

Review

Rethinking Metaheuristics: Unveiling the Myth of “Novelty” in Metaheuristic Algorithms

Chia-Hung Wang ^{1,2,*} , Kun Hu ¹ , Xiaojing Wu ³ and Yufeng Ou ¹ 

¹ College of Computer Science and Mathematics, Fujian University of Technology, No. 69, Xuefu South Road, Fuzhou 350118, China; 2221308055@mail.fjut.edu.cn (K.H.); 2231308030@mail.fjut.edu.cn (Y.O.)

² Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, No. 69, Xuefu South Road, Fuzhou 350118, China

³ College of Electronics, Electrical Engineering and Physics, Fujian University of Technology, No. 69, Xuefu South Road, Fuzhou 350118, China; wxj2732@163.com

* Correspondence: jhwang728@hotmail.com

Abstract

In recent decades, the rapid development of metaheuristic algorithms has outpaced theoretical understanding, with experimental evaluations often overshadowing rigorous analysis. While nature-inspired optimization methods show promise for various applications, their effectiveness is often limited by metaphor-driven design, structural biases, and a lack of sufficient theoretical foundation. This paper systematically examines the challenges in developing robust, generalizable optimization techniques, advocating for a paradigm shift toward modular, transparent frameworks. A comprehensive review of the existing limitations in metaheuristic algorithms is presented, along with actionable strategies to mitigate biases and enhance algorithmic performance. Through emphasis on theoretical rigor, reproducible experimental validation, and open methodological frameworks, this work bridges critical gaps in algorithm design. The findings support adopting scientifically grounded optimization approaches to advance operational applications.

Keywords: optimization algorithm; optimization; metaphors; structural bias; evaluation of algorithm's innovation

MSC: 90-08; 90C59; 68Wxx



Academic Editor: Ioannis Tsoulos

Received: 18 May 2025

Revised: 23 June 2025

Accepted: 29 June 2025

Published: 1 July 2025

Citation: Wang, C.-H.; Hu, K.; Wu, X.; Ou, Y. Rethinking Metaheuristics: Unveiling the Myth of “Novelty” in Metaheuristic Algorithms. *Mathematics* **2025**, *13*, 2158. <https://doi.org/10.3390/math13132158>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Optimization tasks are ubiquitous across various domains in the real world [1–3]. These problems typically involve identifying the most suitable solution from a vast set of candidates [4,5]. The primary objective in such scenarios is to obtain high-quality solutions while maintaining acceptable computational costs, which necessitates a trade-off among factors such as resource consumption, execution time, and solution accuracy [6].

Over the past few decades, a wide range of metaheuristic algorithms has been developed to effectively tackle real-world optimization problems [7]. These algorithms are designed to improve computational efficiency and reduce operational costs across diverse scientific and engineering applications [8–12]. The advancement of metaheuristics has significantly enriched the toolbox available for solving complex optimization problems [13], offering robust performance in approximating near-optimal solutions, even for large-scale and nonlinear scenarios [14]. Due to their adaptability and effectiveness, metaheuristics

have become indispensable for addressing challenging optimization tasks in contemporary science and engineering [15,16].

Many real-world optimization problems are characterized by high dimensionality, nonlinearity, and multimodality, which render traditional mathematical programming techniques inefficient or infeasible. In such contexts, metaheuristic algorithms provide a viable and efficient alternative by delivering high-quality solutions within reasonable computational constraints [17]. Their demonstrated capability to solve complex, large-scale, and nonlinear problems underscores their superiority over conventional deterministic methods in handling modern optimization challenges [18,19].

Figure 1 illustrates the citation counts of 30 metaheuristic algorithms, highlighting notable trends in their academic impact, technical maturity, and research momentum. Among them, the Genetic Algorithm (GA), with over 105,000 citations, and Particle Swarm Optimization (PSO), with more than 92,000 citations, stand out as the most influential. Their early development and foundational contributions to the fields of evolutionary computation and swarm intelligence have cemented their status as benchmarks in algorithmic performance evaluation. Other algorithms include Simulated Annealing (SA) with nearly 60,000 citations and Differential Evolution (DE) with approximately 37,000 citations.

In recent years, a number of newly proposed algorithms have gained traction, despite their relatively short publication history. The Sparrow Search Algorithm (SSA), Marine Predators Algorithm (MPA), and Aquila Optimizer (AO) [20] are notable examples, each with more than 2,000 citations within just a few years.

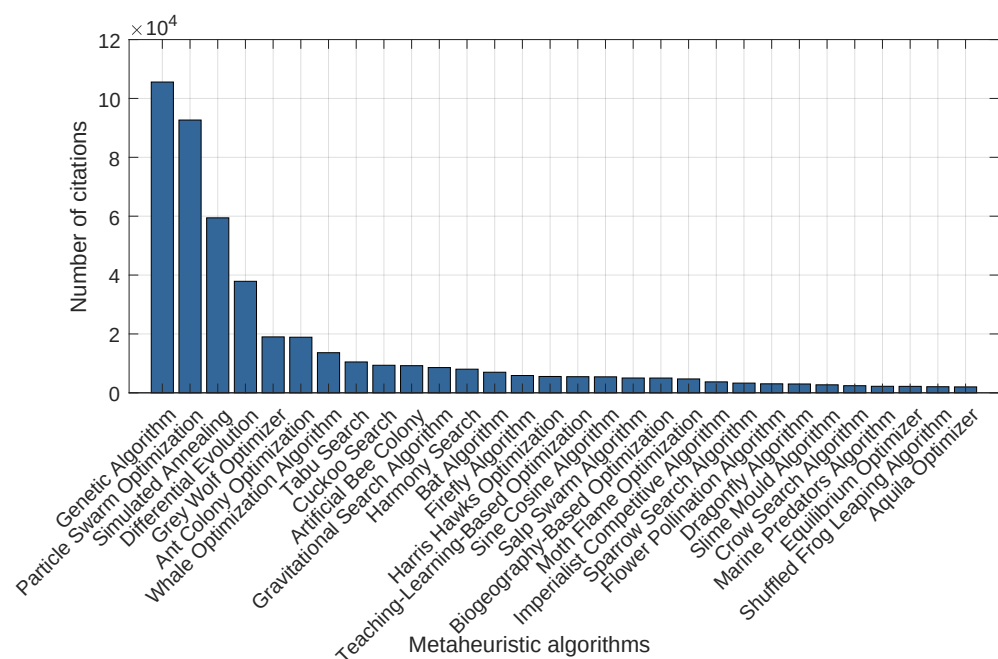


Figure 1. Survey on citations of metaheuristic algorithms.

However, despite their successes, several challenges and limitations continue to hinder their theoretical development and practical adoption [21–26].

Metaheuristic algorithms are not without flaws. Amid the rapid proliferation of numerous “novel” metaheuristic algorithms, researchers are increasingly discovering fundamental weaknesses in many of these methods [27–36]. For example, the well-known phenomenon of “premature convergence” can prevent population-based algorithms from adequately exploring the search space. Premature convergence occurs when the population becomes trapped in local optima early in the search process, leading to stagnation and suboptimal solutions. This issue is exacerbated in high-dimensional and deceptive search

spaces, where the algorithm struggles to balance exploration and exploitation. In fact, only a few algorithms have a general proof of convergence under realistic conditions. For other algorithms, such proofs exist only under rather restrictive hypotheses that are rarely encountered in real-world problems. These deficiencies often stem from the lack of a robust theoretical foundation, making it challenging to analyze their convergence properties, performance guarantees, and stability.

Here are some of the flaws.

- **Ineffective metaphor:** Many algorithms claim novelty through “unique terminology based on new metaphors” that differ only in superficial descriptions rather than substantial mechanisms. These metaphorical approaches—such as biological evolution or physical phenomena—offer intuitive appeal but lack scientific rigor, clear definitions, and explanatory mechanisms. They often obscure the underlying mathematical principles, making it difficult to understand the algorithm’s performance or limitations. In some cases, the metaphor is oversimplified or altered to resemble optimization processes, even when such a link is unsubstantiated. More concerning, many metaheuristic algorithms based on metaphors suffer from a disconnect between conceptualization, mathematical modeling, and implementation, severely impairing transparency and hindering further analysis or practical adoption.
- **Structural bias:** Structural bias in metaheuristic algorithms refers to a tendency for certain regions or patterns in the search space to be favored due to the algorithm’s design. This bias can be unintentionally introduced through initialization, search operators, or parameters, or it can be intentionally embedded to improve performance on specific problems. The intentional introduction of structural bias distorts evaluations, creating misleading conclusions about the algorithm’s capabilities, particularly when tested on simple problems.
- **Old wine in new bottles:** Many metaheuristic algorithms are little more than re-branded versions of classical methods, with only superficial changes. These incremental modifications often lack substantial innovation in the optimization process, leading to a proliferation of algorithms that offer limited practical or theoretical advancement. This oversaturation of similar algorithms complicates performance evaluation and undermines the field’s rigor, diverting attention from addressing core optimization challenges.
- **Controversial experimental comparison:** Fairness in experimental comparisons is essential to ensure the validity and reliability of conclusions. However, selective testing—focusing only on problem instances where an algorithm performs well or comparing against overly simplistic “toy” algorithms—can lead to inflated performance claims. This selective reporting distorts the algorithm’s true capabilities, creating a false impression of superiority and undermining the credibility of experimental results.

In recent decades, there has been an unprecedented surge in the development and publication of new or hybrid bio/nature-inspired metaheuristic algorithms. Figure 2 shows this publication trend (partial statistics are from Reference [37]). The figure reveals a marked surge in the publication volume of metaheuristic algorithms commencing in 2020.

The proliferation of metaphor-centered methods has resulted in a pronounced fragmentation of the literature into numerous small and scarcely distinguishable niches. The frequent use of metaphor-laden language in the presentation of new methods partly contributes to this issue, as it introduces unnecessary barriers to the initial comparison of methodological similarities and differences.

The following key questions remain: Is the proliferation of nature-inspired algorithms truly justified? Are all these emerging approaches essential for solving a wide range of problems? More fundamentally, what does it mean to classify an algorithm as “nature-

inspired”? This paper aims to address these questions, clarifying the relevance, necessity, and theoretical foundations of nature-inspired algorithms in modern problem solving.

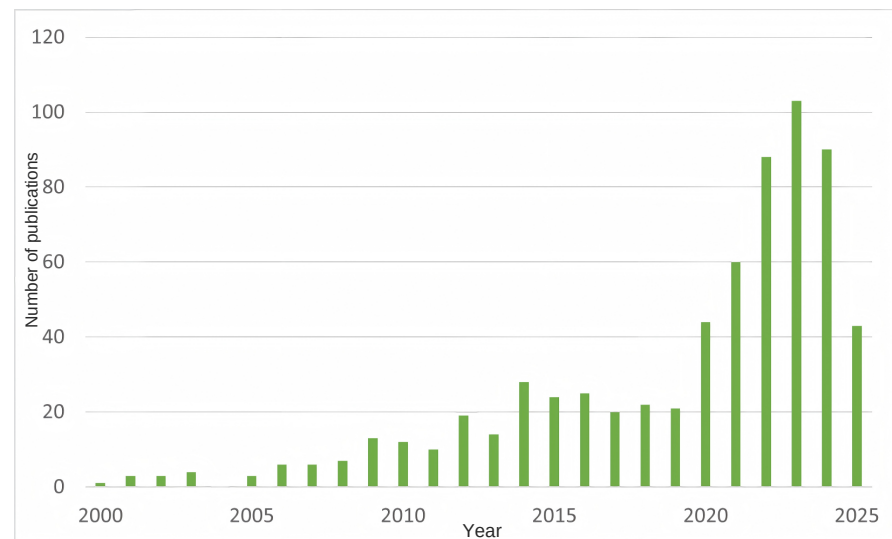


Figure 2. Number of metaheuristic algorithm publications between 2000 and 2025.

In this work, readers are encouraged to reflect on future research directions in metaheuristic optimization and related fields. A comprehensive review is provided of the contributions made by researchers advancing this field [38–40]. The goal is to unify these viewpoints into a holistic, multidisciplinary perspective that will drive the advancement of the field.

The main contributions of this paper can be summarized as follows:

- Critical challenges that have long hindered progress in the field of metaheuristic algorithms were systematically identified and analyzed, including issues such as metaphor-based design, structural bias, and repackaging. These pervasive flaws significantly obstruct the development of more robust, generalizable, and theoretically sound optimization methods, highlighting the urgent need for a paradigm shift in this domain.
- The work of researchers on existing methods for identifying defects was summarized, which provides essential identification techniques for metaheuristic algorithms. These contributions are pivotal in advancing the understanding and mitigation of structural biases in optimization processes, laying the groundwork for more reliable and effective algorithm development.
- A detailed and thorough investigation was performed into the root causes of the negative impacts associated with current metaheuristic approaches, offering valuable insights into their underlying factors.
- An exhaustive summary and analysis of potential research directions in the metaheuristic algorithm domain was presented, complemented by constructive feedback and practical recommendations. These insights are intended to stimulate new avenues of research, address emerging challenges, and promote innovation in the development of more effective and impactful optimization techniques.

The remainder of this paper is organized as follows: Section 2 presents an overview of the definition and evolution of metaheuristic algorithms. Section 3 provides a critical analysis of the challenges faced by these algorithms, including metaphors, structural bias, repackaging, and other key issues. Section 4 then examines the approaches proposed by researchers to identify and address these deficiencies. Subsequently, Section 5 analyzes the root causes of these negative impacts. Section 6 summarizes the proposed suggestions and

recommendations. Finally, Section 7 concludes this review by presenting the key findings and outlining directions for future research.

2. Metaheuristic Algorithms

In his foundational work, Glover [41] introduced the term “metaheuristic,” defining it as a methodology that integrates overarching strategies to guide the search process in optimization problems. Metaheuristic algorithms are often inspired by various natural or artificial systems, leading to the development of a diverse range of techniques. These include swarm intelligence-based algorithms, evolution-based algorithms, physics-based algorithms, and human-based algorithms. Table 1 provides an overview of representative algorithms within each of the four categories, highlighting their core fundamental ideas.

Table 1. Comparison of various categories of metaheuristic algorithms.

Algorithm Type	Classical Algorithm	Fundamental Ideas
Swarm intelligence-based algorithms	Particle Swarm Optimization (PSO)	Collective coordination and information sharing among bird flocks
Evolution-based algorithms	Genetic Algorithm (GA)	Genetic operators (mutation, crossover, and survival), population evolution
Physics-based algorithms	Gravitational Search Algorithm (GSA)	Gravity, mass, acceleration, attraction
Human-based algorithms	Teaching-Based Learning Optimization (TBLO)	Teaching strategies, collaboration, knowledge sharing

Figure 3 presents the classification statistics of 291 metaheuristic optimization algorithms. Among them, Swarm Intelligence-Based Algorithms are the most prevalent, totaling 98 algorithms and accounting for 33.7% of the total. These algorithms are primarily inspired by collective behaviors observed in nature, such as those of ants and birds. Additionally, physics-based algorithms, which draw inspiration from natural physical phenomena like gravity and heat conduction, constitute 23.7% of the total. Human-based algorithms, mimicking human social behaviors, represent 18.2%, while evolutionary algorithms, based on mechanisms of heredity, evolution, and mutation, account for 24.4%. This classification highlights the diverse sources of inspiration in current optimization algorithm research and their varied influences on algorithmic design.

It is obvious from Figure 4 that the distribution of metaheuristic-related publications across major academic publishers reveals that Elsevier is the dominant venue, accounting for 127 publications (43.6% of the total 291). Springer follows with 63 publications (21.6%).

These algorithms typically combine the following two key processes: exploration, which ensures a broad search across the solution space to prevent premature convergence to suboptimal solutions, and exploitation, which focuses the search on promising areas to refine and optimize solutions. Achieving an effective balance between exploration and exploitation is crucial to the success and efficiency of any metaheuristic approach.

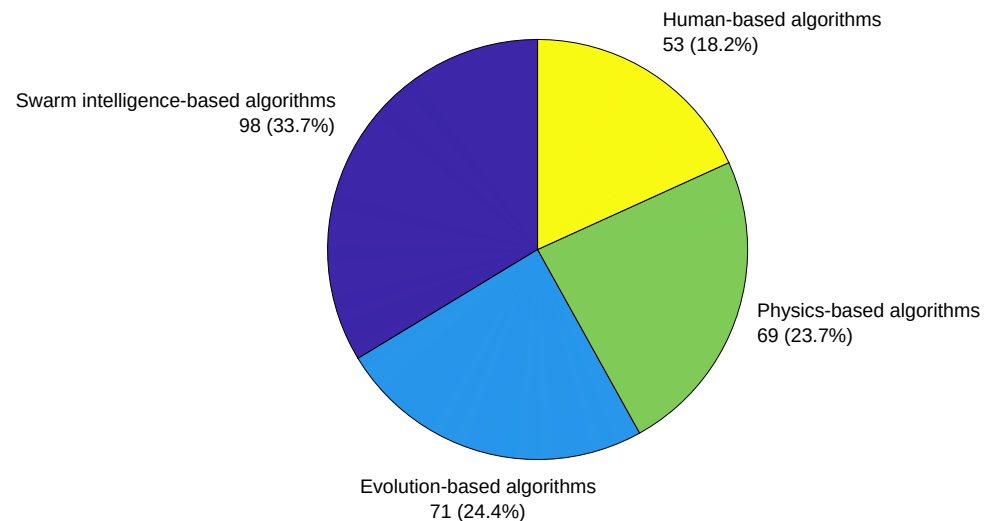


Figure 3. Distribution of algorithm categories.

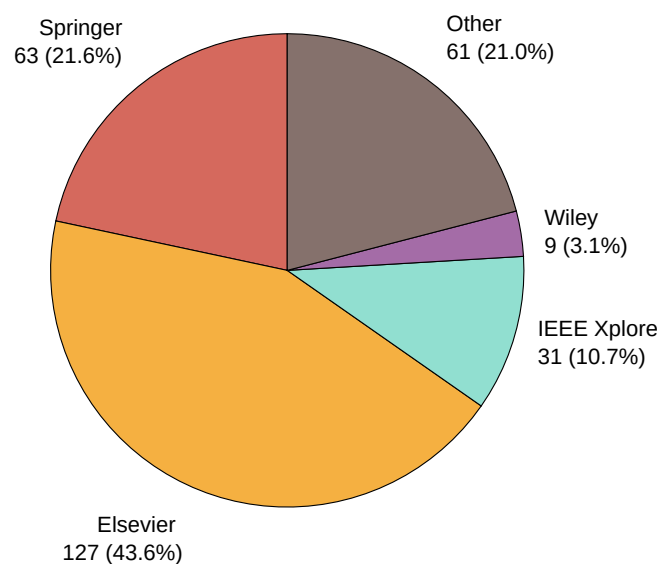


Figure 4. Publication venues with type of metaheuristic publications.

Many metaheuristic algorithms maintain and manipulate a diverse set of candidate solutions. These algorithms rely on fitness functions to evaluate the quality of solutions and iteratively sample the solution space to improve upon the best solutions identified so far. Randomness is often incorporated into the search process to enhance exploration, making these methods particularly robust for solving non-continuous, noisy, or dynamic optimization problems. Unlike gradient-based methods, metaheuristic algorithms do not require derivatives, making them versatile for tackling real-world problems with complex and irregular landscapes [42–44].

2.1. Swarm Intelligence-Based Algorithms

Swarm intelligence is inspired by the communication and collaborative behaviors exhibited by social animals, such as those in a herd, during the optimization process. The core principle of swarm algorithms is based on the collective actions of animals and insects in groups. In particular, the behaviors of species like ants, bees, and other social organisms serve as models for these algorithms. A key feature of swarm-based methods is the exchange of information among agents, which significantly impacts their movement

and decision-making. By regulating this information flow, an effective balance can be achieved between exploration (searching unexplored areas of the solution space) and exploitation (focusing the search on promising regions), similar to how animals adjust their foraging strategies. This interactive process enables the emergence of an optimal or near-optimal solution through the collective intelligence of the swarm.

The most popular algorithm in this category is the Particle Swarm Optimization (PSO), which is one of the pioneering algorithms in this branch. In the PSO algorithm, birds typically rely on collective behavior while searching for food. PSO is widely used in healthcare, environment, industry, commerce, smart cities, etc. [45]. Although PSO is one of the most respected swarm-based algorithms in the literature, it still suffers from the problem of premature convergence [46]. The Ant Colony Optimization (ACO) is another widely studied swarm intelligence algorithm. In ACO, the ant colony relies on a chemical substance known as pheromone to facilitate communication and cooperation, both among ants and between ants and their environment. Pheromones enable indirect communication within the colony and influence the foraging behavior of other ants, ultimately helping the colony discover the shortest path from the nest to the food source through cooperation. ACO is robust against noise and local optima due to its probabilistic solution construction process. However, its performance is highly sensitive to parameters such as the pheromone evaporation rate and heuristic information. Improper parameter settings may lead to premature convergence or stagnation [47].

2.2. Evolution-Based Algorithms

Evolutionary Algorithms (EAs) are a class of optimization techniques inspired by Darwin's theory of evolution. According to Darwin's theory, variation within a species occurs randomly, and natural selection favors the survival of the fittest individuals. EAs adopt this principle, simulating processes such as selection, mutation, and reproduction to explore and exploit the search space. By iteratively evolving a population of candidate solutions, these algorithms aim to identify near-optimal solutions. Through mechanisms analogous to natural evolution, such as survival of the fittest and the introduction of genetic diversity, EAs are capable of navigating complex and high-dimensional search spaces to find robust solutions.

The most popular EA is the Genetic Algorithm (GA), one of the earliest algorithms in this domain. It simulates the concepts of Darwinian evolution, using three primary operators—crossover, mutation, and selection—to construct the algorithm. The GA iteratively applies these operations to a randomly generated population of candidate solutions. Key factors influencing the algorithm's performance include the crossover rate, mutation rate, objective function, and population size. Premature convergence is a prevalent issue in GAs, which can result in the loss of genetic diversity, thereby hindering the identification of optimal solutions. This problem occurs when the population converges too early to a suboptimal solution, preventing the exploration of the solution space. To mitigate this, it is crucial to select appropriate rates for crossover and mutation operators, as these parameters directly influence the algorithm's ability to maintain genetic diversity and explore the search space effectively [48]. Another well-known EA is the Differential Evolution (DE). In DE, the iteration begins with a randomly generated initial population. New individuals are generated by summing the vector difference of any two individuals in the population with a third individual. The newly created individual is then compared with the corresponding individual in the current population. If the new individual has better fitness, it replaces the old individual in the next generation; otherwise, the old individual is retained. This process mimics the natural principle of "survival of the fittest", where through continuous

evolution, superior individuals are preserved while inferior ones are eliminated, guiding the overall search toward the optimal solution.

2.3. Physics-Based Algorithms

Physics-based algorithms replicate physical laws and principles during the optimization process to identify optimal solutions. These methods draw inspiration from fundamental physical concepts underlying natural phenomena, allowing a population of randomly generated search agents to navigate and explore the solution space according to physical principles.

Several well-known algorithms fall under this category. Simulated Annealing (SA), for instance, is inspired by the metallurgical annealing process, where a material undergoes controlled cooling and crystallization. This algorithm solves optimization problems by emulating this cooling process, making it particularly effective for problems with complex or multi-modal landscapes, where multiple local optima exist. SA is efficient, easy to implement, and theoretically sound, but it suffers from a slow convergence rate [49]. Another example is the Lightning Search Algorithm (LSA), which takes inspiration from the erratic and powerful nature of lightning strikes. LSA uses the unpredictable and dynamic behavior of lightning to explore the search space, seeking optimal solutions through a process that mimics the natural phenomenon of lightning's trajectory. The LSA offers several advantages, including fast convergence, effective global exploration, and the ability to handle a wide range of optimization problems with minimal parameter tuning. It is highly adaptable, scalable, and robust, producing high-quality solutions efficiently while being less dependent on initial solutions and less prone to getting stuck in local optima. However, its main drawbacks include susceptibility to premature convergence, a lack of a theoretical convergence framework, and variability in probability distribution across generations [50].

2.4. Human-Based Algorithms

Human-based algorithms are a class of algorithms that leverage human intelligence for computation with the goal of simulating and drawing inspiration from human natural behaviors, cognitive processes, and social patterns. By emulating the flexibility, intuition, and creativity exhibited by humans in problem solving, these algorithms are able to effectively address complex optimization tasks. Specifically, human-based algorithms not only model individual decision-making processes but also often incorporate mechanisms of social interaction and collective collaboration, thereby tapping into the collective intelligence found within human societies.

The Teaching-Based Learning Optimization (TBLO) algorithm is inspired by the dynamics between a teacher and students. It combines the roles of teacher and student to explore the solution space and identify optimal outcomes. The algorithm utilizes strategies such as exploration, exploitation, and knowledge transfer, with the "teacher" guiding the process to help "students" converge on superior solutions. By incorporating these strategies, TBLO iteratively refines candidate solutions while balancing exploration and exploitation. TBLO was successfully applied to photovoltaic model parameter estimation [51]. The Gaining Sharing Knowledge-Based Algorithm (GSKA) is a metaheuristic inspired by human knowledge sharing and collaborative learning. It emphasizes cooperation and information exchange to address optimization problems. By fostering the sharing and acquisition of knowledge, GSKA strengthens the search process, enabling effective exploration of the solution space and convergence toward optimal solutions. This algorithm is grounded in the principle that collaboration and knowledge transfer can significantly en-

hance problem-solving efficiency. This method is applied to fog computing multi-objective task scheduling [52].

3. Problems with Metaheuristic Algorithms

This section systematically identifies and analyzes critical issues that have long hindered the progress of metaheuristic algorithms. Key challenges including metaphor-based design, structural bias, and repackaging are examined in detail. These problems have not only constrained the efficiency and applicability of metaheuristic algorithms but also contributed to stagnation in the field. The identified deficiencies in metaheuristic algorithms are summarized in Table 2.

Table 2. Summary of flawed algorithms.

Ref.	Algorithm	Metaphors	Structural Bias	Repackaging	Others
[53–55]	Harmony Search (HS)	✓	×	A simplification of evolutionary strategies	-
[56]	Intelligent Water Drops (IWD) Algorithm	✓	×	A particular instantiation of ACO	-
[57]	Biogeography-Based Optimization (BBO)	✓	×	A generalization of GA	-
[58]	Firefly Algorithm (FA)	✓	×	-	-
[59]	Gravitational Search Algorithm (GSA)	✓	✓	-	-
[60]	Cuckoo Search (CS)	✓	×	Variant type of differential evolution	Quite some differences between description and implementation
[58,61]	Bat Algorithm (BA)	✓	✓	-	-
[62]	Teaching Learning-based Optimization (TLO)	✓	✓	-	-
[63]	Black Hole Optimization (BHO)	✓	×	A simplified version of PSO	-
[64]	Coral Reef Optimization (CRO)	✓	✓	Deficient mixtures of different evolutionary operators	-
[58,62,65–67]	Grey Wolf Optimizer (GWO)	✓	✓	-	-
[58]	Antlion Optimizer (ALO)	✓	×	-	-
[62]	Elephant Herding Optimization (EHO)	✓	✓	-	-
[58]	Moth-Flame Algorithm (MFA)	✓	×	-	-
[62]	Wind Driven Optimization (WDO)	✓	✓	-	-
[58,62]	Whale Optimization Algorithm (WOA)	✓	✓	-	-
[68]	Grasshopper Optimization Algorithm (GOA)	✓	×	A derivative of PSO	-
[62,69]	Sine Cosine Algorithm (SCA)	✓	✓	-	-
[70]	Raven Roost Optimization (RRO)	✓	✓	A special case of PSO	The inherent bias towards its starting point
[62]	Wildebeest Herd Optimization (WHO)	✓	✓	-	-
[62]	Henry Gas Solubility Optimization (HGSO)	✓	✓	-	-

Table 2. Cont.

Ref.	Algorithm	Metaphors	Structural Bias	Repackaging	Others
[62]	Butterfly Optimization Algorithm (BOA)	✓	✓	-	-
[62]	Harris Hawks Optimization (HHO)	✓	✓	-	-
[62]	Naked Mole-Rat Algorithm (NMRA)	✓	✓	-	-
[62]	Nuclear Reaction Optimization (NRO)	✓	✓	-	-
[62]	Pathfinder Algorithm (PA)	✓	✓	-	-
[71]	Marine Predators Algorithm (MPA)	✓	✓	-	-
[62]	Tunicate Swarm Algorithm (TSA)	✓	✓	-	-
[62]	Sparrow Search Algorithm (SSA)	✓	✓	-	-
[62]	Slime Mould Algorithm (SMA)	✓	✓	-	-
[62]	Bald Eagle Search (BES)	✓	✓	-	-
[62]	Artificial Ecosystem-based Optimization (AEO)	✓	✓	-	-
[62]	Equilibrium Optimizer (EO)	✓	✓	-	-
[62]	Gradient-Based Optimizer (GBO)	✓	✓	-	-
[62]	Marine Predators Algorithm (MPA)	✓	✓	-	-
[62,72]	Chimpanzee Optimization Algorithm (ChOA)	✓	✓	A variant of PSO	-
[64]	Black Widow Optimization (BWO)	✓	✓	Deficient mixtures of different evolutionary operators	-
[62,73]	Arithmetic Optimization Algorithm (AOA)	✓	✓	-	Design artificially improves accuracy over standard benchmarks
[62]	Runge Kutta Optimizer (RKO)	✓	✓	-	-
[62]	Chaos Game Optimization (CGO)	✓	✓	-	-
[62]	Aquila Optimization (AO)	✓	✓	-	-
[62]	Battle Royale Optimization (BRO)	✓	✓	-	-
[62]	Hunger Games Search (HGS)	✓	✓	-	-
[62]	Dandelion Optimizer (DO)	✓	✓	-	-
[62]	Komodo Mlipir Algorithm (KMA)	✓	✓	-	-
[62]	Mountain Gazelle Optimizer (MGO)	✓	✓	-	-

3.1. Metaphors

One of the major factors impeding the development of metaheuristic algorithms is the misuse of metaphors [74]. Early research in the field was cautious in applying metaphorical inspiration, often treating it as a secondary heuristic. However, as the field expanded, metaphors—particularly those drawn from biological, physical, and social systems—began to dominate algorithm design, shifting from inspiration to central justification.

This metaphor-driven trend has led to a surge of algorithms that emphasize thematic novelty over methodological substance. Many such algorithms introduce metaphor-specific terminology, while overlooking algorithmic soundness, theoretical grounding, or practical performance. Often, the modifications are limited to superficial naming or trivial operational changes, without genuine algorithmic innovation.

A growing concern is the inconsistent or weak connection between the metaphor and the actual algorithmic structure. Mathematical models derived from metaphors are frequently ad hoc and fail to capture the essence of the original phenomena. This mismatch undermines both the interpretability and robustness of the algorithms. In many cases, key theoretical aspects such as convergence analysis or complexity bounds are omitted, leaving the algorithm's efficacy poorly justified.

Additionally, the pursuit of attention-grabbing metaphors—ranging from animal behavior to natural disasters—has contributed to semantic inflation in the literature, making it increasingly difficult to discern truly innovative contributions. While these metaphor-based approaches may appear original, they often obscure fundamental limitations and detract from more rigorous, mechanism-driven developments in the field.

As extensively discussed by Sörensen [75], while metaphors are rarely essential for describing metaphor-based algorithms, their significance is often exaggerated by the authors. Unfortunately, the added value of employing metaphors is frequently unclear, and in some cases, their use can be misleading. Without a well-defined taxonomy to categorize the various metaphor-based algorithms, it remains difficult to assess whether these “novel” algorithms genuinely offer any innovation. As a result, formal or empirical evaluations of these algorithms must currently be carried out on a case-by-case basis, hindering the ability to draw general conclusions or identify overarching trends within this field.

Deng and Liu [72] performed a comprehensive, component-based analysis of the widely referenced Chimpanzee Optimization Algorithm (ChOA). Their study involved deconstructing ChOA into its basic components and drawing comparisons with established algorithms, such as Particle Swarm Optimization (PSO). The analysis revealed that the foundational concepts of ChOA are not new to the metaheuristic literature; rather, its supposed novelty stems primarily from the introduction of new terminology inspired by metaphors. Essentially, their findings indicate that ChOA is, at its core, a variant of PSO, distinguished mainly by metaphor-driven nomenclature rather than by any significant methodological advancements.

3.2. Structural Bias

Structural bias refers to the non-uniform search behavior of metaheuristic algorithms in the solution space, caused by their internal structures or design rules. As a result, certain regions of the search space are over-explored, while others are neglected, effectively guiding the operators to specific locations within the search domain. This leads to a non-uniform distribution of the search process across the solution space [76].

During the algorithm design process, developers may deliberately introduce search strategies that are strongly correlated with specific fitness functions in order to achieve artificially high precision on these functions. However, such precision is often limited and fails to generalize to other optimization problems or slightly shifted benchmark functions. This phenomenon highlights the algorithm's inherent lack of robustness and generalizability, significantly undermining its practical applicability and broader utility [77].

Furthermore, this deliberate fitting obscures the algorithm's limitations, creating a superficial impression of effectiveness while lacking true theoretical innovation. Rather than advancing deeper insights into optimization mechanisms, such approaches remain at

the surface level, focusing solely on narrow, task-specific performance rather than achieving meaningful, universal progress in algorithm design.

Castelli et al. [78] identified several limitations in the Salp Swarm Optimization (SSO) algorithm and proposed an improved variant, ASSO, to overcome these challenges. They observed that SSO's performance becomes inconsistent when the decision space boundaries are altered. In particular, even a slight shift in the decision space, when combined with a large constant, can disrupt the algorithm's search process. Additionally, the position update formula for the followers was found to be flawed, deviating from the expected precision according to Newton's law. Simulations on standard benchmark functions further revealed a structural bias in SSO due to its design. Based on these insights, ASSO was developed to address these issues, aiming to improve the algorithm's robustness and overall performance.

Deng and Liu [73] performed an in-depth analysis of the search mechanism in the Arithmetic Optimization Algorithm (AOA). Their findings indicated that the quality of solutions produced by AOA is highly sensitive to the problem boundaries, which leads to a significant structural bias in the algorithm. This bias artificially enhances the algorithm's performance on standard test functions, resulting in high precision. However, when applied to shifted benchmark problems, AOA's performance significantly deteriorates, demonstrating its limited robustness and generalizability across different problem settings.

Kudela [62] proposed a straightforward procedure to identify optimization methods with center bias and applied this approach to analyze 90 methods introduced over the past three decades. The study revealed that 47 of these methods exhibit center bias, a phenomenon characterized by algorithms' performance being overly reliant on the problem's central region. The analysis highlighted that center bias is a relatively recent trend, with its earliest occurrences traced back to methods proposed between 2012 and 2014. However, the adoption of center bias has grown rapidly in the past five years, with a substantial majority of newly proposed methods now exhibiting this issue. In contrast, the number of new methods without center bias has only slightly increased over the same period, indicating a troubling imbalance in the development of optimization techniques. Notably, Kudela's investigation focused exclusively on the baseline or original versions of these methods, excluding the "improved" or "enhanced" variants that are being published at an accelerating pace.

3.3. Repackaging

The phenomenon of "repackaging" is another widespread issue in metaheuristic algorithm research. Numerous purportedly novel algorithms have been introduced under new names, often drawing on natural or social metaphors. However, upon closer inspection, their core mechanisms exhibit minimal deviation from well-established algorithms such as PSO and GA. These efforts represent cosmetic reformulations rather than substantive innovations.

Such practices contribute little to advancing optimization methods and may even introduce unnecessary complexity. By prioritizing superficial differentiation—such as renaming variables or slightly modifying update rules—over genuine algorithmic contributions, repackaged algorithms dilute the clarity and rigor of the field.

There is an increasing number of documented cases where newly proposed algorithms essentially replicate the structure or behavior of existing methods without introducing novel search dynamics, convergence strategies, or theoretical insights. This raises legitimate concerns about originality, scientific value, and the proper standards for methodological contribution in metaheuristic optimization.

Weyland [54] performed a critical evaluation of the Harmony Search (HS) metaheuristic framework, rigorously proving that HS is, in fact, a specific instance of Evolution Strategies (ES). His analysis revealed that HS does not introduce any novel capabilities or performance improvements; rather, its performance is inherently limited to the potential of ES. This conclusion highlights that HS does not expand the horizons of existing optimization methods but instead repackages a well-established approach under a different conceptual guise. Concerns regarding the originality of Harmony Search were first formally articulated in a 2010 study [53], which identified HS as a special case of ES [79], a heuristic that predates HS by several decades. Notably, similar results were independently corroborated in a study two years later [80]. Further conceptual and empirical issues were identified in investigations of Harmony Search's application to the design of water distribution networks [81], suggesting that these concerns may extend to a wide range of other applications of the HS algorithm.

Harandi et al. [68] conducted an in-depth analysis of the Grasshopper Optimization Algorithm (GOA), critically examining its underlying concepts and drawing comparisons with various versions of PSO. The findings reveal that, contrary to claims of novelty, GOA is not a fundamentally new algorithm but rather a derivative of PSO, sharing significant similarities in its core mechanisms. Furthermore, the study highlights a critical inefficiency in GOA's design, outlined as follows: the algorithm requires calculating the distances between all grasshoppers and the target grasshopper during each iteration. This operation significantly increases its computational cost compared to PSO, which typically adopts more efficient mechanisms for position updates. This analysis underscores the need for a more nuanced evaluation of purportedly novel algorithms, particularly with respect to their computational efficiency and their genuine contributions to the field.

3.4. Other Issues

3.4.1. Inconsistency Between Theory and Code Implementation

Camacho-Villalón et al. [60] identified discrepancies between the algorithm proposed for Cuckoo Search and the publicly available implementation provided by the authors. Furthermore, both the proposed algorithm and the implementation deviate from the metaphor of cuckoos that was intended to inspire its design. Of particular concern is the fact that the core concepts underlying Cuckoo Search were not novel but were instead originally introduced by the evolutionary computation community—many of them over three decades prior to the first publication of Cuckoo Search. Camacho-Villalón et al. presented substantial evidence showing that Cuckoo Search is essentially equivalent to the $(\mu + \lambda)$ -evolutionary strategy [82], incorporating recombination mechanisms similar to those proposed for differential evolution [83]. This analysis calls into question the originality of Cuckoo Search and suggests that it is largely a reconfiguration of existing evolutionary techniques rather than a truly novel approach.

3.4.2. Controversial Empirical Comparison

Numerous studies have underscored the contentious and often inconsistent nature of empirical comparisons between metaheuristic algorithms. For example, Derrac et al. [84] noted in their research on the CEC2011 real-world problems [85] that the rankings of metaheuristic algorithms can fluctuate significantly depending on whether statistical tests are used to evaluate the significance of differences in final results or to focus on variations in convergence properties. Similarly, Veček et al. [86] highlighted the crucial role that the choice of statistical tests plays in shaping the final assessment of algorithm performance. In a study by Dymond et al. [87], it was emphasized that the selection of control parameter values for various metaheuristic algorithms is strongly influenced by the number

of function calls used during testing. Liao et al. [88] demonstrated that the performance advantages of newer artificial bee colony (ABC) variants over older methods were evident only in simpler, low-dimensional problems, with the performance gap narrowing in more complex, high-dimensional settings. Furthermore, Draa [89] showed that the Flower Pollination Algorithm, a relatively recent approach, performs weaker than classical algorithms across a wide range of problems, challenging some of the claims in the literature. Finally, Piotrowski [90] pointed out significant discrepancies in algorithm rankings, and Piotrowski and Napiorkowski [91] illustrated how rankings can change when comparing algorithms aimed at finding the minimum versus the maximum of the same functions. These findings highlight the complexities and variability in evaluating the effectiveness of metaheuristic algorithms, suggesting that algorithm comparisons must carefully consider testing methodologies and problem characteristics.

The previous discussion highlights that the methodology used in comparative studies can have a profound impact on the assessment of competing metaheuristic algorithms, sometimes distorting the results to support predetermined conclusions. This raises significant concerns about the reliability of research on metaheuristic algorithms, an issue that has been frequently addressed in the literature. Mernik et al. [92] reviewed all studies on the artificial bee colony (ABC) algorithm published up to August 2013, including the original ABC algorithm introduced in [93]. Their analysis revealed that, depending on the evaluation criteria applied, between 40% and 67% of the studies produced unreliable or incorrect results, primarily due to improper comparisons between different algorithms. Furthermore, some studies [94] introduced both novel algorithms and new methods for comparing algorithms, raising questions about the objectivity and accuracy of their evaluations. Crepinšek et al. [95] provided a detailed discussion on how manipulating the number of experimental replications can lead to unfair comparisons between metaheuristic algorithms. This issue of insufficient rigor in metaheuristic algorithm comparisons, which can result in misleading conclusions, has been widely addressed in other studies as well [28,96–98]. While similar problems have been observed in other scientific disciplines [99,100], this should not be viewed as a justification for overlooking the importance of methodological rigor in evaluating algorithms.

4. Identifying Defects

This section summarizes the contributions of researchers in identifying flaws within metaheuristic algorithms. Table 3 presents a concise comparative analysis of these methodologies. These studies have introduced critical identification techniques, forming a foundation for uncovering and addressing structural biases in optimization processes. Collectively, researchers have advanced the understanding of the so-called “novelty” myth surrounding metaheuristic algorithms, challenging the assumption that such algorithms are inherently “novel”. Their efforts represent a significant step toward fostering the healthy development of the field, paving the way for more reliable and efficient algorithms while supporting its continued advancement.

4.1. Generalized Signature Test

The signature test introduced by Clerc [101] is a commonly used method for detecting algorithmic bias. However, it has certain limitations and does not provide a comprehensive evaluation of algorithm performance. To overcome these limitations, Rajwar and Deep [102] proposed an enhanced approach known as the Generalized Signature Test (GST), which offers a more robust way to detect and quantify bias in population-based stochastic optimization algorithms.

Table 3. Summary of detection methods.

Test Method	Type	Advantages	Disadvantages	Computational Cost
Signature test	Visual Test	Simple, provides clear visual representation of biases, and suitable for 2D problems.	Subjective, limited to 2D problems, and cannot fully eliminate landscape bias in greedy algorithms.	Low; suitable for small-scale 2D functions.
Generalized signature test	Grid-based Test	Flexible, scalable for high-dimensional data, and can detect various types of bias more comprehensively.	High computational complexity, difficult to implement and interpret, and prone to overfitting in high-dimensional scenarios.	High; increases exponentially with problem dimensionality.
Shifted benchmark function	Numerical Analysis	Effectively reveals structural biases such as central bias and allows observation of algorithm behavior under shifted problem landscapes.	Relies on prior knowledge of biased regions and may be ineffective in detecting boundary biases or biases distributed across multiple regions.	Moderate; requires multiple evaluations under varied shift configurations.
Parallel coordinates test	Visual Test	Visualizes high-dimensional data, identifies patterns, and compares multiple variables.	Cluttered with many dimensions, subjective interpretation, and overlapping lines reduce clarity.	Moderate to high; depends on dimensionality and data volume.
BIAS toolbox	Statistical Test	Systematic bias analysis, quantifies algorithm behavior, and supports benchmarking.	Requires parameter tuning, limited to specific problem types, and may not capture all biases.	Moderate; may vary based on sample size and statistical complexity.
Region scaling	Visual Test	Allows investigation of algorithm scalability and behavior under varying landscape resolutions.	Improper scaling can introduce artificial difficulty or obscure algorithm strengths.	Low to moderate; scaling and visualization are relatively lightweight.

The GST assesses population diversity by dividing the solution space into multiple hypercubes based on the population size. The process begins by generating an initial population of size N , uniformly distributed within the search region J . Upon executing algorithm X once, it generates N solutions. The search region J is then divided into N equally sized hypercubes. At each iteration i , the test calculates the number of empty hypercubes (E_i), serving as an indicator of population density within the search space. For a population size N and a problem with D dimensions, the signature factor $\eta_{X(i,N,D)}$ for algorithm $\mathcal{A}$ at iteration i is defined as follows:

$$\eta_{X(i,N,D)} = \frac{\text{Number of empty hypercubes by the algorithm } \mathcal{A}}{\text{Total number of hypercubes } (N)} \quad (1)$$

The signature factor is a critical metric for identifying bias in metaheuristic algorithms. It measures how the population of an algorithm is distributed within the search space, represented by the value $\eta_{X(i,N,D)}$. Given the stochastic nature of these algorithms, the η values can fluctuate across different runs, even when the iteration and population parameters are kept constant. As a result, performing multiple independent runs is necessary to achieve statistically reliable results.

GST offers a fundamental understanding of structural bias and describes it in greater detail. It provides an efficient and straightforward approach for evaluating algorithmic structural bias, enabling in-depth analysis and facilitating tailored investigations into biased search behaviors.

The GST is particularly effective in identifying bias within well-established algorithms and evaluating emerging algorithms during their early stages of development. By uncovering inherent limitations and structural tendencies, GST fosters a deeper understanding of algorithmic behavior, ultimately contributing to the design of robust and unbiased optimization algorithms.

4.2. Shifted Benchmark Function

Kudela [62] proposed a shift operation for evaluating optimization methods, where a benchmark function $f(x)$ is transformed into $f(x + s)$ by applying a predetermined shift vector s . The purpose of this operation is to test the sensitivity of optimization methods to changes in the problem's reference point. Ideally, for methods without center bias, a "small" shift s (e.g., 10% of the problem's range) should not lead to significant deviations in the algorithm's behavior, as the shifted and unshifted problems remain inherently similar.

In Kudela's study, the performance on the shifted problem is compared to that on the unshifted problem for individual benchmark functions. This comparison is quantified using the ratio of performance metrics between the shifted and unshifted problems. For methods without center bias, this ratio is expected to remain close to 1, indicating consistent performance across both scenarios. Conversely, for methods with center bias, this ratio tends to be significantly greater than 1, reflecting a notable performance degradation on the shifted problem.

To quantify the impact of center bias, Kudela calculated the geometric mean of these ratios across various benchmark functions. If the geometric mean exceeds 10 (indicating that performance on unshifted problems is roughly an order of magnitude better than on shifted ones), it serves as evidence of the presence of a center bias operator in the method. This analysis provides a straightforward yet effective approach to identifying and quantifying center bias in optimization algorithms.

4.3. Parallel Coordinates Test

Kononova et al. [103] proposed the use of Parallel Coordinates [104] as a method for visualizing and identifying structural bias in high-dimensional search spaces. A similar approach was previously employed by Cleghorn and Engelbrecht to analyze Particle Swarm Optimization (PSO) [105].

In this approach, each solution in the population is represented by its values across multiple dimensions and plotted using parallel coordinates. This allows for the visualization of patterns, clusters, or potential biases towards specific regions within the search space. By examining the distribution of lines across the axes, researchers can detect whether the algorithm displays any structural bias. For instance, if a significant number of lines converge towards particular values or ranges in certain dimensions, regardless of the objective function's landscape, it may indicate a bias. Kononova et al. [103] observed a mild bias in Genetic Algorithms (GAs) and Differential Evolution (DE) towards the center of the search space.

The main advantage of this technique is its ability to provide a clear visual representation of an algorithm's behavior in high-dimensional spaces, helping to identify patterns, trends, and potential biases during both the exploration and exploitation stages. However, as the dimensionality of the problem increases, parallel coordinate plots can become congested and harder to interpret, potentially limiting their effectiveness in very high-dimensional spaces. Additionally, the interpretation of these plots is highly dependent on the scaling and normalization of the axes, with different scaling choices leading to varying visual outcomes. Hence, it is crucial to maintain consistency in scaling across comparisons. To complement the insights derived from visual analysis, additional numerical techniques are often necessary.

4.4. BIAS Toolbox

Vermetten et al. [106] extended the work of Kononova et al. [103] by introducing a comprehensive toolbox called BIAS (Bias in Algorithms, Structural), specifically designed to detect and analyze biases in metaheuristic algorithms. The BIAS toolbox leverages

a Random Forest model to classify different types of biases within algorithms, offering deeper insights into how these biases influence algorithmic performance. By integrating machine learning techniques, such as Random Forest, the toolbox can accurately identify and categorize biases based on the algorithm's input data and behavior. Focusing on structural and procedural biases, BIAS equips researchers with valuable information to improve the fairness, robustness, and reliability of metaheuristic algorithm design and evaluation. Furthermore, this approach highlights the potential of incorporating advanced data-driven methodologies into the analysis of algorithmic systems, opening the door for more systematic and evidence-based research.

4.5. Region Scaling

Gehlhaar [107] proposed the “Region Scaling” technique as an innovative method for initialization in evolutionary algorithms. However, when considered in isolation, this method introduces two potential biases. First, operators such as the intermediate crossover, commonly used in evolutionary strategies, can lead to offspring being positioned near the center of the initialization region. This occurs because recombining parents from opposite sides of the origin typically results in offspring close to the center of the search space. Second, since the exact location of the optimal solution is usually unknown, no initialization method can guarantee that the optima will be adequately covered within the search space. To address these biases, the study recommends initializing the population in regions that explicitly exclude the optima during testing. This strategy allows for a more rigorous validation of results derived from symmetric initialization schemes, providing more reliable insights into algorithm performance.

5. Analysis of the Causes of These Negative Effects of Metaheuristic Algorithms

Over the past fifty years, nature has inspired the development of numerous heuristic optimization algorithms [108,109]. However, the question of whether inspiration is necessary to create an efficient optimization approach remains unresolved [110].

The proliferation of negative impacts in the metaheuristic algorithm literature is a complex, multi-faceted issue involving various stakeholders with diverse motivations. Nonetheless, several factors can be identified as potential contributors to this phenomenon.

- **Over-reliance on simplified metaphors:** Many metaheuristic algorithms are inspired by natural phenomena like bird flocking or species evolution. While these metaphors offer intuitive insights, they often oversimplify complex real-world problems. This can lead to overlooking nonlinear relationships, complex dynamics, and systemic constraints. Consequently, algorithm designs may lack thorough analysis of practical contexts, relying more on metaphors than rigorous theoretical research. Excessive reliance on such metaphors can also neglect the algorithm's core principles, resulting in poor performance on complex problems.
- **Overemphasis on performance:** Performance is often prioritized in metaheuristic research, with a focus on optimizing speed and accuracy. However, this emphasis can overshadow important factors such as stability, interpretability, and applicability. Selective experimental designs—choosing specific test problems or parameter configurations—can introduce bias. Additionally, an excessive focus on performance may cause algorithms to become stuck in local optima or overfit, limiting their effectiveness on more complex, diverse real-world problems.
- **Low-risk, low-cost “repackaging”:** The “repackaging” phenomenon involves minor modifications to existing algorithms, such as parameter tweaks, presented as new methods. This low-cost, low-risk approach allows researchers to bypass the challenges

of designing new algorithms, making it a shortcut to publishing papers. However, this method often lacks innovation and does not contribute to significant advancements in the field. Table 4 summarizes the key differences between genuinely novel metaheuristic algorithms and those that merely represent repackaged variants. As shown, repackaged algorithms tend to reuse existing structures with minimal modification, offering limited theoretical or practical contributions to the field.

- **Perverse incentive structure:** Campelo and Aranha [111] highlighted perverse incentives within academia, where the “publish or perish” mentality prioritizes short-term achievements over deep scientific inquiry. This environment rewards substandard methodologies, fostering what has been termed the “natural selection of bad science” [112]. Publishing metaphor-based methods is perceived as a relatively low-effort, low-risk endeavor with potentially high rewards.

Table 4. Comparison between genuinely novel and repackaged metaheuristic algorithms.

Criteria	Genuinely Novel Algorithms	Repackaged Algorithms
Originality of Concept	Introduce fundamentally new metaphors, operators, or dynamics not present in existing algorithms.	Often reuse known metaphors with superficial changes.
Core Mechanistic Innovation	Employ novel solution update rules, selection mechanisms, or search strategies that alter algorithm behavior significantly.	Retain the core mechanisms (e.g., mutation, crossover, and velocity update) from classical algorithms like GA or PSO.
Performance Improvement	Demonstrate substantial improvements over state-of-the-art algorithms across a wide range of benchmark problems.	Exhibit similar or worse performance compared to the base algorithms; improvements are usually minor or dataset-specific.
Theoretical Foundation	Accompanied by theoretical analyses such as convergence proofs, complexity analysis, or formal models.	Rarely include theoretical justification; mostly evaluated via empirical trials.
Computational Efficiency	Designed with consideration for scalability and computational cost; often introduce efficient operators.	May involve redundant or costly operations that increase complexity without added value.
Contribution to the Field	Open new avenues for research, inspire follow-up work, or integrate with other frameworks.	Provide limited insight or innovation; mostly serve as incremental publications with little long-term impact.

6. Actionable Recommendations for Advancing Metaheuristic Algorithm Research

This section presents a comprehensive analysis of potential research directions in metaheuristic algorithms, accompanied by constructive feedback and practical recommendations. The identified insights may stimulate novel research avenues, address emerging challenges, and facilitate the development of more effective optimization techniques. In particular, we advocate for the adoption of the METRICS framework to evaluate the novelty, rigor, and practical value of newly developed algorithms.

The METRICS framework encompasses the following seven essential dimensions: Mathematical formulation, Exploration/exploitation balance, Theoretical analysis, Reproducibility, Innovation justification, Comparison protocol, and Structural bias testing. Each component offers a concrete criterion for assessing key properties of an algorithm, from the clarity of its formal definition and the soundness of its theoretical foundations to the fairness of its experimental comparisons and the transparency of its implementation. For instance, rigorous exploration–exploitation analysis and ablation studies can shed light on the internal dynamics and effectiveness of each algorithmic component, while structural

bias testing can reveal hidden tendencies that may compromise generalization. The specific contents of the METRICS framework are shown in Table 5.

By systematically applying the METRICS checklist, researchers can enhance the transparency, comparability, and scientific robustness of their algorithmic contributions. These efforts not only support the advancement of metaheuristic algorithms but also provide valuable guidance for future research endeavors in this rapidly evolving field.

Table 5. METRICS checklist for reporting new algorithms.

Item	Aspect	Checklist Description
M	Mathematical Formulation	Clearly define the algorithm using formal mathematical notations. Include objective functions, constraints, and update rules.
E	Exploration/Exploitation Balance	Provide analysis or visualizations that demonstrate the balance between exploration and exploitation.
T	Theoretical Analysis	Present theoretical properties such as convergence, time complexity, or approximation bounds, if applicable.
R	Reproducibility	Ensure reproducibility by including pseudocode, hyperparameters, datasets, and code availability (e.g., via GitHub).
I	Innovation Justification	Justify the novelty and necessity of the proposed mechanism. Include ablation studies to isolate the contribution of each component.
C	Comparison Protocol	Compare the proposed algorithm against state-of-the-art baselines using standardized benchmarks. Report fair and consistent evaluation metrics.
S	Structural Bias Testing	Include tests for structural bias. Provide visual or statistical analysis to support claims.

Additional Notes for Implementation:

- **Visualization:** Include plots or tables to illustrate performance trade-offs such as exploration vs. exploitation or convergence speed.
- **Ablation Studies:** Systematically remove or vary components to quantify their contributions.
- **Ethics/Societal Impact:** If applicable, discuss potential risks, biases, or misuse scenarios in real-world deployment.

The decision tree illustrated in Figure 5 provides a systematic approach for identifying structural bias in metaheuristic algorithms. It begins by assessing whether the algorithm's search trajectory is concentrated in a specific region. If concentration is detected, the user is directed to perform either a Signature Test or a Generalized Signature Test to further investigate the bias. The next step involves checking for central bias, with two possible outcomes: if central bias is present, the user should employ the Shifted Benchmark Test to evaluate landscape shifts; if not, the user should examine the algorithm for symmetry or axis preference.

Depending on the results of these tests, the decision tree guides the user to apply specific tools such as Region Scaling or the BIAS Toolbox to address the identified issues. The final branches of the tree categorize the outcome as either "Likely unbiased" or "Structural bias identified," providing clear conclusions based on the analysis. This structured approach ensures a thorough and methodical evaluation of potential structural biases in metaheuristic algorithms.

Some other key recommendations are as follows:

1. Propose actionable improvements to existing algorithms rather than relying solely on metaphors: While metaphor-based algorithms can be creative, true innovation requires authors to critically analyze and enhance the underlying mechanisms of existing methods. Instead of merely introducing a new metaphor, researchers should consider the following: (1) Through a systematic algorithm evaluation framework [113,114], specific limitations in existing algorithms (e.g., premature convergence, poor scalability) can be identified, and corresponding modifications can be proposed to address these issues. (2) Justify improvements with theoretical analysis (e.g., convergence guarantees and complexity reduction) or empirical validation on real-world problems.

- (3) Clearly articulate how the proposed changes differ from prior work and why they constitute a meaningful advance.
- ◇ **Mechanism-Level Diagnosis:** Conduct an in-depth analysis of existing algorithms to identify specific issues such as premature convergence, stagnation, or high computational cost.
 - ◇ **Component-Wise Modification:** Replace or enhance key components (e.g., selection, mutation, or update rules) with well-justified alternatives designed to improve performance on identified weaknesses.
 - ◇ **Empirical and Theoretical Justification:** Support the modifications with rigorous theoretical analysis (e.g., complexity bounds and convergence proofs) and ablation studies that isolate the effect of each change.
 - ◇ **Highlight Novelty and Relevance:** Clearly explain how the proposed modifications differ from prior work and why they address a real shortcoming rather than simply rebranding an existing idea.
2. Prioritize solving real-world optimization problems rather than tailoring structurally biased algorithms to fit biased benchmark functions: The transition from theoretical research to practical applications necessitates moving beyond algorithm optimization for benchmark functions toward addressing real-world optimization challenges. The use of biased benchmarks can guide the development of algorithms that exploit such biases, but this should not be the end goal. The true objective is to solve the specific problem at hand in the most efficient and effective way possible. In many cases, real-world problems differ significantly from the idealized scenarios presented in benchmark problems. When benchmarks are designed solely for performance testing, they often contain features that favor algorithms with specific structural biases, which can lead to misleading conclusions and unfair comparisons. This overemphasis on fitting algorithms to benchmark problems can result in solutions that perform well in artificial settings but fail to generalize to more complex, dynamic, and noisy real-world environments.
- ◇ **Incorporate Real-World Case Studies:** Validate algorithms on real-world problems from domains such as logistics, energy, scheduling, or engineering design to assess practical relevance.
 - ◇ **Cross-Validation on Application Domains:** Evaluate algorithm performance across multiple real-world scenarios instead of focusing solely on synthetic benchmarks.
 - ◇ **Problem-Aware Algorithm Design:** Integrate domain knowledge into algorithm design (e.g., constraints and objective structure) to better adapt to specific application needs.
3. A correct and comprehensive view of the NFL Theorem: The No Free Lunch (NFL) Theorem [115], a foundational result in optimization theory, asserts that when averaged over all possible objective functions, all optimization algorithms perform equally in terms of their success rates. In other words, no algorithm is universally superior across the entire space of optimization problems. This theorem has profound implications for the design and evaluation of metaheuristic algorithms [64]. Unfortunately, the NFL Theorem is often misunderstood or misused in the metaheuristics community. Some works cite it merely to justify the creation of new algorithms without rigorously addressing the problem-specific contexts in which such algorithms are to be applied. This superficial use neglects the theorem's core implication—algorithmic performance is inherently problem-dependent. Simply put, the success of any given metaheuristic is contingent upon the structural characteristics of the target problem

domain, such as modality, dimensionality, constraint complexity, or noise properties. A deeper understanding of the NFL Theorem encourages a paradigm shift in metaheuristic design—from the pursuit of generic, one-size-fits-all algorithms to the development of customized, domain-aware strategies. For instance, integrating prior knowledge, adaptive control parameters, or hybrid mechanisms tailored to a specific problem landscape aligns with the spirit of the NFL Theorem. It promotes a scientific, problem-driven methodology rather than blind algorithm proliferation. Therefore, rather than serving as a pretext for proposing yet another algorithm inspired by arbitrary metaphors, the NFL Theorem should be interpreted as a call to rigorously align algorithmic mechanisms with the structural features of specific optimization problems. This fosters a more meaningful balance between theoretical insight and practical effectiveness in the ongoing advancement of metaheuristic research.

- ◇ **Perform problem landscape analysis:** Analyze key characteristics of the optimization problem landscape—such as modality, ruggedness, separability, and epistasis—to classify problem types.
 - ◇ **Construct algorithm portfolios or hybrid frameworks:** Develop algorithm portfolios or hybrid metaheuristic frameworks that can dynamically adapt to different problem structures.
 - ◇ **Incorporate domain knowledge or problem-specific priors:** Integrate relevant domain-specific information—such as physical laws, empirical constraints, or expert heuristics—into the design of algorithmic operators (e.g., initialization, mutation, repair).
4. Use fair experimental comparisons, rather than relying on a subset of problems or choosing “toy” competitors: To ensure the validity and reliability of experimental comparisons, it is crucial to adopt a fair and comprehensive approach, rather than relying on a subset of problems or choosing “toy” competitors. These subset problems are unlikely to represent all scenarios in the real world and can lead to misleading conclusions about an algorithm’s true capabilities. Furthermore, choosing “toy” competitors—algorithms intentionally designed to be weak or perform poorly—only serves to artificially inflate the performance of the evaluated algorithm. Such practices distort the competitive landscape and undermine the credibility of the research. A fair experimental comparison should involve a diverse set of benchmark problems spanning a wide range of difficulty levels and characteristics, reflecting the complexity, uncertainty, and dynamic nature of real-world optimization tasks.
- ◇ **Adopt Standard Benchmark Suites:** Employ well-established and widely accepted benchmark sets (e.g., CEC [116]) that span multiple problem types and difficulties.
 - ◇ **Select Strong Baselines:** Compare against state-of-the-art and high-performing algorithms from both classical and recent literature, rather than underperforming ones.
 - ◇ **Ensure Reproducibility:** Report experimental setups in full detail (parameter settings, stopping criteria, and platform) and release source code when possible.
 - ◇ **Conduct Statistical Tests:** Use non-parametric tests (e.g., Wilcoxon signed-rank and Friedman test) to ensure performance differences are statistically significant.
5. Providing more comprehensive and transparent analysis: Researchers are encouraged to incorporate comprehensive and transparent analytical approaches when reporting their findings. Simply listing raw data in full-page tables is no longer sufficient; the proper validation of the results is essential and should include not only the application of appropriate statistical tests but also a clear justification for their use, ensuring that

all necessary assumptions are met. In addition to statistical testing, authors should employ visualization techniques that effectively summarize large volumes of data in a way that is easily interpretable by readers.

- ◇ **Use Appropriate Statistical Testing:** Apply statistical analysis with clear explanation of the assumptions, test types, and confidence levels (e.g., 95% confidence intervals).
- ◇ **Visual Data Summarization:** Use plots such as boxplots, convergence curves, or heatmaps to visually communicate algorithm behavior and performance trends.
- ◇ **Perform Ablation Studies:** Show how each algorithmic component contributes to overall performance by systematically removing or modifying them.
- ◇ **Report Variability and Robustness:** Report Variability and Robustness: Include metrics such as standard deviation, interquartile range, or success rate to reflect algorithm stability across runs.

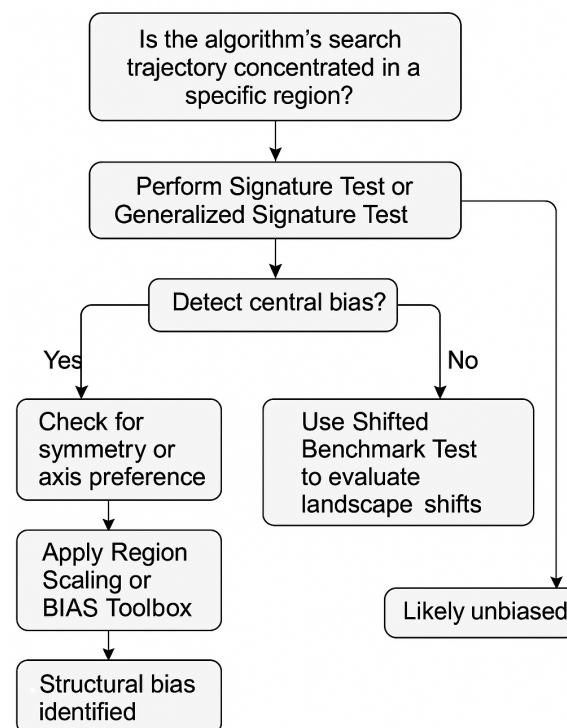


Figure 5. Decision tree for identifying structural bias in metaheuristic algorithms.

7. Conclusions

The development of metaheuristic algorithms has experienced significant growth in recent decades, but the theoretical understanding of their mechanisms has not kept pace with their practical advancements. While excessive reliance on biological metaphors has been criticized, we acknowledge that appropriately formalized metaphors can serve as valuable intuition-building tools—for instance, the thermodynamic interpretation of simulated annealing's cooling schedule provides both mathematical rigor (through Markov chain theory) and physical intuition for parameter tuning. Similarly, the “pheromone trail” concept in ant colony optimization, when grounded in probability transition matrices, offers both algorithmic guidance and cognitive accessibility. Another representative example is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which incorporates rigorous statistical learning to adapt the sampling distribution in continuous spaces, with well-established convergence proofs and benchmark performance across diverse optimization problems. Differential Evolution (DE) also exemplifies a well-designed algorithm whose simple yet effective mutation and crossover mechanisms have been math-

ematically analyzed and empirically validated, especially in the context of continuous numerical optimization. This imbalance between metaphorical novelty and theoretical depth threatens the field's progress across all subdomains—from single-objective optimization to multi-objective and hybrid approaches—potentially reducing it to an exercise in novelty without substantial problem-solving capabilities. To move forward, it is essential to ground bio-inspired algorithms in solid theoretical principles, focusing on their efficiency, convergence speed, and fitness stability, while recognizing that well-formalized metaphors can bridge the gap between mathematical abstraction and practical implementation.

Although this study focuses on single-objective metaheuristics, it is important to recognize the fundamental challenges associated with multi-objective optimization algorithms, which include the following: (1) the computational complexity of maintaining Pareto-optimal solutions, where metrics like hypervolume and generational distance often scale poorly with objective space dimensionality; (2) the tension between convergence (measured by indicators like inverted generational distance) and diversity preservation (evaluated through spreading metrics), which frequently leads to degraded performance when handling non-convex or discontinuous Pareto fronts; (3) the sensitivity to weight selection in decomposition-based approaches. These limitations motivate our focus on single-objective frameworks, where theoretical analysis of convergence can be more rigorously established.

The discussion of hybrid metaheuristics warrants careful delineation: Whereas combining evolutionary strategies with local search or integrating population-based methods with surrogate models can enhance efficiency, our analysis reveals that many hybrid variants suffer from the following: (1) diminishing returns when blending components without theoretical justification; (2) overfitting to specific benchmark sets despite improved hypervolume or error rate metrics; (3) increased configuration complexity that undermines reproducibility. This reinforces our advocacy for modular, theoretically grounded designs over ad hoc hybridization in single-objective contexts.

This paper systematically identifies and analyzes critical challenges impeding the development of robust and generalizable metaheuristic algorithms, including metaphor-based design, structural bias, and repackaging issues prevalent across algorithm variants. These insights reveal the need for a paradigm shift that emphasizes theoretical soundness and practical effectiveness. The study further examines existing approaches for detecting and mitigating structural biases in optimization processes, presenting methodologies applicable to single-objective, multi-objective, and hybrid algorithmic frameworks.

Our findings encourage a shift away from metaphor-oriented research and the uncritical use of bio-inspired elements. The field should adopt modular algorithm development frameworks that ensure transparency and maintain scientific rigor across pure and hybrid methodologies. By emphasizing theoretical foundations and strengthening experimental validation—particularly through rigorous sensitivity analysis of hybrid components and systematic benchmarking of multi-objective techniques. This work aims to shift research practices from the prevailing “publish first, analyze later” paradigm toward more methodologically sound approaches.

These recommendations seek to promote research cohesion in metaheuristics, advancing the field through scientifically rigorous and practically impactful approaches applicable to single-objective optimization, multi-objective scenarios, and hybrid algorithm development.

Author Contributions: Conceptualization, C.-H.W.; Methodology, C.-H.W. and K.H.; Software, K.H.; Funding acquisition, C.-H.W.; Validation, K.H., X.W. and Y.O.; Formal analysis, C.-H.W. and K.H.; Investigation, C.-H.W., X.W. and K.H.; Data curation, K.H., X.W. and Y.O.; Writing—original draft preparation, K.H. and C.-H.W.; Writing—review and editing, C.-H.W., K.H. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Fujian Provincial Department of Science and Technology, China, under Grant Number 2021J011070.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, C.H.; Nguyen, T.T.; Pan, J.S.; Dao, T.K. An optimization approach for potential power generator outputs based on parallelized firefly algorithm. *Smart Innov. Syst. Technol.* **2017**, *64*, 297–306.
2. Dong, Z.; Wang, C.H.; Zhao, Q.; Wei, Y.; Chen, S.; Yang, Q. A study on intelligent optimization algorithms for capacity allocation of production networks. *Lect. Notes Electr. Eng.* **2022**, *804*, 734–743.
3. Wei, Y.; Wang, C.H.; Suo, Y.; Zhao, Q.; Yuan, J.; Chen, S. FHO-based hybrid neural networks for short-term load forecasting in economic dispatch of power systems. *J. Netw. Intell.* **2025**, *10*, 262–284.
4. Chen, S.; Wang, C.H.; Dong, Z.; Zhao, Q.; Yang, Q.; Wei, Y.; Huang, G. Performance evaluation of three intelligent optimization algorithms for obstacle avoidance path planning. *Lect. Notes Electr. Eng.* **2022**, *833*, 60–69.
5. Wang, C.H.; Lee, C.J.; Wu, X. A coverage-based location approach and performance evaluation for the deployment of 5G base stations. *IEEE Access* **2020**, *8*, 123320–123333. [[CrossRef](#)]
6. Liu, H.; Wang, C.H. SEAMS: A surrogate-assisted evolutionary algorithm with metric-based dynamic strategy for expensive multi-objective optimization. *Expert Syst. Appl.* **2025**, *265*, 126050. [[CrossRef](#)]
7. Fister Jr, I.; Mlakar, U.; Brest, J.; Fister, I. A new population-based nature-inspired algorithm every month: Is the current era coming to the end. In *Proceedings of the 3rd Student Computer Science Research Conference, Ljubljana, Slovenia, 12 October 2016*; University of Primorska Press: Koper, Slovenia, 2016; pp. 33–37.
8. Wang, C.H.; Tian, R.; Hu, K.; Chen, Y.T.; Ku, T.H. A Markov decision optimization of medical service resources for two-class patient queues in emergency departments via particle swarm optimization algorithm. *Sci. Rep.* **2025**, *15*, 2942. [[CrossRef](#)]
9. Zhao, Q.; Duan, Q.; Yan, B.; Cheng, S.; Shi, Y. Automated design of metaheuristic algorithms: A survey. *arXiv* **2023**, arXiv:2303.06532.
10. Hussain, K.; Mohd Salleh, M.N.; Cheng, S.; Shi, Y. Metaheuristic research: A comprehensive survey. *Artif. Intell. Rev.* **2019**, *52*, 2191–2233. [[CrossRef](#)]
11. Tomar, V.; Bansal, M.; Singh, P. Metaheuristic algorithms for optimization: A brief review. *Eng. Proc.* **2024**, *59*, 238.
12. Nassef, A.M.; Abdelkareem, M.A.; Maghrabie, H.M.; Baroutaji, A. Review of metaheuristic optimization algorithms for power systems problems. *Sustainability* **2023**, *15*, 9434. [[CrossRef](#)]
13. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [[CrossRef](#)]
14. Wang, C.H.; Yuan, J.; Zeng, Y.; Lin, S. A deep learning integrated framework for predicting stock index price and fluctuation via singular spectrum analysis and particle swarm optimization. *Appl. Intell.* **2024**, *54*, 1770–1797. [[CrossRef](#)]
15. Whitacre, J.M. Survival of the flexible: Explaining the recent popularity of nature-inspired optimization within a rapidly evolving world. *Computing* **2011**, *93*, 135–146. [[CrossRef](#)]
16. Karafotias, G.; Hoogendoorn, M.; Eiben, A.E. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Trans. Evol. Comput.* **2015**, *19*, 167–187. [[CrossRef](#)]
17. Wang, C.H.; Zhao, Q.; Tian, R. Short-term wind power prediction based on a hybrid Markov-based PSO-BP neural network. *Energies* **2023**, *16*, 4282. [[CrossRef](#)]
18. Yang, Q.; Wang, C.H.; Dao, T.K.; Nguyen, T.T.; Zhao, Q.; Chen, S. An Optimal Wind Turbine Control Based on Improved Chaotic Sparrow Search Algorithm with Normal Cloud Model. *J. Netw. Intell.* **2024**, *9*, 108–125.
19. Wang, C.H.; Chen, S.; Zhao, Q.; Suo, Y. An Efficient End-to-End Obstacle Avoidance Path Planning Algorithm for Intelligent Vehicles Based on Improved Whale Optimization Algorithm. *Mathematics* **2023**, *11*, 1800. [[CrossRef](#)]
20. Taleb, S.M.; Yasin, E.T.; Saadi, A.A.; Dogan, M.; Yahia, S.; Meraihi, Y.; Koklu, M.; Mirjalili, S.; Ramdane-Cherif, A. A Comprehensive Survey of Aquila Optimizer: Theory, Variants, Hybridization, and Applications. *Arch. Comput. Methods Eng.* **2025**, 1–47. [[CrossRef](#)]
21. Kazikova, A.; Pluhacek, M.; Senkerik, R. How Does the Number of Objective Function Evaluations Impact Our Understanding of Metaheuristics Behavior? *IEEE Access* **2021**, *9*, 44032–44048. [[CrossRef](#)]
22. Valencia-Rivera, G.H.; Benavides-Robles, M.T.; Morales, A.V.; Amaya, I.; Cruz-Duarte, J.M.; Ortiz-Bayliss, J.C.; Avina-Cervantes, J.G. A systematic review of metaheuristic algorithms in electric power systems optimization. *Appl. Soft Comput.* **2024**, *150*, 111047. [[CrossRef](#)]
23. Sadeghian, Z.; Akbari, E.; Nematzadeh, H.; Motameni, H. A review of feature selection methods based on meta-heuristic algorithms. *J. Exp. Theor. Artif. Intell.* **2025**, *37*, 1–51. [[CrossRef](#)]

24. Benaissa, B.; Kobayashi, M.; Al Ali, M.; Khatir, T.; Elmeliani, M.E.A.E. Metaheuristic optimization algorithms: An overview. *Hcmou J. -Sci.-Adv. Comput. Struct.* **2024**, *14*, 33–61. [\[CrossRef\]](#)
25. Li, G.; Zhang, T.; Tsai, C.Y.; Yao, L.; Lu, Y.; Tang, J. Review of the metaheuristic algorithms in applications: Visual analysis based on bibliometrics (1994–2023). *Expert Syst. Appl.* **2024**, *255*, 124857. [\[CrossRef\]](#)
26. Rautray, R.; Dash, R.; Dash, R.; Chandra Balabantaray, R.; Parida, S.P. A review on metaheuristic approaches for optimization problems. In *Computational Intelligence in Healthcare Informatics*; Springer: Singapore, 2024; pp. 33–55.
27. Hooker, J.N. Needed: An Empirical Science of Algorithms. *Oper. Res.* **1994**, *42*, 201–212. [\[CrossRef\]](#)
28. Eiben, A.E.; Jelasity, M. A critical note on experimental research methodology in EC. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Honolulu, HI, USA, 12–17 May 2002; Volume 1, pp. 582–587.
29. García-Martínez, C.; Gutiérrez, P.D.; Molina, D.; Lozano, M.; Herrera, F. Since CEC 2005 competition on real-parameter optimisation: A decade of research, progress and comparative analysis's weakness. *Soft Comput.* **2017**, *21*, 5573–5583. [\[CrossRef\]](#)
30. Campelo, F.; Takahashi, F. Sample size estimation for power and accuracy in the experimental comparison of algorithms. *J. Heuristics* **2019**, *25*, 305–338. [\[CrossRef\]](#)
31. Song, Q.; Fong, S. Brick-Up Metaheuristic Algorithms. In Proceedings of the 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, Japan, 10–14 July 2016; pp. 583–587.
32. Sarhani, M.; Voß, S.; Jovanovic, R. Initialization of metaheuristics: Comprehensive review, critical analysis, and research directions. *Int. Trans. Oper. Res.* **2023**, *30*, 3361–3397. [\[CrossRef\]](#)
33. Piotrowski, A.P.; Napiorkowski, J.J. Some metaheuristics should be simplified. *Inf. Sci.* **2018**, *427*, 32–62. [\[CrossRef\]](#)
34. Dragoi, E.N.; Dafinescu, V. Review of Metaheuristics Inspired from the Animal Kingdom. *Mathematics* **2021**, *9*, 2335. [\[CrossRef\]](#)
35. Oliveira, M.; Pinheiro, D.; Macedo, M.; Bastos-Filho, C.; Menezes, R. Uncovering the social interaction network in swarm intelligence algorithms. *Appl. Netw. Sci.* **2020**, *5*, 24. [\[CrossRef\]](#)
36. Hodashinsky, I.A. Methods for Improving the Efficiency of Swarm Optimization Algorithms. A Survey. *Autom. Remote Control* **2021**, *82*, 935–967. [\[CrossRef\]](#)
37. Campelo, F.; Aranha, C. Sharks, Zombies and Volleyball: Lessons from the Evolutionary Computation Bestiary. In Proceedings of the LIFELIKE Computing Systems Workshop 2021. CEUR-WS.org, 2021. Available online: <https://publications.aston.ac.uk/id/eprint/43161/1/main.pdf> (accessed on 20 December 2024).
38. Del Ser, J.; Osaba, E.; Molina, D.; Yang, X.S.; Salcedo-Sanz, S.; Camacho, D.; Das, S.; Suganthan, P.N.; Coello, C.A.C.; Herrera, F. Bio-inspired computation: Where we stand and what's next. *Swarm Evol. Comput.* **2019**, *48*, 220–250. [\[CrossRef\]](#)
39. Osaba, E.; Villar-Rodriguez, E.; Del Ser, J.; Nebro, A.J.; Molina, D.; LaTorre, A.; Suganthan, P.N.; Coello, C.A.C.; Herrera, F. A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. *Swarm Evol. Comput.* **2021**, *64*, 100888. [\[CrossRef\]](#)
40. LaTorre, A.; Molina, D.; Osaba, E.; Poyatos, J.; Del Ser, J.; Herrera, F. A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm Evol. Comput.* **2021**, *67*, 100973. [\[CrossRef\]](#)
41. Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **1986**, *13*, 533–549. [\[CrossRef\]](#)
42. Abualgah, L.; Shehab, M.; Alshinwan, M.; Mirjalili, S.; Elaziz, M.A. Ant lion optimizer: A comprehensive survey of its variants and applications. *Arch. Comput. Methods Eng.* **2021**, *28*, 1397–1416. [\[CrossRef\]](#)
43. Makhadmeh, S.N.; Al-Betar, M.A.; Abasi, A.K.; Awadallah, M.A.; Doush, I.A.; Alyasseri, Z.A.A.; Alomari, O.A. Recent advances in butterfly optimization algorithm, its versions and applications. *Arch. Comput. Methods Eng.* **2023**, *30*, 1399–1420. [\[CrossRef\]](#)
44. Liu, Y.; As'arry, A.; Hassan, M.K.; Hairuddin, A.A.; Mohamad, H. Review of the grey wolf optimization algorithm: Variants and applications. *Neural Comput. Appl.* **2024**, *36*, 2713–2735. [\[CrossRef\]](#)
45. Gad, A.G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [\[CrossRef\]](#)
46. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle swarm optimization: A comprehensive survey. *IEEE Access* **2022**, *10*, 10031–10061. [\[CrossRef\]](#)
47. Priyadarshi, R.; Kumar, R.R. Evolution of Swarm Intelligence: A Systematic Review of Particle Swarm and Ant Colony Optimization Approaches in Modern Research. *Arch. Comput. Methods Eng.* **2025**, 1–42. [\[CrossRef\]](#)
48. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [\[CrossRef\]](#)
49. Guilmeau, T.; Chouzenoux, E.; Elvira, V. Simulated annealing: A review and a new scheme. In Proceedings of the 2021 IEEE Statistical Signal Processing Workshop (SSP), Rio de Janeiro, Brazil, 11–14 July 2021; pp. 101–105.
50. Abualgah, L.; Elaziz, M.A.; Hussien, A.G.; Alsalihi, B.; Jalali, S.M.J.; Gandomi, A.H. Lightning search algorithm: A comprehensive survey. *Appl. Intell.* **2021**, *51*, 2353–2376. [\[CrossRef\]](#) [\[PubMed\]](#)

51. Abdel-Basset, M.; Mohamed, R.; Chakraborty, R.K.; Sallam, K.; Ryan, M.J. An efficient teaching-learning-based optimization algorithm for parameters identification of photovoltaic models: Analysis and validations. *Energy Convers. Manag.* **2021**, *227*, 113614. [\[CrossRef\]](#)
52. Navaneetha Krishnan, M.; Thiagarajan, R. Multi-objective task scheduling in fog computing using improved gaining sharing knowledge based algorithm. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7227. [\[CrossRef\]](#)
53. Weyland, D. A Rigorous Analysis of the Harmony Search Algorithm: How the Research Community can be Misled by a “Novel” Methodology. *Int. J. Appl. Metaheuristic Comput.* **2010**, *1*, 50–60. [\[CrossRef\]](#)
54. Weyland, D. A critical analysis of the harmony search algorithm-How not to solve sudoku. *Oper. Res. Perspect.* **2015**, *2*, 97–105. [\[CrossRef\]](#)
55. Saka, M.P.; Hasançebi, O.; Geem, Z.W. Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm Evol. Comput.* **2016**, *28*, 88–97. [\[CrossRef\]](#)
56. Camacho-Villalón, C.L.; Dorigo, M.; Stützle, T. The intelligent water drops algorithm: Why it cannot be considered a novel algorithm. *Swarm Intell.* **2019**, *13*, 173–192. [\[CrossRef\]](#)
57. Simon, D.; Rarick, R.; Ergezer, M.; Du, D. Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. *Inf. Sci.* **2011**, *181*, 1224–1248. [\[CrossRef\]](#)
58. Camacho-Villalón, C.L.; Dorigo, M.; Stützle, T. Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: Six misleading optimization techniques inspired by bestial metaphors. *Int. Trans. Oper. Res.* **2023**, *30*, 2945–2971. [\[CrossRef\]](#)
59. Davarynejad, M.; van den Berg, J.; Rezaei, J. Evaluating center-seeking and initialization bias: The case of particle swarm and gravitational search algorithms. *Inf. Sci.* **2014**, *278*, 802–821. [\[CrossRef\]](#)
60. Camacho-Villalón, C.L.; Dorigo, M.; Stützle, T. An analysis of why cuckoo search does not bring any novel ideas to optimization. *Comput. Oper. Res.* **2022**, *142*, 105747. [\[CrossRef\]](#)
61. Baronti, L.; Castellani, M.; Pham, D.T. An analysis of the search mechanisms of the bees algorithm. *Swarm Evol. Comput.* **2020**, *59*, 100746. [\[CrossRef\]](#)
62. Kudela, J. The Evolutionary Computation Methods No One Should Use. *arXiv* **2023**, arXiv:2301.01984.
63. Piotrowski, A.P.; Napiorkowski, J.J.; Rowinski, P.M. How novel is the “novel” black hole optimization approach? *Inf. Sci.* **2014**, *267*, 191–200. [\[CrossRef\]](#)
64. Velasco, L.; Guerrero, H.; Hospitaler, A. A Literature Review and Critical Analysis of Metaheuristics Recently Developed. *Arch. Comput. Methods Eng.* **2024**, *31*, 125–146. [\[CrossRef\]](#)
65. Deng, L.; Liu, S. Deficiencies of the whale optimization algorithm and its validation method. *Expert Syst. Appl.* **2024**, *237*, 121544. [\[CrossRef\]](#)
66. Niu, P.; Niu, S.; liu, N.; Chang, L. The defect of the Grey Wolf optimization algorithm and its verification method. *Knowl.-Based Syst.* **2019**, *171*, 37–43. [\[CrossRef\]](#)
67. Hu, J.; Chen, H.; Heidari, A.A.; Wang, M.; Zhang, X.; Chen, Y.; Pan, Z. Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowl.-Based Syst.* **2021**, *213*, 106684. [\[CrossRef\]](#)
68. Harandi, N.; Van Messem, A.; De Neve, W.; Vankerschaver, J. Grasshopper Optimization Algorithm (GOA): A Novel Algorithm or A Variant of PSO? In *Proceedings of the International Conference on Swarm Intelligence, Konstanz, Germany, 9–11 October 2024*; Springer: Cham, Switzerland, 2024; pp. 84–97.
69. Rajwar, K.; Deep, K.; Mathirajan, M. Impact of Structural Bias on the Sine Cosine Algorithm: A Theoretical Investigation Using the Signature Test. In *Proceedings of the International Conference on Metaheuristics and Nature Inspired Computing, Marrakech, Morocco, 1–4 November 2023*; Springer: Cham, Switzerland, 2023; pp. 131–141.
70. Halsema, M.; Vermetten, D.; Bäck, T.; Van Stein, N. A Critical Analysis of Raven Roost Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, Melbourne, VIC, Australia, 14–18 July 2024*; pp. 1993–2001.
71. Kumar, M.; Rajwar, K.; Deep, K. Analysis of Marine Predators Algorithm using BIAS toolbox and Generalized Signature Test. *Alex. Eng. J.* **2024**, *95*, 38–49. [\[CrossRef\]](#)
72. Deng, L.; Liu, S. Exposing the chimp optimization algorithm: A misleading metaheuristic technique with structural bias. *Appl. Soft Comput.* **2024**, *158*, 111574. [\[CrossRef\]](#)
73. Deng, L.; Liu, S. Metaheuristics exposed: Unmasking the design pitfalls of arithmetic optimization algorithm in benchmarking. *Appl. Soft Comput.* **2024**, *160*, 111696. [\[CrossRef\]](#)
74. Chen, S.; Islam, S.; Lao, S. The Danger of Metaphors for Metaheuristic Design. In *Proceedings of the 2023 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Recife-Pe, Brazil, 29 October–1 November 2023*; pp. 1–6.
75. Sörensen, K. Metaheuristics—the metaphor exposed. *Int. Trans. Oper. Res.* **2015**, *22*, 3–18. [\[CrossRef\]](#)
76. Kononova, A.V.; Caraffini, F.; Wang, H.; Bäck, T. Can Single Solution Optimisation Methods Be Structurally Biased? In *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020*; pp. 1–9.
77. Caraffini, F.; Kononova, A.V. Structural bias in differential evolution: A preliminary study. *Aip Conf. Proc.* **2019**, *2070*, 020005.

78. Castelli, M.; Manzoni, L.; Mariot, L.; Nobile, M.S.; Tangherloni, A. Salp Swarm Optimization: A critical review. *Expert Syst. Appl.* **2022**, *189*, 116029. [\[CrossRef\]](#)
79. Vent, W. *Rechenberg, Ingo, Evolutionsstrategie—Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*; 170 S. mit 36 Abb. Frommann-Holzboog-Verlag: Stuttgart, Germany, 1973.
80. Padberg, M. Harmony Search Algorithms for binary optimization problems. In *Operations Research Proceedings 2011*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 343–348.
81. De Corte, A.; Sörensen, K. Optimisation of gravity-fed water distribution network design: A critical review. *Eur. J. Oper. Res.* **2013**, *228*, 1–10. [\[CrossRef\]](#)
82. Schwefel, H.P. *Numerical Optimization of Computer Models*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1981.
83. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
84. Derrac, J.; García, S.; Hui, S.; Suganthan, P.N.; Herrera, F. Analyzing convergence performance of evolutionary algorithms: A statistical approach. *Inf. Sci.* **2014**, *289*, 41–58. [\[CrossRef\]](#)
85. Das, S.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*; Jadavpur University, Nanyang Technological University: Kolkata, India, 2010; pp. 341–359.
86. Veček, N.; Mernik, M.; Črepinšek, M. A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. *Inf. Sci.* **2014**, *277*, 656–679. [\[CrossRef\]](#)
87. Dymond, A.S.; Engelbrecht, A.P.; Kok, S.; Heyns, P.S. Tuning Optimization Algorithms Under Multiple Objective Function Evaluation Budgets. *IEEE Trans. Evol. Comput.* **2015**, *19*, 341–358. [\[CrossRef\]](#)
88. Liao, T.; Aydın, D.; Stützle, T. Artificial bee colonies for continuous optimization: Experimental analysis and improvements. *Swarm Intell.* **2013**, *7*, 327–356. [\[CrossRef\]](#)
89. Draa, A. On the performances of the flower pollination algorithm—Qualitative and quantitative analyses. *Appl. Soft Comput.* **2015**, *34*, 349–371. [\[CrossRef\]](#)
90. Piotrowski, A.P. Regarding the rankings of optimization heuristics based on artificially-constructed benchmark functions. *Inf. Sci.* **2015**, *297*, 191–201. [\[CrossRef\]](#)
91. Piotrowski, A.P.; Napiorkowski, M.J. May the same numerical optimizer be used when searching either for the best or for the worst solution to a real-world problem? *Inf. Sci.* **2016**, *373*, 124–148. [\[CrossRef\]](#)
92. Mernik, M.; Liu, S.H.; Karaboga, D.; Črepinšek, M. On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. *Inf. Sci.* **2015**, *291*, 115–127. [\[CrossRef\]](#)
93. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [\[CrossRef\]](#)
94. Ampellio, E.; Vassio, L. A hybrid swarm-based algorithm for single-objective optimization problems involving high-cost analyses. *Swarm Intell.* **2016**, *10*, 99–121. [\[CrossRef\]](#)
95. Črepinšek, M.; Liu, S.H.; Mernik, L.; Mernik, M. Is a comparison of results meaningful from the inexact replications of computational experiments? *Soft Comput.* **2016**, *20*, 223–235. [\[CrossRef\]](#)
96. Weise, T.; Chiong, R.; Tang, K. Evolutionary Optimization: Pitfalls and Booby Traps. *J. Comput. Sci. Technol.* **2012**, *27*, 907–936. [\[CrossRef\]](#)
97. Črepinšek, M.; Liu, S.H.; Mernik, L. A note on teaching–learning-based optimization algorithm. *Inf. Sci.* **2012**, *212*, 79–93. [\[CrossRef\]](#)
98. Chinta, S.; Kommadath, R.; Kotecha, P. A note on multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO). *Inf. Sci.* **2016**, *373*, 337–350. [\[CrossRef\]](#)
99. Hall, J.C.; Mills, B.; Nguyen, N.; Hall, J.L. Methodologic standards in surgical trials. *Surgery* **1996**, *119*, 466–472. [\[CrossRef\]](#)
100. Kitchenham, B.A.; Pfleeger, S.L.; Pickard, L.M.; Jones, P.W.; Hoaglin, D.C.; El Emam, K.; Rosenberg, J. Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Softw. Eng.* **2002**, *28*, 721–734. [\[CrossRef\]](#)
101. Clerc, M. Biases and signatures. In *Guided Randomness in Optimization*; John Wiley & Sons: Hoboken, NJ, USA, 2015; pp. 139–145.
102. Rajwar, K.; Deep, K. Uncovering structural bias in population-based optimization algorithms: A theoretical and simulation-based analysis of the generalized signature test. *Expert Syst. Appl.* **2024**, *240*, 122332. [\[CrossRef\]](#)
103. Kononova, A.V.; Corne, D.W.; De Wilde, P.; Shneer, V.; Caraffini, F. Structural bias in population-based algorithms. *Inf. Sci.* **2015**, *298*, 468–490. [\[CrossRef\]](#)
104. Inselberg, A. The plane with parallel coordinates. *Vis. Comput.* **1985**, *1*, 69–91. [\[CrossRef\]](#)
105. Cleghorn, C.W.; Engelbrecht, A.P. A generalized theoretical deterministic particle swarm model. *Swarm Intell.* **2014**, *8*, 35–59. [\[CrossRef\]](#)
106. Vermetten, D.; van Stein, B.; Caraffini, F.; Minku, L.L.; Kononova, A.V. BIAS: A Toolbox for Benchmarking Structural Bias in the Continuous Domain. *IEEE Trans. Evol. Comput.* **2022**, *26*, 1380–1393. [\[CrossRef\]](#)

107. Gehlhaar, D.K. Tuning evolutionary programming for conformationally flexible molecular docking. In Proceedings of the 5th Annual Conference on Evolutionary Programming, San Diego, CA, USA, 29 February–2 March 1996; pp. 419–429.
108. de Castro, L.N. Fundamentals of natural computing: An overview. *Phys. Life Rev.* **2007**, *4*, 1–36. [[CrossRef](#)]
109. Bonabeau, E.; Dorigo, M.; Theraulaz, G. Inspiration for optimization from social insect behaviour. *Nature* **2000**, *406*, 39–42. [[CrossRef](#)] [[PubMed](#)]
110. Beyer, H.G.; Schwefel, H.P.; Wegener, I. How to analyse evolutionary algorithms. *Theor. Comput. Sci.* **2002**, *287*, 101–130. [[CrossRef](#)]
111. Campelo, F.; Aranha, C. Lessons from the Evolutionary Computation Bestiary. *Artif. Life* **2023**, *29*, 421–432. [[CrossRef](#)]
112. Smaldino, P.E.; McElreath, R. The natural selection of bad science. *R. Soc. Open Sci.* **2016**, *3*, 160384. [[CrossRef](#)]
113. Hayward, L.; Engelbrecht, A. How to Tell a Fish from a Bee: Constructing Meta-Heuristic Search Behaviour Characteristics. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation, Lisbon, Portugal, 15–19 July 2023*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1562–1569.
114. Ivković, N.; Kudelić, R.; Črepinšek, M. Probability and Certainty in the Performance of Evolutionary and Swarm Optimization Algorithms. *Mathematics* **2022**, *10*, 4364. [[CrossRef](#)]
115. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
116. Biedrzycki, R.; Arabas, J.; Warchulski, E. A version of NL-SHADE-RSP algorithm with midpoint for CEC 2022 single objective bound constrained problems. In Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC), Padua, Italy, 18–23 July 2022; pp. 1–8.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.