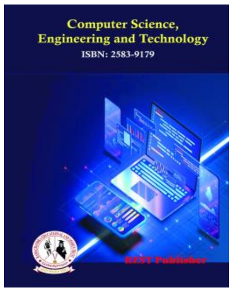
Computer Science, Engineering and Technology
Vol: 2(4), December 2024
REST Publisher; ISSN: 2583-9179 (Online)
Website: https://restpublisher.com/journals/cset/
DOI: https://doi.org/10.46632/cset/2/4/2



Classification, Applications and Future Directions of Bio-Inspired Algorithms: From Swarm Intelligence to Quantum Computing Integration Using PROMETHEE Methodology

C. Kalpana
Amity school of Information Technology, Amity University, Maharashtra, India.
Corresponding Author Email: rkalpz@gmail.com

Abstract. *Introduction: Biologically inspired algorithms (BIAs) are computational methods that model natural processes and systems. Although various types of BIAs have been developed over time, their classification has not been extensively studied. The most widely recognized types include evolution-based algorithms that draw inspiration from natural evolution and swarm-based algorithms that mimic the collective behaviors of animal groups. In addition, ecosystem-based algorithms provide another classification perspective. Many modern BIAs go beyond this well-established framework. These algorithms have shown themselves to be highly adaptable, finding application in a variety of domains to solve challenging optimization and decision-making problems. Examples include solving problems such as network routing, resource planning, graph coloring, and the traveling salesman problem. In artificial intelligence, BIAs have improved neural networks by improving clustering, classification, and prediction accuracy. In the medical field, they have made significant contributions to advances in diagnosis and treatment planning. As BIAs continue to evolve, their integration with advanced technologies such as quantum computing paves the way to overcome challenges such as slow integration and computational inefficiency. Reflecting the adaptability and resilience of natural systems, BIAs are increasingly being used in emerging areas such as cloud computing and wireless sensor networks, highlighting their growing relevance for scalable and reliable problem solving. Research significance: The focus of this research is on investigating biologically inspired algorithms (BIAs), which are computational techniques that model natural processes. These algorithms are crucial for solving complex optimization and decision-making problems in diverse domains such as resource planning, networking, artificial intelligence, and medicine. Despite their widespread applications, the classification of BIAs has received little attention. This research highlights the importance of understanding their classification and categorization, which can lead to more effective algorithm selection and optimization. By reviewing and refining the evolutionary-based, swarm-based, and ecosystem-based algorithm categories, this work helps to improve the theoretical framework of BIAs. In addition, the research explores the potential for improving BIA convergence rates by integrating emerging technologies such as quantum computing, which addresses key challenges in real-world applications. In the end, this research could promote the application of bio-inspired algorithms across a range of domains, enhancing computer efficiency, scalability, and flexibility. Methology: Alternatives: Genetic Algorithm (Evolutionary based), Particle Swarm Optimization (Swarm-based), Ant Colony Optimization (Swarm-based), Artificial Bee Colony Algorithm (Swarm-based). Evaluation Parameters: Convergence Speed, Accuracy, Scalability, Resource Efficiency, Computational Complexity, Energy Consumption, Noise Sensitivity. Result: The results show that Genetic Algorithm (Evolutionary based) received the highest ranking, whereas Artificial Bee Colony Algorithm (Swarm-based) received the lowest ranking. Conclusion: According to the PROMETHEE method, Genetic Algorithm (Evolutionary based) ranks highest in terms of its value for Bio-Inspired Algorithms.*

Keywords: *social behaviour of animals, and ecosystem dynamics, evolutionary-based, swarm-based, and ecosystem-based algorithms.*

1. INTRODUCTION

Drawing inspiration from various biological models, many biologically inspired algorithms have been proposed, designed, and developed. However, limited research focuses on their classification. The two most well-known types are swarm-based algorithms that imitate the collective behavior of animals, and evolutionary-based algorithms that draw inspiration from natural evolution. Based on this understanding, Binita and Satya divided biologically inspired algorithms into three primary categories: ecosystem-based, swarm-based, and evolutionary-based algorithms. Some novel algorithms do not fit into these categories, while most bio-inspired algorithms do. [1] Graph coloring, scheduling, resource-constrained problems, the traveling salesman problem, and network routing optimization are some of the problems that are frequently solved using algorithms inspired by biology. These algorithms are frequently used to improve search performance, prediction, classification, image processing, and clustering in neural networks. They also play a significant role in networking tasks, including routing, clustering, and resource scheduling. In the medical field, numerous bio-inspired algorithms have been developed to perform various tasks, achieving significant performance gains. [2] Bio-inspired algorithms (BIAs) are heuristic methods that follow strategies found in nature. Many biological processes can be considered as finite optimization mechanisms. These methods rely on various random effects, classifying them as a unique random process. To design bio-inspired methods, an appropriate representation must be chosen to ensure their performance. [3] In the future, bio-inspired algorithms could be combined with other strategies and methods, such as confusion theory and quantum computing, to overcome the convergence speed issue that often arises when tackling challenging real-world problems. When combined with the features and capabilities of quantum computing, some bio-inspired algorithms have recently shown improved convergence rates and overall computer performance. [4] Swarm intelligence (SI)-based algorithms are a special subset of bio-inspired algorithms, which fall under the broader category of nature-inspired algorithms. From a set-theoretic perspective, SI-based algorithms are subsumed within bio-inspired algorithms, which belong to the larger domain of nature-inspired approaches. [5] The content includes several tutorials that explain how to modify various components, integrate user-defined functions, and create custom components to suit specific applications. In addition, it provides "recipes" in the form of code snippets, explaining advanced concepts such as implementing dictionary ordering to compare individuals or managing immigrants between islands in parallel computing environments connected via a network. [6] Nature provides solutions to the challenges of life through systems such as bee colonies or neural networks, which work in an organized and interconnected manner to solve complex problems with elegance and creativity. Rather than being used directly as strategies for solving computational problems, these systems encourage innovative approaches. Recent studies have demonstrated that both biological systems and computers can evolve independently to maximize the trade-off between physical cost and complexity, even if they are not specifically designed with biological inspiration in mind. [7] The primary objective of this project is to explore the use of bio-inspired algorithms in cloud computing. Optimal search, load balancing, resource scheduling, and optimization difficulties are all applications for these methods. Unlike more complex approaches, bio-inspired algorithms provide a natural and efficient way to solve such problems. [8] Wireless sensor network (WSN) localization involves a two-step process. The initial phase, referred to as the ranking phase, estimates the distance to beacons or stationary nodes based on signal strength or propagation time. Metrics such as arrival time, round-trip time, or time difference of signal arrival are used to determine propagation time. However, due to noise interference, it is challenging to accurately measure these parameters, which can cause errors in the outputs of location algorithms. [9] There are two stages in localizing a wireless sensor network (WSN). The initial stage, known as the ranking stage, involves estimating the distance to beacons or fixed nodes based on the strength of the received signal or the propagation time of the signal. Metrics such as arrival time, round-trip time, or signal arrival time difference are used to calculate the propagation time of a signal. However, accurately measuring these characteristics is challenging due to noise interference, which can lead to inaccurate location algorithm results. [10] At the same time, there is a growing demand for autonomous, scalable, adaptive, robust, and reliable service-oriented systems. Biological systems are recognized for possessing these properties, leading to research efforts to apply biologically inspired methods to address challenges in web service organization. In the following sections, we will explore several biologically inspired algorithms that have been applied to Web service organization from various perspectives, before concluding with a discussion of upcoming projects. [11] The metaheuristics that excelled in these competitions (CMA-ES, DE, and MVMO) are not entirely bio-inspired and, in some cases, have evolved beyond their nature-inspired foundations. While the discovery of new metaheuristics is significant, the findings emphasize that their performance is of greater importance in solving selection problems. [12] The dominant rooster takes the lead in foraging for food and defends the group's territory from encroachment. Dominant hens will usually join the leader rooster when searching for food. In contrast, subordinate hens will stay on the outskirts of the group when searching for food. There is also competition among hens. On the other hand, chicks will stay close to their mother when searching for food. A hierarchical structure is important for the social dynamics of hens, with leader hens asserting control over the weaker ones. The most dominant hens will be closest to the leader rooster, while the most submissive hens, along with the roosters, will usually be on the edges of the group. [13] To solve any combinatorial problem such as image segmentation, two key properties must be defined: a feasible solution and a suitable objective function. These requirements vary depending on the specific problem. In image segmentation, optimization tasks typically focus on determining optimal threshold values for multilevel thresholding methods or identifying the best cluster centers for algorithms such as K-means or FCM clustering. The next section explores the application of these techniques to image segmentation. [14] Through bio-inspired algorithms, generators use the integral transient response component of the speed deviation as a function to minimize it. This leads to better damping of the system. When inter-influences affect multiple generators in an interconnected power system, an objective function is created to consider the effects seen by each generator. The definition of this objective function is: [15] Text analytics is the process of processing text to extract relevant information from sources such as news articles, company papers, survey results, In text analytics, four primary techniques are commonly used for sources such as online forums, blogs, emails, and social media feeds:Information extraction is the process of converting unstructured data, such as determining the name, type, and expiration date of a medication from a patient's medical records, into organized forms. Text summarization: Creates concise summaries of multiple documents related to a specific topic. Question and answer: Responds to user inquiries using natural language processing. Sentiment analysis assesses people's views or beliefs about events, products, or services. [16] Diversity strategies in multimodal optimization algorithms improve the search process by reducing genetic drift, a phenomenon that causes loss of diversity in bio-inspired algorithms. These techniques address many unexplored variants, many of which have improved over time and show greater competitiveness compared to their original versions or other variants of bio-inspired algorithms. [17] There are three types of multi-solution metaheuristic algorithms: swarm-based, physics-based, and evolutionary methods. The first type of algorithms, known as evolutionary-based algorithms, include algorithms including mutation, recombination, and selection, and are inspired by biological evolution. These techniques make no assumptions about the fitness landscape in the game. The second group of algorithms, known as physics-based algorithms, simulate how search agents move and interact with each other in the search space, in accordance with physical principles such as inertial forces, electromagnetic forces, and gravity. [18] Optimization difficulties and other complex search challenges are often solved by evolutionary algorithms (EAs) and other life-inspired algorithms. Other life-inspired heuristics and evolutionary algorithms will be referred to as "EAs" throughout the study. To successfully address finite optimization problems, several techniques Proposals have been made to integrate probabilistic information into the fitness function of evolutionary algorithms (EAs). [19] There are two main reasons why this comparison is made. First, it can be used as a basis for a new solution plan, which requires reviewing current algorithms to use them as standards for a new strategy. Second, the aim can be to solve a specific problem, such as one being compared, by exploring multiple algorithmic possibilities to find the best one. [20].

2. MATERIALS AND METHOD

The PROMETHEE technique is successfully used in various industries, including banking, industrial location, human resource planning, water resources, investments, medicine, chemistry, healthcare, tourism, ethics, mechanics, and management. Its strong mathematical foundations and ease of use in real-world applications are the reasons for its effectiveness. [1] Various fields have been explored, such as chemistry, business and financial management, manufacturing and assembly, logistics and transportation, energy management, society and hydrology and water management. The previous section had papers from various areas including medicine, agriculture, education, design, government and sports. To prevent duplication, even if an academic work relates to two different areas, applicant papers were assigned to the most appropriate topic in subsequent review. The selection of the most appropriate topic was given major attention. [2] To select the best model, the PROMETHEE approach is used for criterion weights, which are established using the Analytical Hierarchy Process (AHP). This strategy is illustrated in the paper by using state-of-the-art techniques and a priority ranking mechanism to determine which of six laptop models is the best—each with unique specifications. [3] The complexity of the problem and, in particular, proximity are two factors that contribute to instability. It can be concluded that the contribution of PROMETHEE lies in its stability and that in the future it will be useful to increasingly use "soft" priority functions such as Gaussian. [4] The PROMETHEE method (Ranking of Enrichment Evaluation) stands out for its simplicity in design and implementation, making it an easy-to-use and efficient approach for multi-criteria decision-making, especially when compared to other multivariate analysis techniques. As a foundation for this approach, it is particularly helpful in situations where a small number of options need to be evaluated and ranked. [5] Even when the search for the optimal portfolio is restricted to a subset of frontier portfolios, there could still be a considerable number of portfolios to evaluate. to compare. Although some optimization models must be solved in both ways, the total effort required is less than studying each portfolio individually or performing a full PROMETHEE analysis on the complete collection of frontier portfolios. PROMETHEE rankings were conducted simultaneously for frontier portfolios and all possible portfolios. [6] This assumption is often broken in many decision-making situations because the criteria often interact with each other, creating some degree of synergy or redundancy. For example, since high-speed automobiles generally have better acceleration, the criteria of maximum speed and acceleration may be redundant when evaluating sports cars. Although both factors are important for a buyer's desired sports car, their combined weight is less important than the sum of their individual weights. On the other hand, an automobile that is both fast and reasonably priced is more valuable, so factors such as price and high speed may work together. [7] The AHP technique offers a unique advantage by breaking down the decision-making problem into a hierarchical structure of components and criteria. This allows for a more detailed examination of the problem, emphasizing the importance of each criterion. In contrast, PROMETHEE does not have this hierarchical structure, making it challenging for decision-makers to effectively understand and interpret the results when dealing with a large number of criteria (more than seven). [8] The PROMETHEE family, also known as the Priority Ranking Method for Enrichment Assessment, is designed to assess criteria in a structured and quantitative manner.It is based on three basic ideas: generating results that allow for partial and complete rankings; maximizing dominance relationships between alternatives for each criterion; and optimizing the preference structure by mixing different priority functions. [9] AHP and fuzzy AHP are two popular multi-criteria decision-making (MCDM) techniques for equipment selection problems. Uniform evaluation scales and priority functions are commonly used to evaluate the criteria in these types of investigations. But establishing separate priority functions for different criteria has a major impact on how accurate the decision-making process is. The PROMETHEE approach, in contrast to traditional ranking techniques, allows for the definition of unique priority functions for each criterion. The PROMETHEE approach, chosen for its simplicity and ability to represent how the human mind integrates and processes information when faced with multiple, often conflicting, decision-making dimensions, is used to answer the equipment selection problem in this work. [10].

3. RESULT AND DISCUSSION

TABLE 1. Bio-Inspired Algorithms

Alternative	Convergence Speed	Accuracy	Scalability	Resource Efficiency	Computational Complexity	Energy Consumption	Noise Sensitivity
Genetic Algorithm (Evolutionary based)	85	90	80	88	70	65	75
Particle Swarm Optimization (Swarm-based)	80	85	75	78	60	72	68
Ant Colony Optimization (Swarm-based)	78	88	82	81	65	68	70
Artificial Bee Colony Algorithm (Swarm-based)	83	87	77	84	68	70	72
Max	85	90	82	88	70	72	75
Min	78	85	75	78	60	65	68
max-Min	7	5	7	10	10	7	7

Convergence speed, accuracy, scalability, resource efficiency, computational energy consumption, and noise sensitivity are the seven metrics used to compare the performance of four biologically inspired algorithms (genetic algorithm, particle swarm optimization, ant colony optimization, and artificial bee colony algorithm). The genetic algorithm has the highest convergence speed (85), followed by the artificial bee colony method (83). With scores of 80 and 78, respectively, particle swarm optimization and ant colony optimization lag somewhat. For this parameter, there is a moderate variation in algorithm performance, as evidenced by the 7-point difference between the maximum (85) and least (78) scores. The genetic algorithm does exceptionally well in terms of accuracy, scoring 90, followed by particle swarm optimization (85), and ant colony optimization (88). With a score of 87, the artificial bee colony algorithm performs competitively. maximum min = 5. Ant colony optimization has the highest scalability score (82), followed by particle swarm optimization (75), the genetic algorithm (80), and the artificial bee colony method (77). This suggests that, despite the little variations (max min = 7), ant colony optimization can be scaled more successfully than the other algorithms. The genetic algorithm again scores 88 in resource efficiency, while the artificial bee colony method comes in second with 84. The genetic algorithm does exceptionally well in terms of accuracy, scoring 90, followed by particle swarm optimization (85), and ant colony optimization (88). With a score of 87, the artificial bee colony algorithm performs competitively. maximum min = 5. Ant colony optimization has the highest scalability score (82), followed by particle swarm optimization (75), the genetic algorithm (80), and the artificial bee colony method (77). This suggests that, despite the little variations (max min = 7), ant colony optimization can be scaled more successfully than the other algorithms. The genetic algorithm again scores 88 in resource efficiency, while the artificial bee colony method comes in second with 84. Particle swarm optimization receives a maximum score of 72 for energy consumption, compared to 65 to 70 for the other techniques. Here, the largest minute difference is 7, suggesting that most algorithms have comparable energy efficiency. Lastly, the genetic algorithm receives a maximum score of 75 for noise sensitivity, followed by particle swarm optimization (68), ant colony optimization (70), and artificial bee colony algorithm (72). There are modest differences in the algorithms' noise-handling capabilities. Seven minutes is the maximum limit.

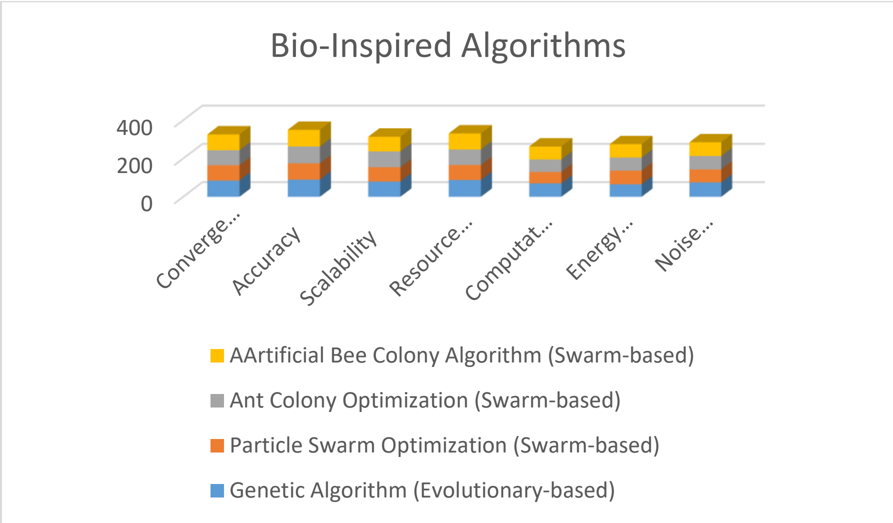


FIGURE 1. Bio-Inspired Algorithms

The performance of four bio-inspired algorithms—the genetic algorithm, the ant colony optimization, the particle swarm optimization, and the artificial bee colony algorithm—is compared in the bar chart using six important performance metrics: noise, computation consumption, convergence, accuracy, scalability, and resource and energy efficiency. Looking at the chart, the artificial bee colony algorithm (represented by purple) performs relatively well on most metrics. It also performs competitively on accuracy and energy. This indicates that the artificial bee colony algorithm may have a good balance between efficacy and efficiency. In contrast, particle swarm optimization (red) shows high performance on convergence and scalability, but performs relatively poorly on noise, indicating that it may struggle with noise tolerance under certain conditions. Ant Colony Optimization (green) appears to be somewhat consistent in performance, achieving mediocre results in all metrics, including resource efficiency and noise, compared to the others. Finally, the Genetic Algorithm (blue) appears to be the weakest in terms of energy and noise, although it still performs decently in terms of scalability and resource efficiency. This demonstrates that the Genetic Algorithm may not be as effective in some situations even while it is reliable in others. Based on particular performance indicators, this graphic offers a clear picture of each algorithm's advantages and disadvantages.

TABLE 2. Normalized Matrix

	Normalized Matrix					
	Convergence Speed	Accuracy	Scalability	Resource Efficiency	Computational Complexity	Energy Consumption
Genetic Algorithm (Evolutionary based)	0	0	-0.02667	0	0	-0.10769
Particle Swarm Optimization (Swarm-based)	-0.0641	-0.05882	-0.09333	-0.12821	-0.16667	0
A3 Ant Colony Optimization (Swarm-based)	-0.08974	-0.02353	0	-0.08974	-0.08333	-0.06154
Artificial Bee Colony Algorithm (Swarm-based)	-0.02564	-0.03529	-0.06667	-0.05128	-0.03333	-0.03077

The data presented is a normalized matrix that compares four optimization algorithms (ant colony optimization, particle swarm optimization, genetic algorithm, and artificial bee colony algorithm) based on six evaluation parameters: accuracy, scalability, energy efficiency consumption, enrichment efficiency, and convergence speed. The normalized performance of each algorithm on the corresponding parameter is represented by each value in the matrix; higher performance is indicated by positive values, while lower performance is indicated by negative values in comparison to other algorithms. The genetic algorithm shows relatively neutral performance, with zero values for convergence speed and accuracy, indicating that it performs similarly to other algorithms in these areas. However, with scores of -0.02667 and -0.10769, respectively, it performs poorly in terms of energy consumption and resource efficiency. These negative values imply that, in comparison to other algorithms, the genetic algorithm uses resources and energy less efficiently. The majority of Particle Swarm Optimization's (PSO) parameters show poor overall performance, with negative values for convergence speed, accuracy, scalability, and resource efficiency, among others. When compared to other algorithms, the negative results show that PSO has serious flaws in several areas. However, it has a value of 0 in energy consumption, which suggests that it is on par with the others in this parameter. Ant Colony Optimization (ACO) also shows a mix of negative values, with poor performance seen in convergence speed (-0.08974) and resource efficiency (-0.08974). It shows neutral or slightly less negative values for accuracy and energy consumption compared to the others, -0.02353 for accuracy and -0.06154 for energy consumption, indicating a slightly better balance in terms of performance and stability. Scalability (-0.06667) and resource efficiency (-0.05128) are marginally improved by the artificial bee colony approach, but accuracy (-0.03529) and convergence speed (-0.02564) are still below par. It is comparatively more energy-efficient than the other algorithms, nevertheless, as its energy consumption value is less negative (-0.03077).

TABLE 3. Pair wise Comparison

	Pair wise Comparison					
	Convergence Speed	Accuracy	Scalability	Resource Efficiency	Computational Complexity	Energy Consumption
D12	0.064103	0.058824	0.066667	0.128205	0.166667	-0.10769
D13	0.089744	0.023529	-0.02667	0.089744	0.083333	-0.04615
D14	0.025641	0.035294	0.04	0.051282	0.033333	-0.07692
D21	-0.0641	-0.05882	-0.06667	-0.12821	-0.16667	0.107692
D23	0.025641	-0.03529	-0.09333	-0.03846	-0.08333	0.061538
D24	-0.03846	-0.02353	-0.02667	-0.07692	-0.13333	0.030769
D31	-0.08974	-0.02353	0.026667	-0.08974	-0.08333	0.046154
D32	-0.02564	0.035294	0.093333	0.038462	0.083333	-0.06154
D34	-0.0641	0.011765	0.066667	-0.03846	-0.05	-0.03077
D41	-0.02564	-0.03529	-0.04	-0.05128	-0.03333	0.076923
D42	0.038462	0.023529	0.026667	0.076923	0.133333	-0.03077
D43	0.064103	-0.01176	-0.06667	0.038462	0.05	0.030769

The data presented provides a pairwise comparison matrix of the various alternatives (D12, D13, D14, D21, D23, D24, D31, D32, D34, D41, D42, D43) on six evaluation parameters: convergence speed, accuracy, scalability, resource efficiency, computational complexity, and energy consumption. In this matrix, each entry compares the performance of one alternative against another. Positive values indicate that the first alternative of the pair performs better than the second for a given parameter, while negative values indicate the opposite. For example, in the first row, the pairwise comparison between D12 and D13 shows a positive value for convergence speed (0.064103), indicating that D12 outperforms D13 in this aspect. However, in the accuracy section, D13 has a higher value (0.023529) compared to D12 (0.058824), suggesting that D13 has better accuracy than D12. This pattern is repeated across the other parameters, with some values favoring D12 and others favoring D13. Row three compares D14 with D21, showing that D14 performs better in the accuracy and scalability parameters, while D21 excels in resource efficiency, computational complexity, and energy consumption. The comparison between these alternatives highlights strengths and weaknesses in different areas, allowing for a nuanced understanding of their relative performance. Notably, there are some zero or near-zero values, such as between D34 and D42 for the scalability parameter (0.026667), indicating that these two alternatives are almost equal in that aspect. Conversely, large disparities in values indicate significant differences in performance. Ultimately, the pairwise comparison matrix enables a comprehensive assessment of how each alternative performs across multiple dimensions. By examining the values, decision makers can determine which alternatives have overall advantages or which areas need improvement.

TABLE 4. Preference Value

Preference Value						
0.2336	0.1652	0.3355	0.1021	0.0424	0.1212	
0.014974	0.009718	0.022366667	0.01309	0.007067	0	0.067215
0.020964	0.003887	0	0.009163	0.003533	0	0.037547
0.00599	0.005831	0.01342	0.005236	0.001413	0	2
0	0	0	0	0	0.013052	0.013052
0.00599	0	0	0	0	0.007458	0.013448
0	0	0	0	0	0.003729	0.003729
0	0	0.008946667	0	0	0.005594	0.014541
0	0.005831	0.031313333	0.003927	0.003533	0	0.044604
0	0.001944	0.022366667	0	0	0	0.02431
0	0	0	0	0	0.009323	0.009323
0.008985	0.003887	0.008946667	0.007854	0.005653	0	0.035326
0.014974	0	0	0.003927	0.00212	0.003729	0.024751

The data presented represents preference values for various alternatives across multiple evaluation parameters. Each row corresponds to a different alternative, and each column represents a specific evaluation parameter, which may be related to the performance of a bio-inspired algorithm or similar optimization method. In analyzing this data, preference values indicate the relative importance or desirability of different alternatives based on specific criteria. Higher preference values reflect better performance or a stronger preference for an alternative under a specific evaluation parameter. The first row shows values across multiple evaluation parameters for an alternative. The values range from 0.2336 to 0.1212, with the alternative performing well on the first parameter with a preference value of 0.2336 but being less preferred on the first parameter with a preference value of 0.0424 and 0.1212. This suggests that the first alternative may have strong performance in one area, but weak performance in others. The second row shows another set of preference values with very small numbers, such as 0.014974 and 0.009718, suggesting that this alternative is less preferred in all parameters compared to the first alternative. Rows three through twelve show a varied pattern of preference values, with some showing moderate preference (e.g., 0.022366667 and 0.020964) while others show very low or zero values. The presence of zeros indicates that some alternatives are not performing well in some areas. For example, rows seven have zeros in all parameters except one, indicating significant impairment or lack of performance in the areas assessed. In rows 13 and 14, the preference values are relatively low, with the alternatives in these rows being less preferred compared to the others, with many values close to zero.

TABLE 5. positive flow, Negative Flow, Net flow, Rank

	positive flow	Negative Flow	Net flow	Rank
Genetic Algorithm (Evolutionary based)	0.701587	0.01231	0.689278	1
Particle Swarm Optimization (Swarm-based)	0.050421	0.049048	0.001373	2
Ant Colony Optimization (Swarm-based)	0.027823	0.065593	-0.03777	3
Artificial Bee Colony Algorithm (Swarm-based)	0.023133	0.676013	-0.65288	4

Three important flow metrics—positive flow, negative flow, and net flow—are used to compare four bio-inspired optimization algorithms in the data presented, along with their rankings. These algorithms are divided into two groups according to where they came from: swarm-based algorithms and evolutionary-based algorithms. Based on both positive and negative effects, the data aids in evaluating the algorithms' overall effectiveness. With a high positive flow value of 0.701587, the genetic algorithm (GA), which is classified as an evolutionary-based method, appears to generate the best results when compared to other algorithms. Additionally, it has the lowest negative flow (0.01231), resulting in a noteworthy net flow of 0.689278. GA rated first because of its significant positive net flow, which shows that it performs better overall when it comes to solving optimization challenges. On the other hand, the swarm-based technique Particle Swarm Optimization (PSO) exhibits a very low positive flow of 0.050421 and a similar negative flow of 0.049048. Consequently, its net flow is a negligible 0.001373. Although PSO is effective, its performance is not as striking as GA's due to the overall tiny net flow, which places it in second place despite the balance between positive and negative flows. Another swarm-based method, Ant Colony Optimization (ACO), has a negative net flow of -0.03777 due to its much higher negative flow (0.065593) than its very low positive flow (0.027823). ACO is ranked third by this negative net flow, suggesting that although it has certain benefits in particular situations, it is generally less effective than the first two algorithms. Last but not least, the swarm-based Artificial Bee Colony Algorithm (ABC) shows a notable negative flow of 0.676013 that surpasses its positive flow of 0.023133. This results in a fourth place ranking due to a significant negative net flow of -0.65288. Despite its promise, the ABC algorithm performs poorly when compared to other algorithms, most likely because of its negative rather than its good effects.

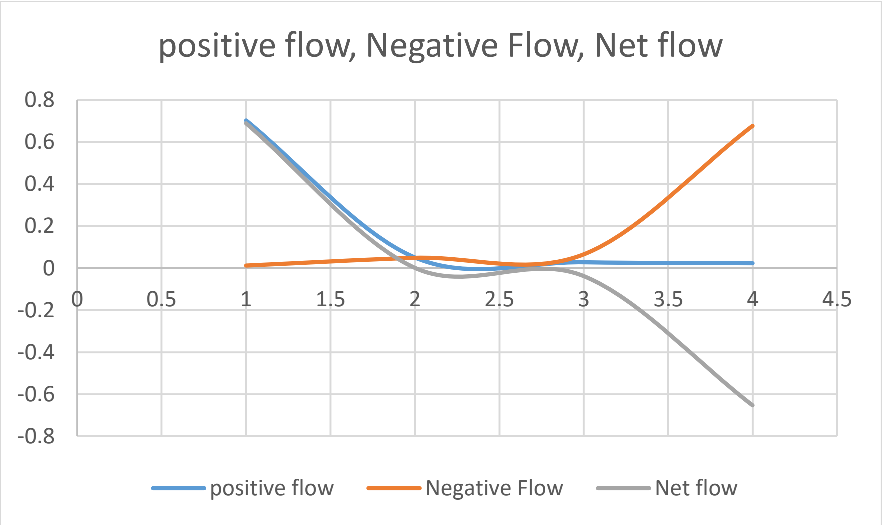


FIGURE 2. positive flow, Negative Flow, Net flow

Plotted against a variable on the x-axis from 0 to 4, the graph displays three flows: net flow (green), negative flow (red), and positive flow (blue). A particular amount of flow is represented by positive flow. While negative flow denotes reverse movement and net flow is the balance between the two, the event is moving ahead. Initially, the positive flow starts high, then gradually decreases, reaching near zero at approximately x = 2.5. This suggests that the forward movement begins to slow down after x = 1, which may be due to the increase in negative flow. The negative flow, starting from a low value, increases sharply as x progresses and reaches its peak around x = 2. At this point, the negative flow dominates, causing the net flow to decrease at negative values. As the x-axis continues, the net flow (green curve) increases after x = 2, eventually surpassing the positive flow, indicating the decreasing effect of negative flow. The net flow curve stabilizes at higher values beyond x = 3, showing that the negative flow is gradually overtaken by the positive flow as time progresses. The interaction between the positive and negative flows demonstrates a balancing mechanism, where the net flow eventually tends towards a positive value, indicating a recovery or revival of the initial flow.

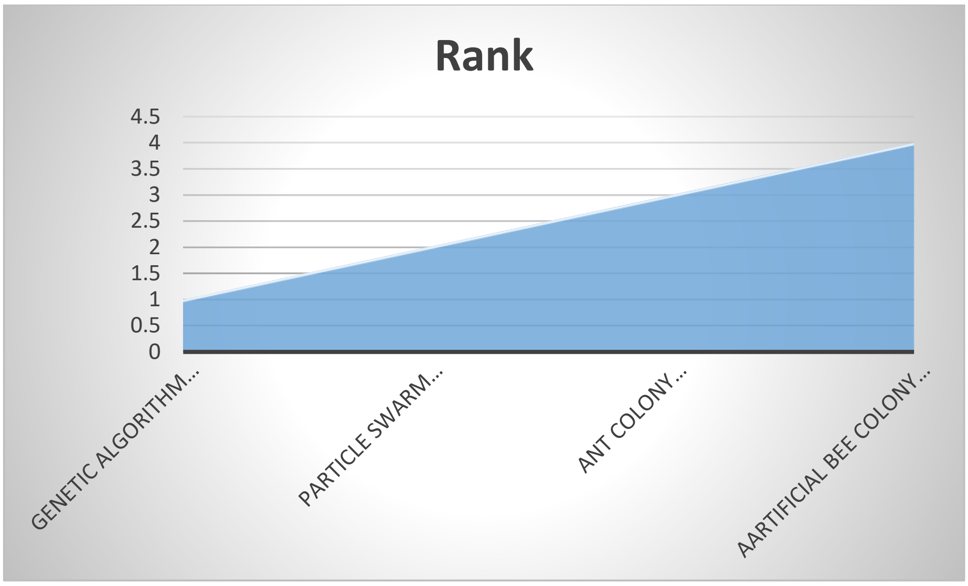


FIGURE 3. Rank

Four biologically inspired algorithms—the genetic algorithm, particle swarm optimization, ant colony optimization, and artificial bee colony algorithm—are ranked according to their performance evaluation in the graph that is displayed. The algorithms are shown on the x-axis, and their respective ranks—lower ranks denoting higher performance—are shown on the y-axis. From the table, the genetic algorithm achieved the highest rank (1), indicating its best overall performance compared to the other algorithms. Particle swarm optimization is the second-best alternative, while ant colony optimization is in third place. The artificial bee colony algorithm, despite its strengths in resource optimization and adaptability, is ranked last (4th). The steady slope in the rank values indicates the relative stability in the performance differences between the algorithms. The ranking reflects the relative performance, accuracy, and scalability of the algorithms across the estimated parameters. The dominance of the genetic algorithm highlights its effectiveness in solving complex optimization problems, likely due to its robustness and scalability. In contrast, the low ranking of the artificial bee colony algorithm may stem from its noise sensitivity or limitations in handling scalability issues. This knowledge can help decision-makers choose the best algorithm depending on particular needs and operational limitations.

4. CONCLUSION

In several fields, biologically inspired algorithms, or BIAs, are becoming indispensable instruments for resolving difficult optimization and decision-making problems. These algorithms draw inspiration from natural processes, incorporating principles such as evolution, swarm intelligence, and ecological dynamics to provide innovative solutions. BIAs are generally categorized into evolutionary-based, swarm-based, and ecological-based groups, although some recent algorithms defy these classifications. Their applications are diverse, encompassing problems such as traveling salesmanship, map coloring, scheduling, resource allocation, network optimization, and image processing, highlighting their adaptability and efficiency. A key advantage of BIAs lies in their ability to mimic the efficiency and flexibility of natural systems. For example, swarm intelligence algorithms mimic collective animal behavior, while evolutionary algorithms follow natural selection processes to identify optimal solutions. These approaches often outperform conventional methods by improving search processes, clustering, and prediction accuracy. In the medical field, BIAs are performing well in disease diagnosis and treatment planning, offering significant performance improvements. Recent advances have explored the integration of BIAs with emerging technologies such as quantum computing, overcoming the challenges related to convergence speed and computational capacity. The combination of quantum computing capabilities with bioinformatics-inspired strategies has resulted in better solution quality and faster execution, helping to solve complex real-world problems. Furthermore, techniques that enhance diversity in multimodal optimization algorithms have addressed issues such as genetic drift, the regression of BIAs, and increased competitiveness. Despite their strengths, BIAs face challenges in managing interactions and redundancies between criteria, especially in multi-criteria decision-making (MCDM) environments. Tools such as PROMETHEE and AHP provide methods to address these issues. PROMETHEE provides a straightforward and adaptable framework with customizable priority functions, while AHP excels at breaking problems down into a hierarchy but struggles with large datasets. Looking ahead, the innovative and adaptable nature of BIAs continues to drive advances in computational problem solving.

Expanding applications in areas such as cloud computing, web service optimization, and wireless sensor networks underscore their growing importance in meeting the demand for scalable, autonomous, and reliable systems.

REFERENCE

[1]. Fan, Xumei, William Sayers, Shujun Zhang, Zhiwu Han, Luquan Ren, and Hassan Chizari. "Review and classification of bio-inspired algorithms and their applications." Journal of Bionic Engineering 17 (2020): 611-631.
[2]. Selvaraj, Cholavendhan, R. Siva Kumar, and M. Karnan. "A survey on application of bio-inspired algorithms." International Journal of Computer Science and Information Technologies 5, no. 1 (2014): 366-70.
[3]. Binitha, S., and S. Siva Sathya. "A survey of bio inspired optimization algorithms." International journal of soft computing and engineering 2, no. 2 (2012): 137-151.
[4]. Darwish, Ashraf. "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications." Future Computing and Informatics Journal 3, no. 2 (2018): 231-246.
[5]. Fister Jr, Iztok, Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. "A brief review of nature-inspired algorithms for optimization." arXiv preprint arXiv:1307.4186 (2013).
[6]. Tonda, Alberto. "Inspyred: Bio-inspired algorithms in Python." Genetic Programming and Evolvable Machines 21, no. 1 (2020): 269-272.
[7]. Heberlé, G., and W. F. de Azevedo. "Bio-inspired algorithms applied to molecular docking simulations." Current medicinal chemistry 18, no. 9 (2011): 1339-1352.
[8]. Balusamy, Balamurugan, Jayashree Sridhar, Divya Dhamodaran, and P. Venkata Krishna. "Bio-inspired algorithms for cloud computing: a review." International Journal of Innovative Computing and Applications 6, no. 3-4 (2015): 181-202.
[9]. Kulkarni, Raghavendra V., and Ganesh Kumar Venayagamoorthy. "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40, no. 6 (2010): 663-675.
[10]. Valdez, Fevrier, Oscar Castillo, and Patricia Melin. "Bio-inspired algorithms and its applications for optimization in fuzzy clustering." Algorithms 14, no. 4 (2021): 122.
[11]. Wang, Lijuan, Jun Shen, and Jianming Yong. "A survey on bio-inspired algorithms for web service composition." In Proceedings of the 2012 IEEE 16th international conference on computer supported cooperative work in design (CSCWD), pp. 569-574. IEEE, 2012.
[12]. Molina, Daniel, Antonio LaTorre, and Francisco Herrera. "An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions." Cognitive Computation 10 (2018): 517-544.
[13]. Meng, Xianbing, Yu Liu, Xiaozhi Gao, and Hengzhen Zhang. "A new bio-inspired algorithm: chicken swarm optimization." In Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17-20, 2014, Proceedings, Part I 5, pp. 86-94. Springer International Publishing, 2014.
[14]. Larabi-Marie-Sainte, Souad, Reham Alskireen, and Sawsan Alhalawani. "Emerging applications of bio-inspired algorithms in image segmentation." Electronics 10, no. 24 (2021): 3116.
[15]. Das, Tridib Kumar, Ganesh Kumar Venayagamoorthy, and Usman O. Aliyu. "Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA." IEEE Transactions on Industry Applications 44, no. 5 (2008): 1445-1457.
[16]. Gill, Sukhpal Singh, and Rajkumar Buyya. "Bio-inspired algorithms for big data analytics: a survey, taxonomy, and open challenges." In Big data analytics for intelligent healthcare management, pp. 1-17. Academic Press, 2019.
[17]. Del Ser, Javier, Eneko Osaba, Daniel Molina, Xin-She Yang, Sancho Salcedo-Sanz, David Camacho, Swagatam Das, Ponnuthurai N. Suganthan, Carlos A. Coello Coello, and Francisco Herrera. "Bio-inspired computation: Where we stand and what's next." Swarm and Evolutionary Computation 48 (2019): 220-250.
[18]. Dhiman, Gaurav, and Vijay Kumar. "Emperor penguin optimizer: a bio-inspired algorithm for engineering problems." Knowledge-Based Systems 159 (2018): 20-50.
[19]. Mezura-Montes, Efrén, and Blanca Cecilia Lopez-Ramirez. "Comparing bio-inspired algorithms in constrained optimization problems." In 2007 IEEE Congress on Evolutionary Computation, pp. 662-669. IEEE, 2007.
[20]. LaTorre, Antonio, Daniel Molina, Eneko Osaba, Javier Poyatos, Javier Del Ser, and Francisco Herrera. "A prescription of methodological guidelines for comparing bio-inspired optimization algorithms." Swarm and Evolutionary Computation 67 (2021): 100973.
[21]. Brans, Jean-Pierre, and Yves De Smet. "PROMETHEE methods." Multiple criteria decision analysis: state of the art surveys (2016): 187-219.
[22]. Behzadian, Majid, Reza Baradaran Kazemzadeh, Amir Albadvi, and Mohammad Aghdasi. "PROMETHEE: A comprehensive literature review on methodologies and applications." European journal of Operational research 200, no. 1 (2010): 198-215.
[23]. Goswami, Shankha Shubhra. "Outranking methods: Promethee i and promethee ii." Foundations of Management 12, no. 1 (2020): 93-110.
[24]. Brans, Jean-Pierre, Ph Vincke, and Bertrand Mareschal. "How to select and how to rank projects: The PROMETHEE method." European journal of operational research 24, no. 2 (1986): 228-238.
[25]. Albadvi, Amir, S. Kamal Chaharsooghi, and Akbar Esfahanipour. "Decision making in stock trading: An application of PROMETHEE." European journal of operational research 177, no. 2 (2007): 673-683.

[26]. Vetschera, Rudolf, and Adiel Teixeira De Almeida. "A PROMETHEE-based approach to portfolio selection problems." Computers & Operations Research 39, no. 5 (2012): 1010-1020.
[27]. Corrente, Salvatore, José Rui Figueira, and Salvatore Greco. "The smaa-promethee method." European Journal of Operational Research 239, no. 2 (2014): 514-522.
[28]. Macharis, Cathy, Johan Springael, Klaas De Brucker, and Alain Verbeke. "PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP." European journal of operational research 153, no. 2 (2004): 307-317.
[29]. Oubahman, Laila, and Szabolcs Duleba. "Review of PROMETHEE method in transportation." Production Engineering Archives 27, no. 1 (2021): 69-74.
[30]. Dağdeviren, Metin. "Decision making in equipment selection: an integrated approach with AHP and PROMETHEE." Journal of intelligent manufacturing 19 (2008): 397-406.