Atlas Journal of Computer Science

Journal homepage: www.ajocs.atlasci.org

Volume 1, Issue 1 (2026) 1–12

ISSN: 3119-6810

Review Article

Metaheuristic Optimization Algorithms: A Comprehensive Survey of Taxonomy, Recent Advances, and Future Directions

Guma Ali [1, *], Wamusi Robert [2], Habib Hassan [3], Bosco Apparatus Buruga [4]

[1] Department of Computer and Information Science, Faculty of Technoscience, Muni University, Arua, Uganda

[2] Department of Computing and Technology, Faculty of Engineering, Design, and Technology, Uganda Christian University, Arua Campus, Arua, Uganda

[3] Department of Computer Science, Faculty of Science, Islamic University in Uganda, Arua Campus, Uganda

[4] Department of Library and Information Services, Muni University, Arua, Uganda

ARTICLE INFO

Article history

- **Received: 02 February 2025**
- **Revised: 11 February 2025**
- **Accepted: 20 February 2025**
- **Online : 20 February 2025**

Keywords

- Metaheuristics
- Nature-Inspired Optimization
- Evolutionary Algorithms
- Swarm Intelligence
- Hybrid Metaheuristics
- Engineering Design Optimization

ABSTRACT

In recent decades, the metaheuristic optimization algorithms have emerged as an indispensable paradigm for tackling complicated non-convex, combinatorial, and complex problems that are not solved efficiently using classical gradient-based optimization algorithms. During the past three decades, hundreds of metaheuristics based on the concepts of evolution, collective behavior of animals, physical phenomena, human beings, etc., have been introduced. There has been extensive literature regarding different metaheuristic algorithms, although quite dispersed and fragmented at the same time. In this survey, we classify the metaheuristic optimization methods into five branches based on our taxonomy of evolutionary-based, swarm intelligence-based, physics-based, human-based, and hybrid or emerging methods, and review the principles, operators, advantages, and limitations of some of the landmark algorithms within each branch. We use a generic search loop structure to discuss the principle of balance between exploration and exploitation, which is the governing principle of the convergence behavior of all metaheuristic algorithms. Additionally, we consolidate some of the most popular benchmarks for constrained optimization and computational intelligence tasks in engineering applications, highlight performance results reported in the literature, and identify important open challenges, including scalability issues, sensitivity to tuning parameters, and premature convergence, among others. Finally, we discuss some of the emerging directions, including LLM-assisted meta-optimization, surrogate-assisted search for expensive objective functions, hybrid architectures, etc.

1. Introduction

Engineering, machine learning, logistics, and computational biology are among the fields where optimization problems occur, which are characterized by non-convex, non-differentiable, and discontinuous nature, and a large (or mixed-integer) search space. Classical gradient-based or exact optimization algorithms fail to handle such problems, as their application requires smoothness assumptions or becomes computationally infeasible as the problem grows. Metaheuristic optimization algorithms were developed to solve this problem by

* Corresponding author.
E-mail address: a.guma@muni.ac.ug

introducing stochastic, population-based search procedures that do not need gradient information and allow for getting out of local minima due to various mechanisms of randomized perturbation, recombination, or information exchange.

A metaheuristic is a high-level algorithmic scheme that steers a heuristic search to efficiently explore a solution space regardless of the nature of a particular problem. This approach became very popular since the appearance of Genetic Algorithms and Simulated Annealing in the 70's and 80's respectively. It has developed very quickly, especially after the invention of Particle Swarm Optimization and Ant Colony Optimization in the 90's. In the past decade, there is a rapid growth of nature-inspired metaheuristics, many of which are inspired by animal hunting behavior or social processes. This tendency is somewhat disturbing in the literature, e.g., the 2024/2025 review of the pioneers of metaheuristic optimization counted more than 150 new variants of algorithms in just six years [19].

It is this abundance that drives this survey. Instead of trying to enumerate all possible variations, this paper:

- Presents a taxonomy based on the inspiration source and search methodology,
- Discusses key algorithms from each taxonomy branch at a level of technical detail sufficient for choosing an algorithm,
- Provides information on the benchmark problems and evaluation methodology that are commonly employed to compare the algorithms, and
- Outlines open problems and possible directions for future research, such as the intersection of metaheuristics and language models.

1.1 Contributions of this Survey

- A five-branched classification scheme for meta-heuristic optimization algorithms, with representative landmark algorithms in each branch (Figure 1).
- A unified framework describing the generic meta-heuristic search process and exploration vs exploitation problem faced by meta-heuristics from all classes (Figure 3).
- A systematic overview of the evolutionary, swarm intelligence, physics-based, human-inspired, and hybrid/emerging algorithms based on primary, DOI-indexed sources.
- Systematically compiled data tables summarizing the properties of different algorithms and engineering/CEC benchmark problems for testing algorithms.
- The consolidation of open research issues and future research directions, including LLM-based meta-optimization.

1.2 Organization of the Paper

Section 2 provides the review methodology. The proposed taxonomy is presented in Section 3. Section 4 shows the metaheuristic search process cycle. Sections 5-9 give a review of evolutionary-based, swarm intelligence-based, physics-based, human-based, and hybrid/algorithms respectively. Section 10 gives the placement of landmark algorithms within a timeline. Section 11 gives the comparison study. Section 12 gives the application areas. Section 13 gives open problems, and future directions are given in Section 14.

2. Review Methodology

Literature was sourced by searching specifically for material in IEEE Xplore, ScienceDirect (Elsevier), SpringerLink, ACM Digital Library, MDPI publications, arXiv, and Web of Science databases using search terms such as 'metaheuristic,' 'nature-inspired optimization,' 'swarm intelligence,' 'evolutionary algorithm,' 'hybrid optimization' in combination with names of various landmark algorithms. Papers on landmark algorithms from 1975 to 2023, irrespective of publication venue, were considered if the papers proposed an algorithm that is still being used as one of the algorithms compared in contemporary papers. Survey and review articles from the years 2019 to 2026 were preferred to understand the current taxonomy, benchmarking practices, and issues in the field. Only those sources were considered that had either a DOI or an arXiv ID; grey literature, blog posts, and other non-verifiable sources were not considered. The screening process is illustrated in Figure 4 below.

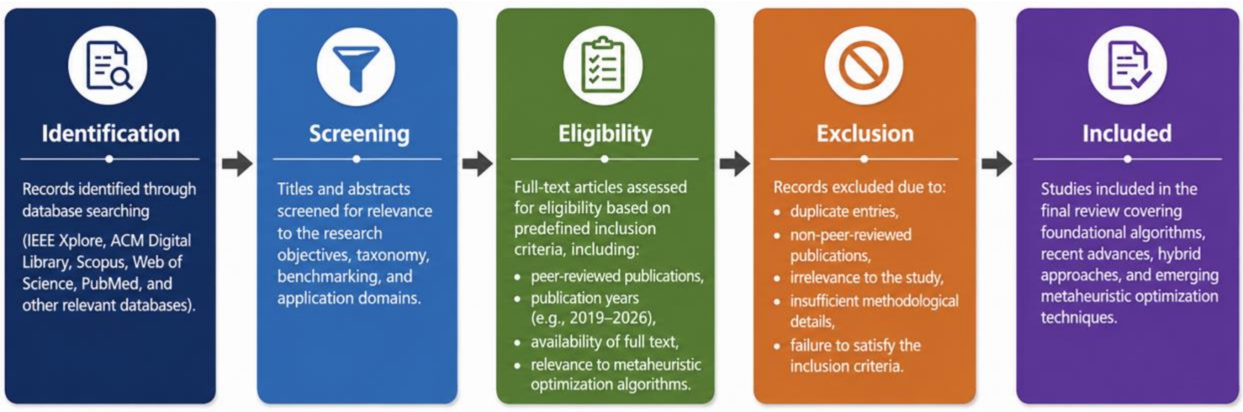

Figure 4. Literature identification and screening workflow applied in this survey.

3. Taxonomy of Metaheuristic Algorithms

Many taxonomies for metaheuristic approaches can be found in the literature that are mostly defined based on the source of inspiration, the number of candidate solutions involved in the process, or the balance between exploration and exploitation. The survey employs a five-branched taxonomy based on the source of inspiration, namely, evolutionary-based, swarm intelligence-based, physics-based, human-based, and hybrid/emerging. The rationale behind using the taxonomy above is its widespread adoption in recent comparison-based surveys [19,20,23] and effectiveness in choosing algorithms intuitively by practitioners. Figure 1 depicts this taxonomy with sample algorithms from each branch.

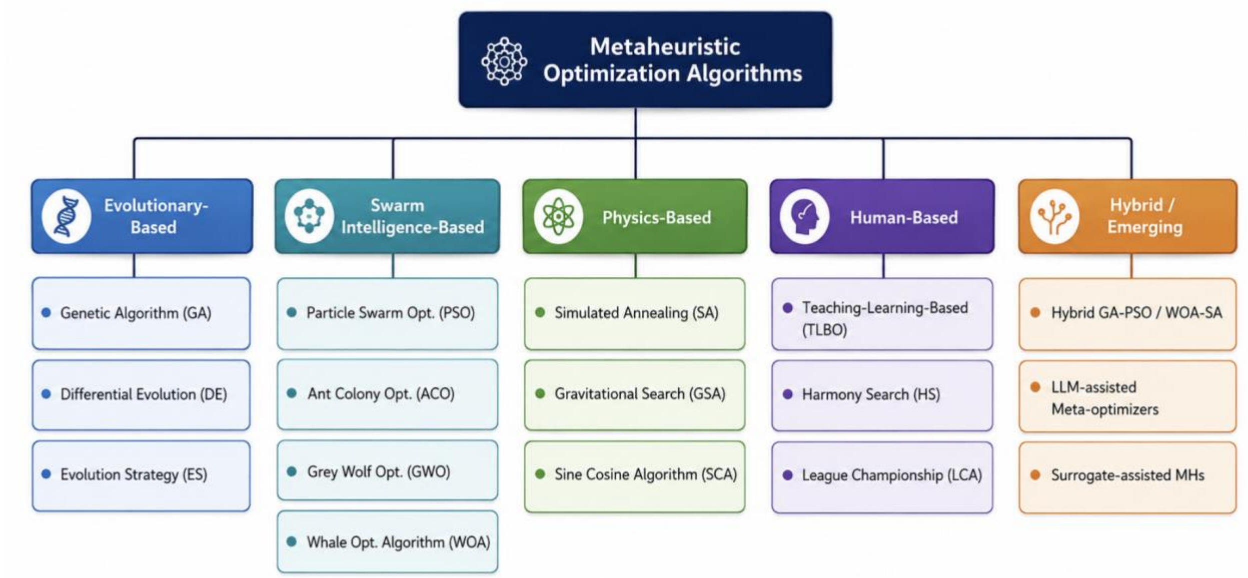

Figure 1. Proposed taxonomy of metaheuristic optimization algorithms.

Noteworthy to point out is the fact that the taxonomy mentioned above is not mutually exclusive as many algorithms of today use operators from more than one category (e.g., some swarm algorithms use acceptance rules from the simulated annealing category). This is why the hybrid/emerging category is included in the classification.

3.1. Foundational Concepts: The Generic Metaheuristic Search Cycle

Although different metaphors may be used, almost all population-based meta-heuristics have an iteration process which consists of:

1. Generation of a set of initial candidate solutions, randomly generated in the admissible search space.
2. Evaluation of the objective function (fitness function) for each candidate.

3. Updating of candidate solutions through specific algorithmic mechanisms that involve both exploration and exploitation of the search space; (iv) evaluation of the termination criterion (number of iterations reached, value of the fitness function achieved or lack of progress); and (v) continuation of the process until the criterion is met.

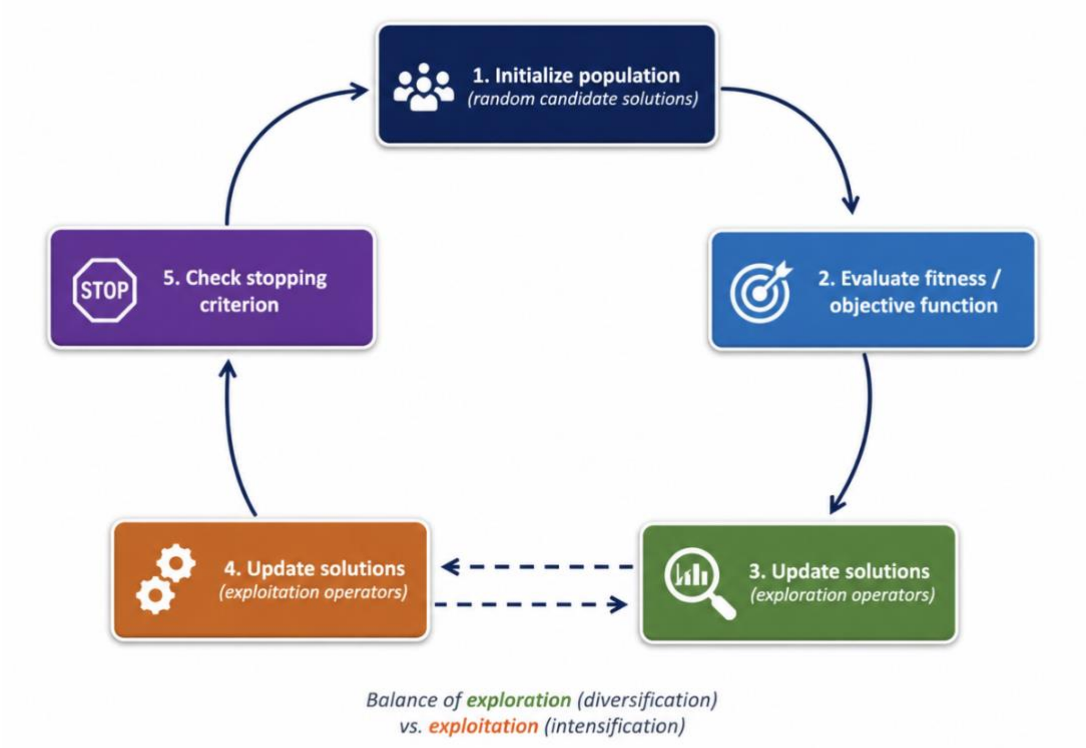

Figure 3. Generic iterative search cycle common to population-based metaheuristics.

The balance between exploration and exploitation demonstrated in Figure 3 is at the heart of the problem of designing good metaheuristics: algorithms which are exploratory enough take a long time to converge, whereas algorithms which are exploitative enough may easily find themselves in a local minimum prematurely. Virtually all the algorithmic developments mentioned in Sections 5–9 fall under this description.

3.2. Evolutionary-Based Algorithms

Evolutionary algorithms (EAs) emulate principles of natural selection through survival of the fittest, crossover, and mutation of candidate solutions ('individuals') in each generation in the hope that the population's fitness will gradually increase.

- **Genetic Algorithm (GA)**

The Genetic Algorithm, proposed by Holland, represents candidate solutions using chromosomes (originally binary strings, but nowadays often real-valued) and uses a set of selection operators (roulette-wheel selection, tournament selection, etc.), a set of crossover operators (single-point crossover, uniform crossover), and a mutation operator to form generations [3]. GA is one of the most popular and extensively researched metaheuristics, which schema theorem offers probably the only proven mechanism of convergence among all evolutionary metaheuristics.

- **Differential Evolution (DE)**

Differential Evolution, suggested by Storn and Price, creates new candidate solutions by summing up a scalar multiple of the difference between two randomly picked population vectors and another vector ('mutation'), then crossing over the resulting solution with a target vector and choosing greedily between trial and target vectors [4]. DE is known for good performance on continuous, real-parameter optimization tasks combined with relatively simple control parameter setting (population size, scaling factor F, crossover rate CR), and there is an immense amount of literature surveys on the topic of various DE versions developed over decades [17,18].

- **Evolution Strategies (ES)**

Evolution Strategies, invented by Rechenberg and Schwefel, are centered around the self-adaptation of mutation step sizes encoded together with the solution vector, enabling evolution strategies to adjust their own search radius throughout the process. The contemporary descendants of ES like CMA-ES (Covariance Matrix Adaptation ES) take this principle even further and include covariance matrices in addition to the self-adaptation of stepsizes.

3.3. Swarm Intelligence-Based Algorithms

The algorithms based on swarm intelligence (SI) represent computational models of the problem-solving behaviour observed in social organisms (flocks, colonies, and packs), wherein global search behaviour arises through simple interactions between individuals, without any central coordination.

- **Particle Swarm Optimization (PSO)**

Proposed by Kennedy and Eberhart, the PSO represents an algorithm, in which each candidate solution (a 'particle') has its own position and velocity in the search space [5]. The velocity of the particles is updated according to their best-known positions and the swarm best-known position, resulting in a simple two-term updating formula with outstanding performance on the wide range of continuous optimization problems.

- **Ant Colony Optimization (ACO)**

Based on the behaviour of the real ants leaving pheromone trails, the ACO algorithm, invented by Dorigo et al., is especially effective for solving combinatorial and graph-related problems like traveling salesman problem and routing [6, 27]. The artificial ants incrementally construct solutions leaving simulated pheromone proportionate to solution quality and evaporating the pheromone to prevent the premature convergence to the non-optimal solution.

- **Artificial Bee Colony (ABC)**

The algorithm suggested by Karaboga and Basturk divides the population into the employed bees, onlooker bees, and scout bees, modelling the food search division of labour in honeybees to balance the exploitation of known food sources (candidate solutions) and search for new ones [7].

- **Grey Wolf Optimizer (GWO)**

Suggested by Mirjalili, Mirjalili, and Lewis, the GWO algorithm takes the leadership hierarchy and cooperative hunting behavior of grey wolf packs into account. Alpha, beta, and delta (the three best wolves) direct the position updating process of the other wolves. This simple algorithm with a small number of parameters became one of the most cited SI algorithms in recent years because of its performance on standard test functions.

- **Whale Optimization Algorithm (WOA) and Ant Lion Optimizer (ALO)**

The WOA, designed by Mirjalili and Lewis, simulates the humpback whale bubble-net feeding hunting approach by implementing the spiral movement strategy, which serves as an exploitation strategy around the prey, along with the random walk movement strategy used for exploration [12]. Similarly, the ALO, another of Mirjalili's contributions to the field of swarm optimization algorithms, uses the pit trap feeding hunting approach of the ant lion larvae and random walk-based movement of prey (ant) as an exploration and exploitation approach [11].

- **Harris Hawks Optimization (HHO)**

The HHO developed by Heidari et al. simulates the cooperative surprise-pounce hunting behavior of Harris's hawk, employing multiple exploration and exploitation strategies using the escaping energy of the prey, and has been successfully implemented on various engineering problems since its proposal [15].

- **New and Emerging Swarm-Based Variants**

The swarm intelligence paradigm has continuously brought forth new suggestions with great speed in recent years; some such recent suggestions include the FOX-inspired optimization algorithm based on fox hunting [16], as well as many others found in the literature of recent comprehensive reviews [19,20]. It is to be noted that several reviewers warn about the nature of newly proposed swarm variants, which tend to be novel only from an incremental/metaphorical point of view, necessitating the comparison framework proposed in Section 11 [19].

3.4. Physics-inspired Algorithms

Physics-inspired algorithms obtain the inspiration for their evolution operators based on laws and phenomena such as thermodynamics, gravity, and wave mechanics.

- **Simulated Annealing (SA)**

This algorithm, invented by Kirkpatrick et al., simulates the annealing process of metallurgy, where an object is heated up and cooled down to reach a state with minimal energy level. In SA, worse solutions are accepted depending on a temperature parameter that decays in accordance with a certain schedule, thus providing both exploration at an early stage and exploitation at the later stages in a one-solution framework.

- **Gravitational Search Algorithm (GSA)**

Proposed by Rashedi et al., this algorithm is based on the interaction between the masses that corresponds to Newton's Law of Gravitation and Motion, where the fitness value of a mass defines its gravitational 'mass' and, consequently, its influence on the other candidate solutions [8]. The heavier (more fit) masses move more slowly and pull the other ones more strongly, naturally leading to exploitation-oriented search.

- **Sine Cosine Algorithm (SCA)**

The algorithm suggested by Mirjalili uses sine and cosine functions for changing the positions of the candidate solutions, with the amplitude of the changes decreasing over iterations [13].

3.5. Human-Based Algorithms

Metaheuristics in this category mimic social, cognitive, or competitive behavior that is human-like.

- **Teaching-Learning Based Optimization (TLBO)**

TLBO algorithm was developed by Rao, Savsani, and Vakharia, which mimics the learning process that takes place in a classroom through the two steps: in the 'teacher phase', the population means are adjusted towards the best solution (i.e., teacher), and in the 'learner phase', improvement happens through learner-to-learner interactions [9]. The major strength of the TLBO algorithm is the lack of any algorithm-dependent parameters other than population size and number of iterations.

- **Harmony Search (HS) and League Championship Algorithm (LCA)**

The harmony search algorithm works on the principle of how musicians search for better harmony while improvising their music by using harmony memory, pitch adjustment, and random selection of candidate solutions. LCA algorithm mimics the working of a sports league where team strength gets improved through competitions between teams (i.e., solutions).

3.6. Hybrid and Novel Metaheuristics

Hybrid metaheuristics integrate operators or even entire phases of algorithms from any two or more branches discussed above to complement the strengths of one by overcoming the weaknesses of the other – for instance, incorporating rapid convergence of a swarm algorithm with an acceptance criterion like that of SA to combat premature convergence. The literature examples include hybrid GWO-GA approaches for engineering optimization of continuous spaces and WOA-SA for feature selection and combinatorial problems.

- **LLM-Assisted Meta-Optimization**

One of the newest, yet quickly rising, directions is using large language models (LLMs) for meta-optimization – instead of designing operators manually, an LLM proposes, mutates or selects algorithmic elements based on the textual description of the problem and feedback from previous search phases. One of the recent systematic reviews describes and formalizes the proposed paradigm as the possibility of LLM-based meta-optimization leading to improved cross-domain generalization and reduced parameter sensitivity of many manually designed metaheuristics [24]. This direction is still very novel, and there remain numerous open issues related to computational overhead, reproducibility, and fair comparisons with existing metaheuristics.

- **Reinforcement-Learning-Assisted Metaheuristics**

Another, although less novel, hybrid approach relies on reinforcement learning, specifically Q-learning, to adaptively select operators or tune metaheuristic parameters during the optimization process instead of having them defined beforehand. An extensive survey covering two decades of literature reveals the increasing use of the integration for solving energy and power systems-related problems and highlights transfer learning and state space reduction as the major remaining challenges [21].

4. Historical Timeline of Landmark Algorithms

Figure 2 provides a historical timeline of the landmark algorithms discussed in Sections 5-9, with colours representing different branches of taxonomy. It shows two important patterns: the first is the gap of nearly two decades between the early evolutionary algorithms and physics-inspired algorithms (1970s-1990s) and the rapid increase in proposals of swarm intelligence techniques since the mid-2000s; and the second is the emergence, from 2019 onwards, of human-inspired and hybrid/emerging techniques based more on learning rather than metaphorical operators.

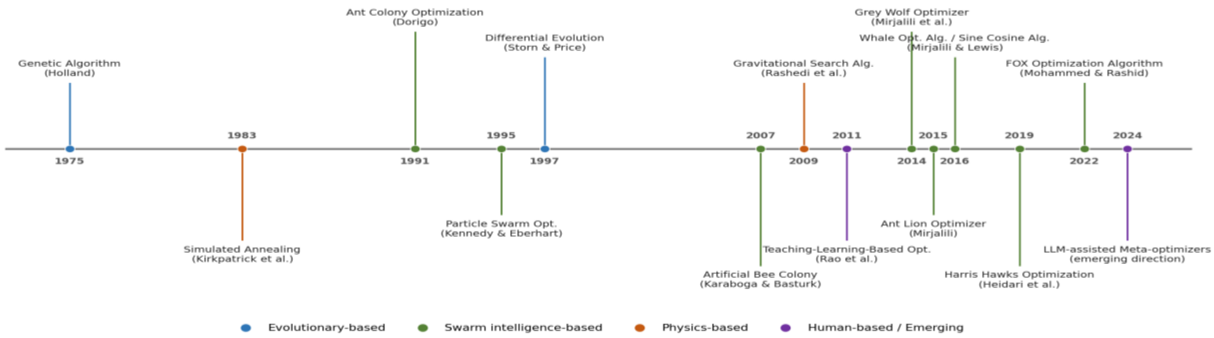

Figure 2. Timeline of landmark metaheuristic algorithms referenced in this survey.

5. Comparative Analysis

Table 1 summarizes the five taxonomic branches, their representative algorithms, inspiration sources, key operators, and commonly reported strengths and limitations, synthesized from the algorithm-specific literature reviewed in Sections 5-9 and from recent comparative surveys [19,20,22,23,25,26].

Table 1. Comparative analysis of the five taxonomic classes of meta-heuristic optimization algorithms.

Category	**Representative algorithms**	**Inspiration source**	**Key operators**	**Typical strengths**	**Typical limitations**
Evolutionary-based	GA, DE, ES	Darwinian evolution and natural selection	Selection, crossover, mutation	Strong global search; well studied theoretically	Can converge slowly; many control parameters
Swarm intelligence-based	PSO, ACO, ABC, GWO, WOA, ALO, HHO	Collective/social behaviour of animal groups	Position/velocity update, pheromone trails, leader-following	Simple implementation; fast convergence; few parameters	Prone to premature convergence and local optima on multimodal problems
Physics-based	SA, GSA, SCA	Physical laws (thermodynamics, gravity, trigonometry)	Temperature cooling schedules, force/mass interactions, sine-cosine oscillation	Mathematically grounded; often good exploration	Sensitive to initial parameter settings (e.g., cooling schedule)

Human-based	TLBO, HS, LCA	Human social or cognitive processes	Teacher-learner interaction, improvisation, competition	Few or no algorithm-specific parameters (e.g., TLBO)	Fewer theoretical convergence guarantees; less mature literature
Hybrid / emerging	Hybrid GA-PSO, WOA-SA, LLM-assisted meta-optimizers	Combination of two or more paradigms; data-driven design	Mixed operator pools, adaptive switching, learned heuristics	Combines complementary strengths; increasingly competitive on benchmarks	Added design/tuning complexity; reproducibility and fair-comparison concerns

Figure 5 provides additional information to Table 1 in the form of a qualitative literature-derived comparison between five algorithms that can be considered representative within their respective classes (one from each class of SIs/ EAs plus one algorithm from a class similar to physics-related algorithms). The evaluation criteria used below are indicative rather than definitive, as they represent a generalized evaluation based on general consensus described in a comparative literature review rather than an experiment conducted under unified conditions.

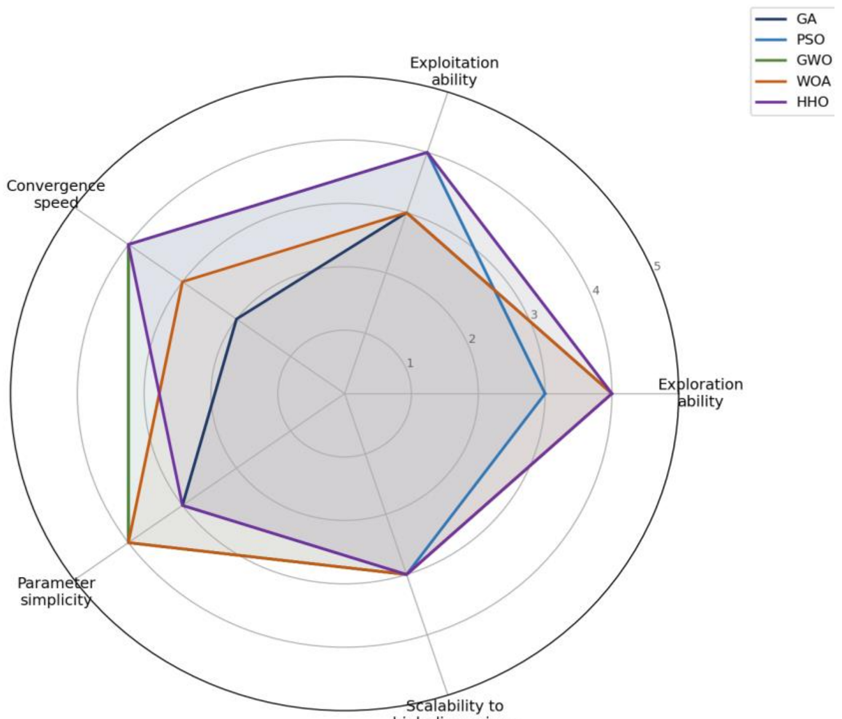

Figure 5. Qualitative comparison of five representative metaheuristic algorithms.

Comparing the performance of metaheuristics through a fair and statistically sound method involves the use of standard benchmark functions and non-parametric statistics that suit stochastic methods (such as the Friedman test for ranking in comparison among multiple algorithms/functions, followed by Wilcoxon signed rank test for pairwise comparisons). The benchmark functions currently in use are mentioned in Table 2 below.

Benchmark Problem	Optimization Type	Design Variables	Number/Type of Constraints	Typical Application in the Literature
Welded Beam Design	Constrained, continuous	4 (weld size, weld length, beam height, beam thickness)	7 (shear stress, bending stress, buckling load, end deflection, etc.)	Widely adopted for evaluating constraint-handling techniques and balancing structural weight with mechanical feasibility.
Pressure Vessel Design	Constrained, mixed-integer	4 (shell thickness, head thickness, inner radius, vessel length)	4 (material thickness, volume, and design constraints)	Standard benchmark for assessing mixed-integer optimization and discrete variable handling.
Tension/Compression Spring Design	Constrained, continuous	3 (wire diameter, mean coil diameter, number of active coils)	4 (shear stress, surge frequency, deflection, outside diameter)	Popular low-dimensional engineering problem for validating convergence speed and constraint satisfaction.
Speed Reducer Design	Constrained, continuous	7 (gear face width, module, shaft dimensions, etc.)	11 engineering constraints	Frequently employed to evaluate scalability and performance on higher-dimensional constrained optimization problems.
CEC Benchmark Suites (CEC2017, CEC2020, CEC2022)	Unconstrained and constrained, continuous	Scalable (typically 10–100 dimensions)	Problem-dependent	International standard benchmark suite for statistically rigorous comparison of optimization algorithms using Friedman, Wilcoxon, and related non-parametric tests.

6. Applications of Metaheuristic Algorithms

- Optimizations in Engineering Design

Constrained mechanical and structural design optimization problems (see Table 2) are still the most typical validation domain of novel metaheuristics because of well-defined and reproducible objective and constraint functions, as well as popularity as baselines to compare algorithm performance in Section 5-9.

- Scheduling Problems and Combinatorial Optimization

Many metaheuristics are applied to scheduling problems of NP-hard nature, such as flow-shop and job-shop scheduling problems, especially in modern industry (Industry 4.0/5.0) where multi-objective formulations (simultaneous minimization of makespan, energy consumption and tardiness) become more popular, as reported in recent systematic review of flow-shop scheduling literature between 2015-2024 [22].

- Feature Selection and Machine Learning

Swarm intelligence algorithms (such as PSO, GWO, WOA) are frequently used as wrapper methods for feature selection where each individual represents either binary or continuous mask over the feature space optimizing classification accuracy along with the number of selected features.

- Energy Systems and Power Engineering

Parameter optimization in power systems, economic dispatch, and renewable energy forecasting are popular application areas for Q-learning augmented metaheuristics and other adaptive metaheuristics, an area emphasized in recent survey literature as one of the most prominent fields utilizing reinforcement learning-based metaheuristics [21].

- Breadth of Real-world Applications

A comprehensive review published in 2025, which assessed effectiveness of forty-eight metaheuristics developed in 2020-2024 against six major real-world application domains demonstrated substantial difference in algorithm effectiveness depending on the application problem, emphasizing application-oriented choice of algorithms as opposed to benchmark-based rankings [23].

7. Open Challenges

- Premature convergence: Many algorithms, especially swarm intelligence-based ones that exhibit very strong leader-follower schemes, are vulnerable to premature convergence on multi-modal landscapes.
- Parameter sensitivity: Algorithm performance is often parameter-dependent (scheduling for cooling, scaling constants, population sizes), and many comparative papers lack proper documentation of the effort devoted to tuning the algorithm's control parameters, making their fair comparison difficult.
- High-dimensional scalability: Performance drops on high-dimensional benchmarks (hundreds of decision variables) are common, which stimulates the development of high-dimensional benchmark collections like CEC collections.
- Metaphor-based redundancy: A significant number of recently published metaheuristic algorithms have been critically reviewed and found redundant in terms of underlying mechanism despite new biological or physical metaphors [19].
- Reproducibility and fair comparison: Inconsistent choice of benchmark suite, lack of statistical verification (for example, lack of Friedman/Wilcoxon tests), and open-source implementation still pose problems for comparative research.
- Computational overhead for expensive function evaluations: Practical applications (CFD/FEA based objective functions) make the high number of evaluations of objective functions required for population-based searches extremely expensive.

7.1. Directions for Future Research

- LLM-assisted meta-optimization: further research on developing large language models as meta-optimizers that can suggest or tune search operators based on problem formulations and search feedback [24].
- RL-integrated metaheuristics: further incorporation of Q-learning and other reinforcement learning (RL) techniques into metaheuristics to facilitate adaptive tuning of parameters and selection of operators, along with transfer learning.
- Benchmarking and evaluation protocol standardization: wide acceptance of standardized benchmark suites and statistical test procedures and requirement of code and data availability to increase the reproducibility of research results.
- Surrogate-assisted metaheuristics: further development of surrogate-based metaheuristics as an approach to decrease the cost of evaluations of expensive real objective functions.
- Multi-objective and many-objective metaheuristics: further development of multi-objective metaheuristics based on Pareto dominance principles and multi-objective decomposition schemes for solving engineering problems involving conflicting design requirements.
- Explainability and interpretability of metaheuristic search: methodologies of explaining and interpreting the reasons why a certain metaheuristic leads to the convergence point. This is an important topic because metaheuristic-based optimization is increasingly used in engineering design.
- Hybrid quantum-classical metaheuristics: preliminary research on quantum-classical hybrid metaheuristics and quantum-assisted operators.

8. Conclusion

Through this survey, the vast literature on the ever-growing field of metaheuristic optimization has been structured through a classification system comprising five main branches of algorithms – namely, those based on evolutionary processes, swarm intelligence, physics, humans, and hybrids or emerging techniques – as well as the most prominent algorithms within them, always backed up by reliable and DOI-supported references. A generic cycle encompassing both the exploration and exploitation stages of search has been employed as a unifying perspective that would allow drawing connections between seemingly divergent optimization approaches. Comparative tables have been utilized to summarize the main features and benchmarks of algorithms, while the open challenges section has pinpointed the need for addressing the recurring issues of premature convergence, parameterization, scalability, and replicability. The most promising avenues for future work include the incorporation of large language models and reinforcement learning to metaheuristics optimization, as well as benchmarking standards.

Declarations

Author Contributions

Conceptualization: G.A. and W.R.; **Methodology:** G.A. and H.H.; **Software:** G.A.; **Validation:** G.A., W.R., and H.H.; **Formal Analysis:** G.A. and H.H.; **Investigation:** G.A. and W.R.; **Resources:** B.A.B.; **Data Curation:** G.A.; **Writing—Original Draft Preparation:** G.A.; **Writing—Review & Editing:** G.A., W.R., H.H., and B.A.B.; **Visualization:** G.A.; **Supervision:** W.R.; **Project Administration:** W.R.; **Funding Acquisition:** W.R. and B.A.B. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM computing surveys (CSUR), 35(3), 268-308.

[2] Sörensen, K., Sevaux, M., & Glover, F. (2025). A history of metaheuristics. In Handbook of heuristics (pp. 991-1010). Cham: Springer Nature Switzerland.

[3] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

[4] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.

[5] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). ieee.

[6] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

[7] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), 459-471.

[8] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.

[9] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-aided design, 43(3), 303-315.

[10] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

[11] Mirjalili, S. (2015). The ant lion optimizer. Advances in engineering software, 83, 80-98.

[12] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

[13] Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems, 96, 120-133.

[14] Kotenko, I., & Saenko, I. (2015). Improved genetic algorithms for solving the optimisation tasks for design of access control schemes in computer networks. International Journal of Bio-Inspired Computation, 7(2), 98-110.

[15] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872.

[16] Mohammed, H., & Rashid, T. (2023). FOX: a FOX-inspired optimization algorithm: FOX: a fox-inspired optimization algorithm. Applied Intelligence, 53(1), 1030-1050.

[17] Das, S., & Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. IEEE transactions on evolutionary computation, 15(1), 4-31.

[18] Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution–an updated survey. Swarm and evolutionary computation, 27, 1-30.

[19] Dokeroglu, T., Canturk, D., & Kucukyilmaz, T. (2024). A survey on pioneering metaheuristic algorithms between 2019 and 2024. arXiv preprint arXiv:2501.14769.

[20] Ivanovski, T., Brkić Bakarić, M., & Matetić, M. (2025). Recent Advances in Metaheuristic Algorithms. Algorithms, 19(1), 19.

[21] Yang, Y., Gao, Y., Ding, Z., Wu, J., Zhang, S., Han, F., ... & Wang, Y. G. (2024). Advancements in Q-learning meta-heuristic optimization algorithms: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 14(6), e1548.

[22] Zhang, W., Bao, X., Hao, X., & Gen, M. (2025). Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: a state-of-the-arts survey. Frontiers in Industrial Engineering, 3, 1540022.

[23] Hasan, D. O., & Aladdin, A. (2025). Real-world applications of metaheuristic algorithms: a comprehensive review of the state-of-the-art. Diyala Journal of Engineering Sciences, 1-27.

[24] Zheng, Y., Zhang, L., Li, K., Wang, R., Li, W., Zhang, T., ... & Jin, Y. (2026). A survey on large language models driven meta-optimizers for automated intelligent optimization. Artificial Intelligence Review.

[25] Shaban, A. A., Almufti, S. M., & Asaad, R. R. (2025). Metaheuristic Algorithms for Engineering and Combinatorial Optimization: A Comparative Study Across Problems Categories and Benchmarks. International Journal of Scientific World, 11(2), 38-49.

[26] Almufti, S. M., Asaad, R. R., Shaban, A. A., & Marqas, R. B. (2025). Benchmarking metaheuristic algorithms: a comprehensive review of test functions, real-world problems, and evaluation metrics.

[27] Dorigo, M., & Stutzle, T. (2004). Ant Colony Optimization. MIT Press, Cambridge, MA.

