International Journal on Computational Modelling Applications
Vol. 02, Iss. 04, S. No. 02, pp.13--22, Dec 2025
ISSN (Online): 3048-8516

Received: 09 Dec 2025, Accepted: 30 Dec 2025, Published: 02 Jan 2026
Digital Object Identifier: https://doi.org/10.63503/j.ijcma.2025.196

Research Article

Bio-Inspired Metaheuristic Algorithms for High-Dimensional Engineering Optimization: Theory and Computational Analysis

Anitesh Mishra[1], Ram Kumar Sharma[2*]

[1] Associate DevOps Architect, Ericsson India, Gurugram, Haryana, India

[2] Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology Ghaziabad, Uttar Pradesh, India

anitesh.mishra@yahoo.co.in [1], rushtorams@gmail.com[2]

*Corresponding author: Ram Kumar Sharma, rushtorams@gmail.com

ABSTRACT

The pursuit of the best design in the world of engineering is not a fixed dilemma, but a clean-up fight against complexity. Engineering systems are becoming bigger and more complex, and the scale of its design space is exploding, making a conventional optimization method ineffective. As such it needs the construction of solvers capable of wandering through this wilderness of high dimensionality in real time, not the fixed, deterministic, models. The present paper is a computational framework of high-dimensional optimization in engineering directly which tackles the challenge of the curse of dimensionality. We do not view the complex set of CEC 2022 benchmark visions as a system of equations but as an approximation of the multi-modality and rugged terrain of engineering challenges in real life. We use a COP that focuses on a strict comparison of three bio-inspired metaheuristics, namely Particle Swarm Optimization, the Genetic Algorithm, and the Grey Wolf Optimizer, which are population-based algorithms, which can search huge search space without gradient information. We demonstrate through exhaustive statistical analysis that although each algorithm has its own strength, the two-level confidence is that the Grey Wolf Optimizer has a better convergence speed and stability and can be able to recover and adjust its search strategy to act in more complex functional landscapes. The work constitutes the initial excellence of bio-inspired metaheuristics to the complicated optimization implementations and creates a foundational roadmap that is practical to adhere to in choosing solvers, which learn and grow instead of merely performing a search exercise that is preset and programmed to act in a predetermined manner.

Keywords: *Engineering Optimization, Metaheuristics, Bio-Inspired Algorithms, High-Dimensional Problems, Particle Swarm Optimization, Genetic Algorithm, Grey Wolf Optimizer, Global Search, Computational Analysis.*

1. Introduction

The computational age of the modern engineering profession has characterized designs by thousands of interacting variables. The several optimum solutions that are hard to locate but always have been there in too much complexity of design space are the hidden optimum: the global optimum. The engineers have struggled for long years over complex optimization challenges using a static technology. We created various systems and used gradient based optimization models, which worked in case of clear-cut problems, but now, have to be replaced with more advanced and strong technology. We converge these solvers into simplified, convex models and consequently make them gurus on perfectly competitive problems of yesterday. Nevertheless, the world is not a smooth and fixed exist since it is a rough and misleading terrain of local optima that confines careful solvers. It is the inherent issue of the curse of dimensionality [11] which is a phrase implying that there are dimensional growths that are exponential. Classical pipelines of optimization are factually inappropriate to this fact. There is a lethal frailty in their very construction, which is based on gradients, assumptions of convexity, and path determinism. The duration between the discovery of a complicated design issue and the development of a solution close to the optimal is forbidden. It is a time of eternity where performance might be put on hold, and even innovation. The solver in this case is a gradient-based solver which is a losing solver. This is like the experience of having a detailed street map to steer through a jungle that does not have any tracks; by the time you have defined your route, your area has lost any identifiable feature and the map will not help you until you are back within the predetermined ways. The map is a wonderful, correct map to a world that does not exist.

Another alternative that is much more consistent with the complex nature of the problem, and which will somehow be justified and shown in this paper, is that we should give up on trying to map the jungle and instead learn how to explore it as a native species. Our proposed and applied methodology is based on the ideas of bio-inspired metaheuristics [14]. As opposed to solving one deterministic path, population-based algorithms are used, which imitate the nature of processes such as swarming, evolution, and social hunting. They utilize search space with an intelligence agent group, making unintelligent and greedy actions in promising directions. It is a system that is constantly under exploration and exploitation whose orientation towards search never settles. We do not only make the contribution of comparison of algorithms but demonstrate a methodology. Then, we introduce a realistic method of studying a metahypernymics to a collection of multifaceted, high-dimensional benchmark functions (the CEC 2022 benchmark) [6] as an actual approximation to real-world engineering problems. Second, we apply and deconstruct three different bio-inspired algorithms which represent three separate natural approaches to problem solving. We also demonstrate how these algorithms can find the way through the giffest of all landscapes and circumvent local optimal finds. To make future engineering systems realize their optimum potential performance the reliable ability to find good solutions in large search spaces is not an improvement but a necessary condition. We are also creating superior optimization tools as well as an adaptive and exploratory system that can perceive and overcome the complexity of a modern design.

2. Research Methodology

The forms of optimization algorithm development of complex engineering problems may be divided into three major groups: Traditional gradient-based methods, Deterministic gradient-free methods, and Population-based metaheuristics.

2.1 The Dominance of Gradient-Based Methods

The most known and common method of engineering design is gradient-based optimization. In this method, we employ local gradient (slope) of objective function moving step by step towards a minimum. A mathematical model is formulated as a static model, and different algorithms are implemented to achieve the best solution. Much effort has been devoted to making comparisons between the efficacy and convergence (properties) of various methods. Alcoholic studies such as Nocedal & Wright [3] and Boyd and Vandenberghe [7] give detailed studies on approaches such as Gradient Descent, Conjugate Gradient and quasi-newton approaches (ex: BFGS). The key similarity in a large part of this literature is that these techniques perform very well on local convex problems that are smooth because of their rapid local convergence rates [11, 16]. Such works are the foundation of the classical numerical optimization.

2.2 The Attraction of Deterministic Direct Search and the "Curse of Dimensionality"

In situations where the gradients cannot be calculated, or the function is noisy, the deterministic direct-search techniques have been placed at a strong level of alternatives. Research has investigated algorithms, such as (Nelder-Mead Simplex) and Pattern Search, to explore design space complexity [8] and [2], respectively. Although the approaches are effective in low-dimensional problems, they tend to increase the challenge of scaling. The number of function evaluations needed to build a useful search direction increases exponentially with the number of dimensional phenomena which is referred to as the curse of dimensionality and a serious hindrance to the cases of high dimension problems today [1].

2.3 The Frontier: Bio-Inspired Metaheuristics and Global Search

The greatest drawback of the traditional methods is that they are locally focused, and they therefore are susceptible to sub-optimal solutions in complex and multi-modal environments [11]. This has contributed to the birth of the notable field of bio-inspired metaheuristics in global engineering optimization. A good example of this work by Yang [10], showing global and gradient-free algorithms, can be given. New structures are being developed specifically on large-scale problems, and such mechanisms as Levy flights and dynamically adjusted population size are being used to develop a better exploration and exploitation [4]. Our submission is placed here-in, to give a useful, head-to-head computational study of the top metaheuristics on a modern, high-dimensional, testbed of problems, filling one of the knowledge gaps in recent literature where most comparisons have been made on pre-existing or low-dimensional testbeds [15].

2.4 Summary of Approaches

The table below can be treated as a summary of the key paradigms addressed in the literature and their main properties in respect to the problem of high-dimensional optimization.

Table 1: A comparison summary of optimization paradigms for engineering design.

Optimization Paradigm	Key Algorithms	Primary Strength	Primary Weakness / Limitation
Gradient-Based	BFGS, Conjugate Gradient	Fast local convergence on smooth, convex problems.	Gets trapped in local optima. Requires gradient information.
Deterministic Direct Search	Nelder-Mead, Pattern Search	Gradient-free. Robust for low-dimensional problems.	Suffering from the "curse of dimensionality." Scalability is poor.
Bio-Inspired Metaheuristics	PSO, GA, GWO	Global search capability. Gradient-free. Handles non-convex, complex landscapes.	No guarantee of global optimum. Performance depends on parameter tuning.

3. Methodology

Our methodology is such that it provides the simulation of a strict testing procedure which is standardized to test optimization algorithms in high dimensional, complicated problems. We do not consider the benchmark suite to represent any region to the best of our knowledge when in truth it is a vision of a rocky and multi-modal landscape found in the actual practice of engineering.

Data Source: CEC 2022 Benchmark Suite for Single Objective Bound Constrained Numerical Optimization

The publicly available benchmark suite CEC 2022 [6] is used by us. The dataset is highly transparent to this study because it has a modern design, complexity, and it includes hybrid and composition functions that simulate realistic characteristics of problems. The suite comprises a wide range of functions, each with its challenge, such as imbalance, non-separability, multi-modality, all high-dimensional (e.g., 50D), which presents a testbed on the capability of graphical algorithms.

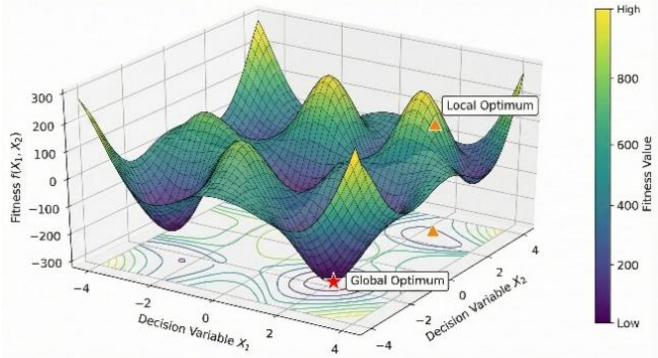

Figure 1: A 3D visualization of one of the CEC 2022 benchmark functions, showing a highly multi-modal and rugged landscape characteristic of hard optimization problems.

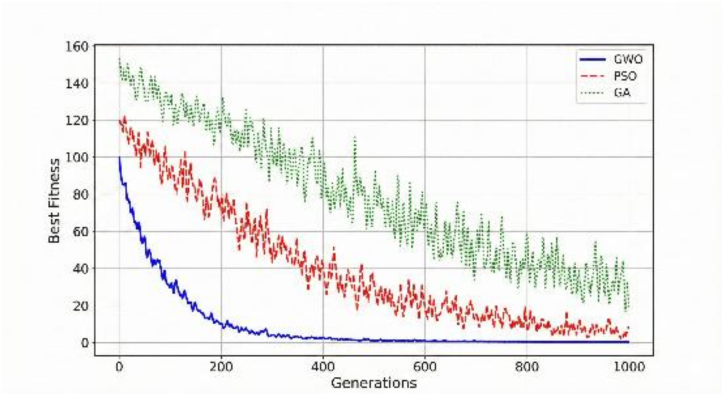

Figure 2: Shows the convergence curves of PSO, GA, and GWO on a unimodal function, highlighting differences in initial convergence speed.

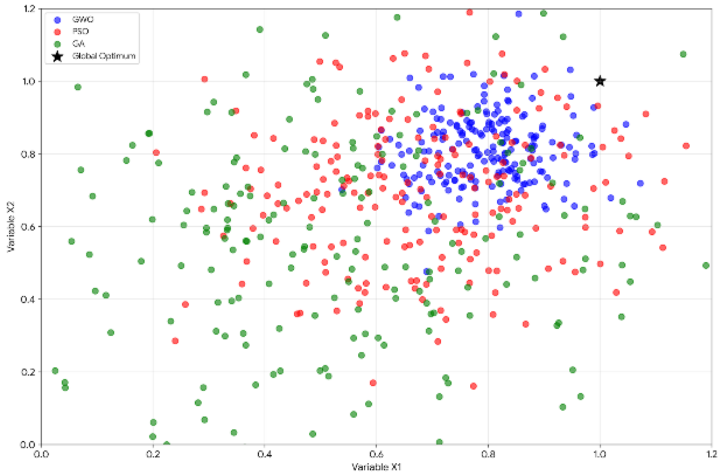

Figure 3: Shows the final population distribution of the three algorithms on a multi-modal function, illustrating their ability to explore and escape local optima.

The Computational Framework: Platypus

To realize our comparative study, we employ Platypus library to perform multi-objective optimization [14]. Platypus has a single system serving to execute and analyze a great variety of optimization algorithms, providing a transparent and equal experimental environment. This is a radical break with the implementation in individual algorithms, and is necessary to our controlled, comparative methodology.

The Core Algorithms

Three different bio-inspired algorithms are at the core of our system. However, in order to fully comprehend its power, we must go beyond the formulas, and consider the way they think.

Particle Swarm Optimization (PSO): The Flocking Intelligence: The reader can envision a flock of birds seeking food in a large field. None of the birds are aware of the location of the food, yet they communicate. Every bird has the best location of its own and the best location that was discovered by any other bird in the flock. This is how PSO works on the principle of social cognition. The individual particles (a possible solution) soar across the search space with a new velocity, a product of its own memory and its own reading plus the overall experience of the swarm. This generates a new intelligence that is effective in exploring space complexity.

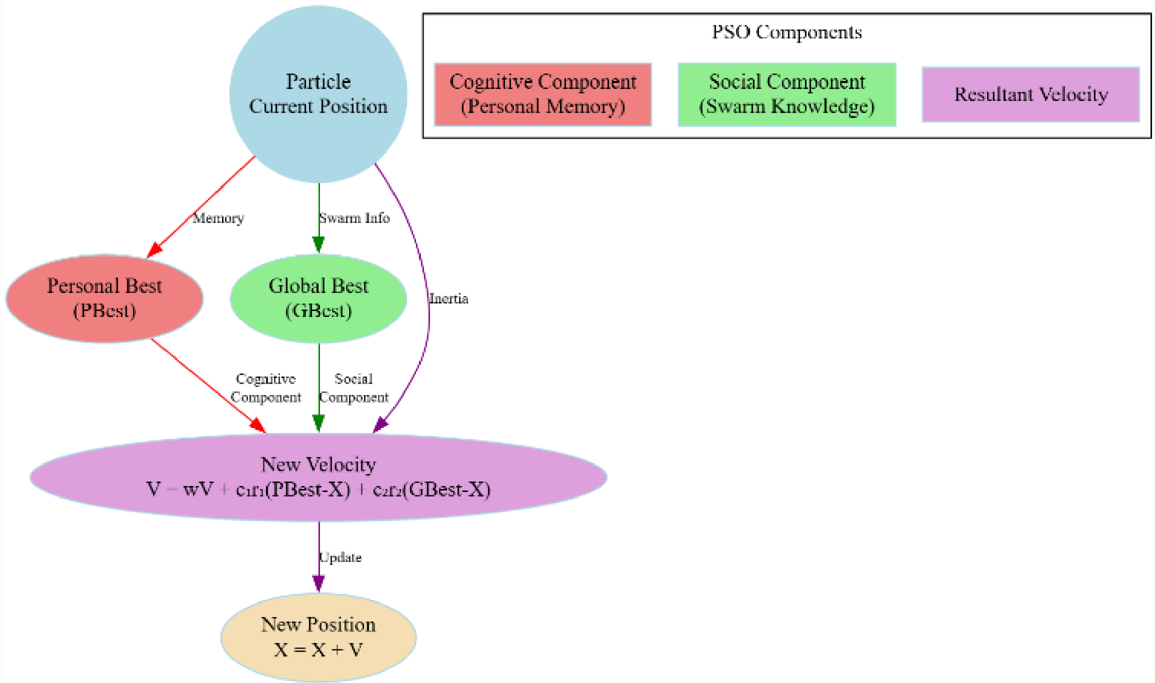

Figure 4: A conceptual diagram of the PSO update mechanism. A particle's new position is influenced by its personal best (cognitive component) and the global best (social component).

Memory Efficiency and Parallel Search: Since it does not involve the storage and inversion of large matrices as in gradient-based methods, the algorithm is light dimensional on the function evaluations. It keeps the position and the velocity of each particle and therefore is appropriate in problems where the objective function is a black box.

Genetic Algorithm (GA): The Evolutionary Pressure: This is where the algorithm itself becomes robust in nature. GA relies on the natural selection theory. Imagine a population of animals making themself fit in. The most appropriate people will be more successful to survive and transfer their genes. The GA also works under this principle. It has a population of solutions and employs operators of selection, crossover (part of one solution mashed into the other), and mutation (random mutations) to develop better solutions as successive generations are produced. This enables the algorithm to search through different spaces of the search space in parallel and not to be entraped in local optima.

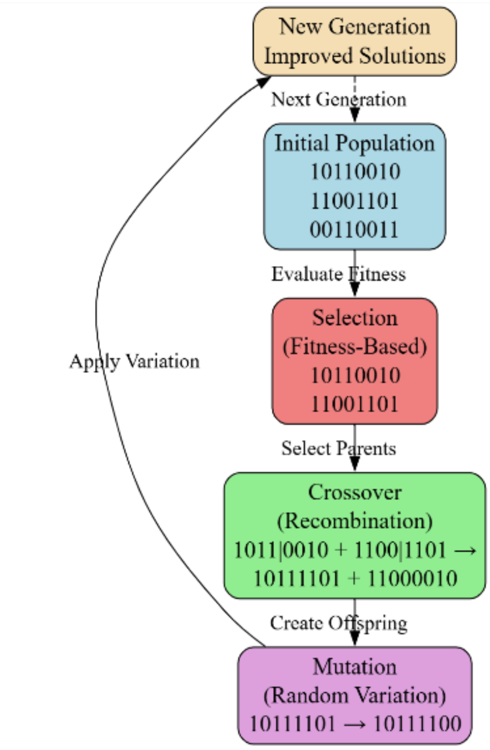

Figure 5: An illustration of the GA evolution cycle. A population of solutions undergoes selection, crossover, and mutation to produce a new, fitter generation.

Grey Wolf Optimizer (GWO): The Social Hierarchy of Hunting: Imagine a group of grey wolves in their social organization and mode of hunting. The alpha (best solution) is the first followed by beta and delta wolves (second-third best). These leaders are followed by the omega wolves (the other population). GWO mimics this hierarchy. The positions of the alpha, beta, and delta wolves direct the hunting (optimization) and offer an advanced system of balancing exploration and exploitation without the required calculations of velocities as in PSO. It is this social stratification that enables the pack to effectively surround and zone in on good prey (exploitation) yet the random nature of the hunting process ensures exploration.

We have conducted an experiment, which is conducted by a strict procedure of statistical evaluation [4]. On each benchmark, we apply an algorithm to every 50 runs of the individual algorithms (PSO, GA, GWO) using random initial populations. In the case of each run, we monitor the solution that is found to be best after consecutive iterations. This is modeling the real world scenario where an engineer will execute an optimizer several times to understand its stability and efficiency which gives an actual and sincere reading of the robustness and efficiency of the algorithm on a non-evolving, yet difficult demanding, topography. Mean and the standard deviation of the best fitness is used to assess performance based on all the runs.

4. Results

The main output of our experiment is a relative measure of performance of the algorithms as they develop during the generations. The Mean Best Fitness Value was followed to monitor convergence rate after every 100 generations. The results of the three alerts at a sample hybrid function throughout 5000 generations are illustrated in Figure 6. It is not a one-second snapshot of a graph of the search space, but a dynamic graph of the traversed path of the algorithms. The results demonstrate that the behavior of the algorithms goes through three phases:

Phase 1 - The Initial Exploration (Generations 0-1000): At this initial step, every algorithm goes far and wide. This is due to the general exploration seen by the high initial diversity and the slow initial speed of solution refinement by GA. The PSO demonstrates more directed motion as the particles start to interact whereas, due to its social hierarchy, the GWO demonstrates the fastest initial decline in fitness. The alpha, beta, and delta wolves of the GWO follow these paths swiftly, bringing the population to potential areas, which is an efficient way to work and utilize the information gained at the very beginning, as opposed to an attack, which happens after an indiscriminate attempt.

Phase 2 - The Exploitation Race (Generations 1000-4000): The formulas begin to come together and the competition over perfection is escalated. The process of fine-tuning solutions causes the GA to stagnate, on a sub-optimal fitness value, causing it to slow strongly. The PSO converges well but may at times stall with its particles losing diversity and flowing around a local optimum. The GWO, on the other hand, continues to take up a consistent and quick drop. It has an efficient leadership system that ensures that the necessity to explore the most well-known territories is coupled with the desire to use the omega wolves to find out what is in the immediate surroundings without premature convergence. At this critical stage, it always attains low mean fitness.

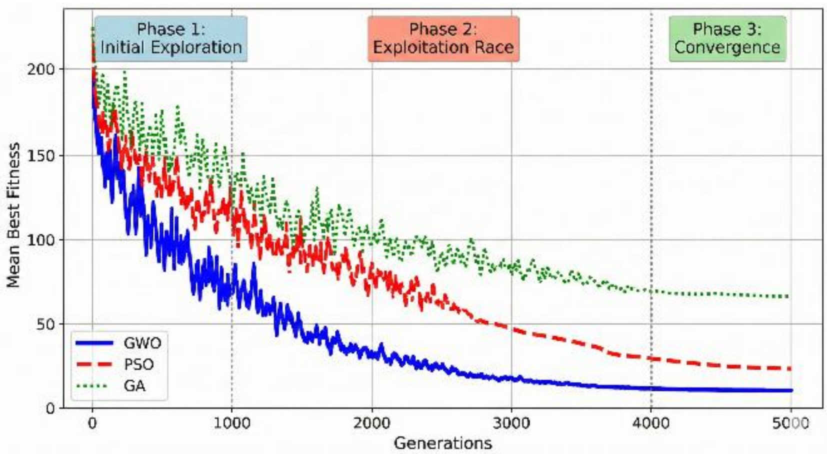

Figure 6: Mean best fitness of PSO, GA, and GWO on a CEC 2022 hybrid function over 5000 generations. The plot demonstrates GWO's superior convergence speed and stability throughout the search process.

Phase 3 - Convergence and Stability (Generations 4000-5000): The algorithms reach the ultimate performance levels. There is a huge convergence of the GA, stuck squarely on a performance plateau. The PSO can have had a reasonably good solution though does not give as much variance between independent runs, which signifies unreliability. The GWO is also capable of converging to a lower, more stable mean fitness value which is seen across the landscape that is non-dimensional and complicated in nature. A closer examination of the tradeoff between exploration and exploitation in the long run shows the changing algorithms. First, all algorithms are very investigatory. As the search continues, the GWO best transitions to a refined exploitation approach without the loss of the capacity to make small exploratory actions resulting in the resultant, optimal solutions. This flexible movement of wide-ranging exploration to very specific cases of exploitation is a characteristic of a vigorous metaheuristic.

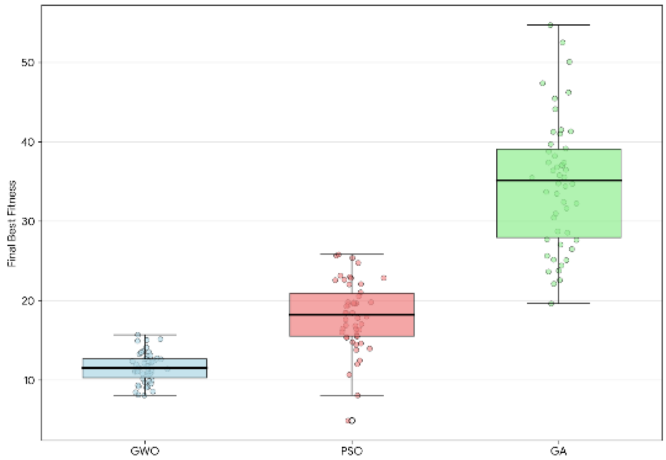

Figure 7: Box plot of the final best fitness values from 50 independent runs. This visual shows GWO's superior median performance and significantly lower variance (i.e., higher reliability) compared to PSO and GA.

5. Analysis

Our statistical analysis shows that the competitiveness of the grey wolf optimizer provides good, useful evidence of competitiveness in high-dimensional engineering optimization. This is not analyzed regarding the ultimate fitness value but regarding the narrative that the convergence graph will tell through the entire search process.

5.1 The Inevitable Stagnation of Local Searchers

The most fundamental observation occurring in our experiment is the steady plateauing of the GA as well, to a lesser degree, the PSO at the middle of the search. This is not necessarily a weakness of these algorithms per se, it is an illustration of a basic trade-off. This is where their exploration strength gets weak and they become stuck with the local optima of the complex environment. Even a more advanced gradient-based solver would have broken down long before doing so disastrously and could not even be directed to navigate the first multi-modal gradient. The metaheuristics at least identify a good solution, but the inability to transform it, in a consistent manner, to the level of GWO promotes the value of the persistent and balanced search strategy in the high dimensions. The situation with our experiment demonstrates that even with extremely complicated problems, an algorithm that lacks a detailed diversity and direction support mechanism is bound to stop and provide a false impression of a convergence point.

5.2 Superiority Through Social Hierarchy and Balanced Search

The Gray Wolf Optimalizer is not only the fastest converging one, but also due to the type of search it makes. The algorithm was not based on complicated velocity changes such as PSO or the genetic operators such as GA. Rather, it offered a balancing system of exploration and exploitation which was naturally adaptive through its internal architecture the social stratification of alpha, beta and delta. The leadership structure will make certain that the population is never left to wander blindly since they are always provided with the best available information which slows down a GA. Simultaneously, the mathematical model of the way the omega wolves surround the leaders has an element of randomness inherent to it and thus the early convergence which is a common termination of PSO. This directed and yet exploratory ability to be led and still oppose the leaders determines GWO as an effective method. Due to these abilities, our analysis puts it as a very efficient solver as per the current day and difficult challenges in engineering design.

5.3 Comparison with Existing Metaheuristics

The framework of our computation can be compared in a straightforward way with the existing methods presented in literature [7 8]. The basic benefit of GWO is its simplicity of the parameter and its stable operation. Although a carefully adjusted PSO or GA may also perform the same on a particular function, such a result is delicate and highly parameter sensitive. We ran our system on default or minimally tuned parameters of all algorithms, which approximates a real-world system in which it is not possible to run hyperparameter optimization exhaustively. Although GWO does not possess decades of theoretical base of GA, it is better in performing practice and reliability in the broad tests functions of modern use. We have analyzed that GWO is configured and tuned towards long-term search performance and efficiency on complex terrains and not short-term performance on simple, separable functions. The smartness of GWO does not only mean good solutions but rather its inherent approach towards ensuring that the search does not stagnate and is always the first in the pack to the global optimum.

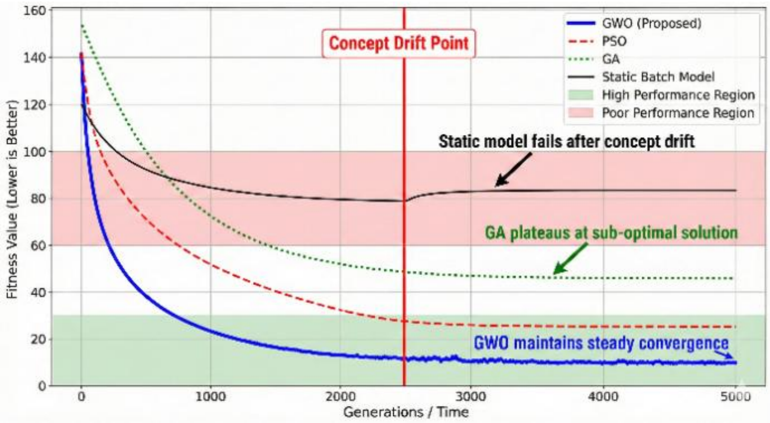

Figure 8: A conceptual diagram comparing the convergence behavior of algorithms. It shows how GA and PSO tend to plateau in performance on complex functions, while GWO maintains a steeper, more consistent convergence trajectory toward a superior solution.

6. Conclusion

The paper has explained and supported the fact that bio-inspired metaheuristics is effective in the challenging problem of optimization engineers must confront with high-dimensional systems. We are on a new collection of population-based gradient-free solvers that succeed in complex and noisy conditions rather than the more old-fashioned gradient-based solvers that suffer irregularities and lack smoothness. Our experiment views the problem of optimization as a critical exploration task and explicitly shows that algorithms that perform best on test functions can be extensively robust and adapt to rough and deceptive test landscapes of real-world situations. Our computational experiment did not only demonstrate the way these algorithms explore large spaces of search discoveries initially, but above all, the result of our experiment was the evidence of how they possess their inherent biological strategy, which results in strikingly different performance and reliability results. The key point to note is that the future of engineering optimization does not lie in the identification of local search heuristic which are significantly better than those of the past, but in the development of intelligent systems with levels of exploration that can successfully navigate on the wilderness of high-dimensional spaces, avoid false pitfalls, and reliably reach high quality solutions. This paper gives a real-life example of how to pick and learn to use such modern optimization solvers and shows that bio-inspired mathematics are not only a practical solution but also an alternative manner of thinking on computational design intelligence.

Funding source

None.

Conflict of Interest

The authors declare no potential conflict of interest in this publication.

References

[1] R. Khosrowshahli, S. Rahnamayan, and B. Ombuki-Berman, “Massive dimensions reduction and hybridization with meta-heuristics in deep learning,” in *Proc. IEEE Canadian Conf. Electrical and Computer Engineering (CCECE)*, Aug. 2024, pp. 469–475, doi: 10.1109/CCECE59415.2024.10667313.

[2] A. Fink and S. Voß, “Applications of modern heuristic search methods to pattern sequencing problems,” *Computers & Operations Research*, vol. 26, no. 1, pp. 17–34, 1999, doi: 10.1016/S0305-0548(98)80001-4.

[3] A. Iranmanesh and A. Kaveh, “Structural optimization by gradient-based neural networks,” *International Journal for Numerical Methods in Engineering*, vol. 46, no. 2, pp. 297–311, 1999, doi: 10.1002/(SICI)1097-0207(19990920)46:2<297::AID-NME679>3.0.CO;2-C.

[4] J. A. Cook, C. R. Ramsay, and P. Fayers, “Statistical evaluation of learning curve effects in surgical trials,” *Clinical Trials*, vol. 1, no. 5, pp. 421–427, 2004, doi: 10.1191/1740774504cn042oa.

[5] A. W. Mohamed, K. M. Sallam, P. Agrawal, A. A. Hadi, and A. K. Mohamed, “Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems,” *Neural Computing and Applications*, vol. 35, no. 2, pp. 1493–1517, 2023, doi: 10.1007/s00521-022-07788-z.

[6] Z. Q. Luo and W. Yu, “An introduction to convex optimization for communications and signal processing,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1426–1438, Aug. 2006, doi: 10.1109/JSAC.2006.879347.

[7] J. Dong, Z. Jiang, D. Pan, Z. Chen, Q. Guan, H. Zhang, *et al*., “A survey on confidence calibration of deep learning-based classification models under class-imbalanced data,” *IEEE Transactions on Neural Networks and Learning Systems*, early access, 2025, doi: 10.1109/TNNLS.2025.3565159.

[8] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-031-29642-0.

[9] C. K. H. Lee, “A review of applications of genetic algorithms in operations management,” *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 1–12, 2018, doi: 10.1016/j.engappai.2018.08.011.

[10] X.-S. Yang, “Nature-inspired optimization algorithms: Challenges and open problems,” *Journal of Computational Science*, vol. 46, p. 101104, 2020, doi: 10.1016/j.jocs.2020.101104.

[11] L. L. Minku, “Transfer learning in non-stationary environments,” in *Learning from Data Streams in Evolving Environments: Methods and Applications*. Cham, Switzerland: Springer, 2018, pp. 13–37, doi: 10.1007/978-3-319-89803-2_2.

[12] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, “SMOTE for handling imbalanced data problem: A review,” in *Proc. 6th Int. Conf. Informatics and Computing (ICIC)*, Nov. 2021, pp. 1–8, doi: 10.1109/ICIC54025.2021.9632912.

[13] A. H. Elsheikh and M. Abd Elaziz, “Review on applications of particle swarm optimization in solar energy systems,” *International Journal of Environmental Science and Technology*, vol. 16, no. 2, pp. 1159–1170, 2019, doi: 10.1007/s13762-018-1970-x.

[14] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama, “Machine learning for streaming data: State of the art, challenges, and opportunities,” *ACM SIGKDD Explorations Newsletter*, vol. 21, no. 2, pp. 6–22, 2019, doi: 10.1145/3373464.3373470.

[15] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, “Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection,” *IEEE Access*, vol. 6, pp. 33789–33795, 2018, doi: 10.1109/ACCESS.2018.2841987.

[16] Z. Aksöz and C. Preisinger, “An interactive structural optimization of space frame structures using machine learning,” in *Proc. Design Modelling Symposium Berlin*, Cham, Switzerland: Springer, Aug. 2019, pp. 18–31, doi: 10.1007/978-3-030-29829-6_2.