimise a variety of problems as the parameter setting enables the strategy to explore and exploit the search space with less complexity. Certainly, the findings of this project have significant implications for optimization problems in a vast range of fields including engineering, physical and life sciences and finance. As a tool for solving real-world issues, the proposed AHPSOGWO algorithm can provide means of optimisation that are efficient.

Prominent areas for further research are directed towards the examination of the algorithm performance with different and more challenging GBOs test problems, as well as real-world problems encountered across a variety of applications such as civil engineering, banking systems and logistics, supply chain and more. Therefore, future work involves the AHPSOGWO algorithm scalability and robustness as well as the examination of further possibilities and applicability of using AHPSOGWO approach in other problem areas. This could include verifying the approach on large instance problems; it also could mean measuring on noisy uncertain regions.

REFERENCES

[1]. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). "Grey Wolf Optimizer." *Advances in Engineering Software*, 69, 46-61.

[2]. Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). "Binary Grey Wolf Optimization Approaches for Feature Selection." *Neurocomputing*, 172, 371-381.

[3]. Mittal, N., Singh, U., & Sohi, B. S. (2016). "Modified Grey Wolf Optimizer for Global Engineering Optimization."

[4]. Gad, A. G. (2022). "Particle Swarm Optimisation." *Archives of Computational Methods in Engineering*, 29, 2531-2561.

[5]. Kennedy, J., & Eberhart, R. (1995). "Particle Swarm Optimization."

[6]. Clerc, M., & Kennedy, J. (2002). "The Particle Swarm – Explosion, Stability, and Convergence in a Multi-dimensional Complex Space."

[7]. Eberhart, R. C., & Shi, Y. (2000, July). "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization." In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC00)* (Vol. 1, pp. 84-88).

[8]. Shaheen, M. A. M., Hasanien, H. M., & Alkuhayli, A. (2021). "A Novel Hybrid GWO-PSO Optimization Technique for Optimal Reactive Power Dispatch Problem Solution." *Ain Shams Engineering Journal*, 12, 621-630.

[9]. Singh, N. (2017). "Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance." *Journal of Applied Mathematics*.

[10]. Esmin, A. A. A., & Matwin, S. (2013). "HPSOM: A Hybrid Particle Swarm Optimization Algorithm with Genetic Mutation." *International Journal of Innovative Computing, Information and Control*, 9, 1919-1934.

[11]. Ouyang, A., Zhou, Y., & Luo, Q. (n.d.). "Hybrid Particle Swarm Optimization Algorithm for Solving Systems of Nonlinear Equations."

[12]. Economic Load Dispatch Using Hybrid GWO-PSO Algorithm. *International Journal of Research in Advent Technology.*

[13]. Yu, S., Wu, Z., Wang, H., Chen, Z., & Zhong, H. (n.d.). "A Hybrid Particle Swarm Optimization Algorithm Based on Space Transformation Search and a Modified Velocity Model."

[14]. An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning.

[15]. Liu, X. H., Zhang, D. G., Zhang, T., Jie, Z., and Zhang, J. (2021). A new path plan method based on hybrid algorithm of reinforcement learning and particle swarm optimization. Eng. Comput. doi: 10.1108/EC-09-2020-0500.

[16]. X.Yu, J.Cao, H.Shan, L.Zhu, and J.Guo, “An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization,” The Scientific World Journal,vol.2014,ArticleID215472,16pages,2014.