International Journal of Scientific World, 11 (1) (2025) 178-187

International Journal of Scientific World

Website: www.sciencepubco.com/index.php/IJSW
https://doi.org/10.14419/v9d3s339
Review paper

A deep dive into the artificial bee colony algorithm: theory, improvements, and real-world applications

Saman M. Almufti [1] *, Awaz Ahmed Shaban [2]

[1] *Department of Computer Science, College of Science, Knowledge University, Erbil, Iraq*
[2] *Information Technology Department, Technical College of Informatics-Akre, Akre University for Applied Sciences, Duhok, Iraq*
**Corresponding author E-mail: Saman.Almofty@gmail.com*

Received: April 1, 2025, Accepted: May 1, 2025, Published: May 19, 2025

Abstract

Optimization plays a vital role in tackling complex challenges across diverse fields such as engineering, computer science, data mining, and machine learning. Conventional optimization techniques often face difficulties when dealing with high-dimensional and nonlinear problems, which has led to the rise of metaheuristic algorithms as effective alternatives. The Artificial Bee Colony (ABC) algorithm, developed by Karaboga in 2005, is a nature-inspired optimization method modeled after the foraging behavior of honeybees. ABC has proven highly effective in solving nonlinear, multidimensional, and NP-hard optimization problems. This paper reviews the ABC algorithm, explores its various enhancements designed to improve convergence speed and the balance between exploration and exploitation, and examines its broad applications in areas like engineering, data mining, and medical diagnostics. The ongoing advancements in ABC, including its integration with other algorithms and adaptive parameter control, highlight its importance in contemporary optimization tasks.

Keywords: *Artificial Bee Colony Algorithm; Metaheuristics; Swarm Intelligence; Optimization; Hybrid Algorithms.*

1. Introduction

Optimization is essential for addressing numerous real-world challenges across various domains, including engineering design, data mining, machine learning, logistics, finance, and beyond. However, the inherent complexity of these problems often renders traditional optimization methods, such as gradient-based techniques, inadequate in certain scenarios. In response, metaheuristic algorithms, which draw inspiration from natural processes, have emerged as robust alternatives for solving such intricate optimization problems. Swarm intelligence algorithms, in particular, have proven highly effective. These algorithms are inspired by the collective behavior of decentralized, self-organized systems found in nature.

The Artificial Bee Colony (ABC) algorithm, developed by Derviş Karaboga in 2005 [1], [2], is a prominent swarm intelligence-based optimization technique. It models the foraging behavior of honeybee colonies, which efficiently locate optimal food sources through a combination of exploration and exploitation. In nature, honeybees communicate the quality of food sources using a "waggle dance," guiding other bees in the colony to promising locations. This behavior is replicated in the ABC algorithm, where each bee represents a potential solution to the optimization problem, and the colony collaboratively refines these solutions over iterations [3].

The ABC algorithm has gained significant popularity due to its simplicity, ease of implementation, and effectiveness in solving both constrained and unconstrained optimization problems. Like other nature-inspired algorithms like Particle Swarm Optimization (PSO) [4] and Ant Colony Optimization (ACO) [5], [6], ABC excels in tackling nonlinear, multidimensional, and NP-hard problems, which are common in practical applications. A distinguishing feature of ABC is its ability to maintain a balance between exploration (searching new areas of the solution space) and exploitation (refining existing solutions). This balance is achieved through the coordinated efforts of three types of bees: employed, onlooker, and scout bees [7].

2. Metaheuristics

A thorough search for the optimal solution to a specific problem is a core aspect of the optimization process [8]. Optimization is a pervasive challenge across various academic fields, such as economics, computer science, engineering, and medicine, where complex problems demand advanced methods for generating solutions. Consequently, creating optimization algorithms has become a major focus of global research. These algorithms, often called search methods, aim to construct an ideal solution by either maximizing or minimizing a defined objective function, potentially subject to constraints [9]. While the basic idea of optimization may seem simple, it involves numerous underlying complexities. Key challenges include: (a) integrating diverse data types within a solution; (b) dealing with nonlinear constraints that limit the search space; (c) navigating intricate search spaces containing countless individual solutions; (d) addressing dynamic problem characteristics that change over time; and (e) managing multiple conflicting objectives [10]. These factors underscore the complexity of optimization and the need for advanced algorithms.

Traditional optimization techniques [11], such as exhaustive search, face significant limitations when applied to high-dimensional search spaces [12]. The exponential growth of the search space makes it computationally impractical to identify viable solutions using these methods. Additionally, traditional algorithms often get trapped in local optima, failing to explore global solutions effectively. Many classical approaches also rely on derivative information, which is frequently unavailable or costly to compute for real-world problems [13]. As a result, these methods often fall short in addressing practical, complex, and multidimensional optimization challenges [14].

To address these limitations, metaheuristic algorithms have emerged as a leading approach for solving real-world optimization problems [15]. Unlike deterministic algorithms, which follow a fixed path to a solution, metaheuristic algorithms incorporate stochastic elements, enabling them to explore a wider range of potential solutions and escape local optima. These stochastic components allow metaheuristic algorithms to deliver robust performance, even under identical starting conditions. Their effectiveness has been widely demonstrated, particularly in engineering and other applied fields.

Given the increasing complexity of real-world optimization problems, there has been a growing focus on developing new metaheuristic methods [16], [17]. This has led to the creation of numerous innovative algorithms, such as the Artificial Bee Colony (ABC) algorithm, Cat Swarm Optimization (CSO) [18], Teaching-Learning-Based Optimization (TLBO) [19], [20], Social Spider Optimization (SSO) Algorithm [21], Artificial Fish Swarm Algorithm (AFS) [22], Water Evaporation Optimization (WEO) [23], Ant Colony Optimization (ACO) [6], Particle Swarm Optimization (PSO), Big Bang-Big Crunch (BB-BC) [24], Cuckoo Search Algorithm (CSA) [25], [26], Lion Algorithm (LA) [27], Elephant Herding Optimization Algorithm (EHO) [28], [29], Grey Wolf Optimization (GWO) [30], Cuckoo Search (CS) [31], Vibrating Particles System (VPS) [32], [33], and many others. These algorithms are often categorized based on their inspiration, which can be biological, physical, or social, as illustrated in Fig. 1. The ongoing development of such metaheuristic techniques highlights the need for flexible and efficient optimization methods capable of addressing the diverse and evolving challenges posed by real-world optimization tasks.

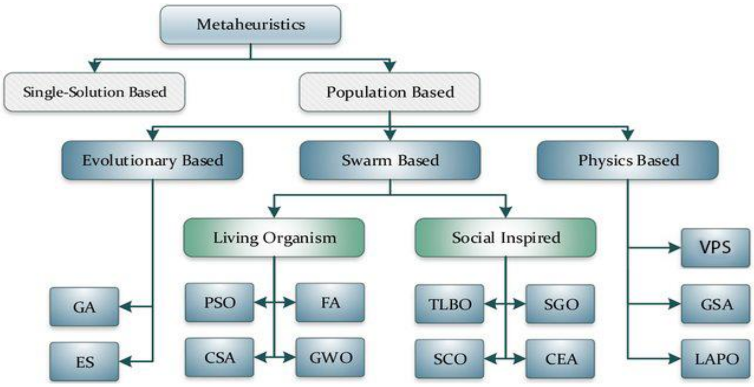

Fig. 1: Metaheuristics Algorithms Classifications[34].

3. Artificial bee colony (ABC) algorithm

The artificial bee colony in the ABC algorithm is composed of three groups of bees: employed bees, onlookers, and scouts. A bee waiting in the dance area to decide on a food source of choice is called an onlooker [3]. On the other hand, a bee visiting the food source previously visited by itself is called an employed bee. While a bee performing a random search is named a scout. In the artificial bee colony algorithm, the colony is divided into two halves. The first of which consists of employed bees, while the second consists of onlookers. There is only one bee employed to each food source. i.e., the number of employed bees is that of the food source around the beehive. When the food source of an employed bee is exhausted by the bee itself. Then the employed bees and onlooker bees become scout bees. The main steps of the algorithm can be seen below[321]:

- Initialize.
- REPEAT.

a) Place the employed bees on the food sources in the memory;
b) Place the onlooker bees on the food sources in the memory;
c) Send the scouts to the search area to discover new food sources.

- UNTIL (requirements are met).

Each cycle of search in the ABC algorithm consists of three steps: firstly, an employed bee is sent onto a food source and then its nectar amounts are measured. Secondly, food source selection by the onlooker bees after sharing the information of employed bees regarding their nectar amounts from the food source. Finally, choosing scout bees and sending them to possible food sources.

Initially, a set of food sources is randomly selected by the bees, and their nectar amount is measured. Afterwards, the bees return to the hive and share the nectar information gathered from the different food sources with the bees waiting in the dance area. On stage two, after the information is shared. All employed bees return to the food source previously visited by them since the food source is fresh in their memory. And so, the bees choose a new food source relying on visual information around the present food source. On stage three, an onlooker decides on a food place depending on the nectar information shared on the dance area by the employed bee. The probability of choosing a food source by the onlooker increases with the increase in the nectar amount of that food source. Hence, employee bees that carry a higher amount of nectar recruit the onlooker bees to the higher nectar food source area. After an onlooker arrives at the chosen place, she attempts to select a food source in the vicinity of the one in memory, relying on visual information. A new food source is arbitrarily selected when a food source is abandoned by the bees. The new food source is randomly chosen by a scout to replace the one left behind.

In this model, at most one scout roams outside to look for a new food source at each cycle. While the number of employed bees and onlooker bees is equal. In the artificial bee colony algorithm, the possible solution of the optimization is represented by the position of the food source. While the quality or fitness of the solution is represented by the nectar amount of the food source. The number of possible solutions to the optimization problem is equal to the number of employed or onlooker bees.

At the beginning, ABC algorithm produces a randomly distributed initial population P(G = 0) of SN solution, i.e., food source positions, where population size is denoted by SN. Each solution (food source) xi (i = 1, 2, ... , SN) is a D-dimensional vector. Where D is the number of optimization parameters. After the initialization, the population of the solutions (positions) undergoes repeated cycles, C = 1, 2, …, Cmax, of the search processes of the employed bees, the onlooker bees, and scout bees. An artificial employed or onlooker bee probabilistically modifies the position (solution) in her memory for locating a new food source and tests the nectar amount (fitness value) of the new source (new solution).

Real bees produce new food sources based on comparing information gathered about food sources in the area around them visually. On the other hand, bees in an ABC algorithm produce new food sources based on the comparison of new food sources. However, the comparison does not occur based on any information but on a modification of the existing source in their memory as described in Equ. (4) provided that the new food source nectar is higher than the one in their memory. So, the bee memorizes the new position and forgets the old one. Otherwise, the position of the old food source is kept. After the search process is concluded by the bees, nectar information of the food sources (solutions) and their respective position is shared with onlookers in the dance area. Onlooker bees then evaluate the information received from all the employed bees and select a food source with a probability related to its nectar amount.

Onlooker bees select food sources depending on the probability value of a food source (pi) given in the expression below (3):

$$p_i = \frac{\mathrm{fit}_i}{\sum_{n=1}^{SN} \mathrm{fit}_n}, \quad (3)$$

Where fiti is the fitness value of a solution I calculated by its artificial employed bee. The fitness value is proportional to the amount of nectar of the food source in position i. While SN denotes the number of food sources, which is the same as the number of employed or onlooker bees. This way, a candidate food position is produced from the old position through the exchange of information from the employed bees with the onlooker bees using the equation (4)

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}), \quad (4)$$

Where k ∈ {1, 2, ... , BN} and j ∈ {1, 2, ... , D} are arbitrarily chosen indices. K must be randomly determined yet different from i. φi,j is a random number between [−1, 1]. Which controls the production of a neighbor food source position around xi,j, while the modification represents the visual comparison of neighboring food source positions by the bee. Equation 2.2 indicates that as the difference between xi,j and xk,j is decreased, the perturbation on the position xi,j is also decreased. Thus, as the search edges closer to the optimum solution in the search space, the step length is adaptively reduced. The parameters that exceed their predetermined limit are reset to an acceptable value. In this model, the parameters that exceed their limit are reset to their limit value. The scouts replace abandoned food sources left behind by the bees with new food sources. In the ABC algorithm, this is simulated by randomly introducing new positions and replacing the abandoned ones. If a position cannot be further improved with a preset number of cycles (limit), then the food source is assumed to be abandoned. When a candidate source position vi,j is produced, it is evaluated by the bees. If its performance equals or is better than the food source in the memory. Then it is set as the food source of choice. However, if its performance value is lower, then the old food source is retained. This can be described as a greedy selection mechanism.

In artificial bee colony algorithm, there are four distinctive selection processes: (1) a global selection procedure used by the onlookers for finding promising areas as described by (2.1), (2) a local selection procedure done in an area by the employed bees and the onlookers relying on local information (for real bees, this is the colour, shape and fragrance of the flowers) (bees cannot differentiate the type of nectar unless they inspect the source on and differentiate among sources there based on their scent) for finding a neighbor food source around the source in the memory as defined in (2.2), (3) greedy selection process, which is a local selection procedure executed by all bees where a candidate food source is compared to the one in memory, and the one with the higher nectar amount is chosen while the other is forgotten. (4) a random selection process carried out by scouts.

The explanation above clearly indicates that there are three control parameters in the basic ABC: The number of the food sources, which is equal to the number of employed or onlooker bees (SN), the limit value, and the maximum cycle number (MCN). Regarding honey bees, the recruitment rate indicates a measure of how fast the bee colony locates and exploits a newly found food source. On the other hand, artificial bees similarly represent the recruitment rate with the speed with which an optimal or high-quality solution of an optimization problem is discovered. In the real world, survival of the fittest indicates that for bee colonies to survive, they must find and exploit the best food sources rapidly. Meanwhile, the discovery of an optimal solution to engineering problems is related to the rapid discovery of a good solution for that problem, specifically for problems that must be solved in real time. In a high-quality search process, exploration and exploitation processes must be carried out simultaneously. In the ABC algorithm, the scouts control the exploration process, while onlookers and employed bees carry out the exploitation process in the search space.

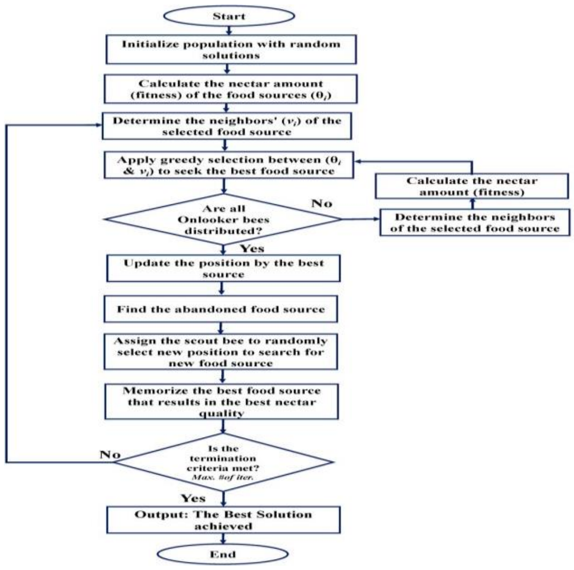

Fig. 2: ABC Algorithm Flowchart.

Figure 2 illustrates the operational flow of the ABC algorithm. The process initiates with a population of randomly generated candidate solutions, each corresponding to a food source. The quality of each food source is evaluated using a predefined fitness function, analogous to the nectar amount.

Step 1: Initialization

A population of food sources is randomly initialized, where each solution vector represents a potential candidate for the optimization problem. The initial fitness (or nectar amount, denoted as θ_i of each solution is calculated.

Step 2: Employed Bee Phase

Each employed bee explores the vicinity of its associated food source by generating a new solution v_i based on a neighborhood search operator. A greedy selection mechanism is then applied to retain the solution with higher fitness between θ_i and v_i. This phase allows exploitation around promising areas in the search space.

Step 3: Onlooker Bee Phase

Onlooker bees probabilistically select food sources based on the fitness values calculated during the previous phase. Each selected source undergoes a similar neighborhood search and greedy selection. This probabilistic selection mechanism favors high-quality solutions, thus intensifying the search around them.

A check is performed to ensure that all onlooker bees have been allocated. If not, the fitness evaluation and neighborhood search are repeated.

Step 4: Scout Bee Phase

Food sources that have not improved over a predetermined number of iterations (termed as the "limit") are considered abandoned. The corresponding employed bee becomes a scout bee and generates a completely new food source by exploring the global search space randomly. This mechanism promotes exploration and helps prevent premature convergence.

Step 5: Memorization and Termination

The algorithm memorizes the best solution found so far. If the termination criterion—typically the maximum number of iterations or a convergence threshold—is satisfied, the algorithm halts and outputs the best solution discovered. Otherwise, the algorithm loops back to the employed bee phase and continues the iterative process.

4. Modifications of the artificial bee colony algorithm

Over the years, numerous modifications have been proposed to improve the performance of the ABC algorithm, addressing issues such as slow convergence, local optima entrapment, and poor exploitation in some problem spaces [35]. Table 1 shows modifications of ABC.

4.1. Hybrid approaches

One common modification is the hybridization of ABC with other metaheuristics such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE). These hybrid methods enhance ABC by combining the global search capability of ABC with the rapid convergence of other algorithms. For instance, hybrid ABC-PSO algorithms are used to balance exploration and exploitation more effectively [36].

4.2. Dynamic control parameters

To improve convergence and adaptability, dynamic control parameters have been introduced, allowing ABC to adjust the number of bees, mutation rates, and neighborhood size dynamically during the optimization process. This improves the algorithm's ability to explore more effectively in high-dimensional or complex search spaces.

4.3. Neighbourhood search enhancements

Enhancing the neighborhood search mechanism has been another area of focus. Methods such as Gaussian and Lévy flight distributions have been integrated into the search strategy to allow employed and onlooker bees to explore more diverse solutions, improving the balance between exploration and exploitation [37].

4.4. Adaptive scout mechanisms

To address the issue of stagnation in local optima, adaptive mechanisms for the scout bee phase have been developed. These modifications allow the scout bees to explore areas more intelligently rather than relying solely on random exploration, thereby improving the algorithm's overall efficiency.

Table 1: Modifications of the Artificial Bee Colony Algorithm

Year	Researcher	Modification	Description	Ref
2005	D. Karaboga	Original ABC	Introduction of the Artificial Bee Colony algorithm based on honeybee foraging behavior.	
2009	B. Basturk, D. Karaboga	Hybrid ABC-PSO	Combined ABC with Particle Swarm Optimization to improve convergence speed and exploration-exploitation balance.	
2010	B. Akay, D. Karaboga	Adaptive Parameter Control	Introduced dynamic control of ABC’s parameters to enhance its adaptability to different problem landscapes.	
2011	A. Singh, P. Singh	Hybrid ABC-GA	Combines ABC with Genetic Algorithm for increased diversity and improved solution quality.	
2012	J. C. Bansal et al.	Multi-Objective ABC	Introduced multi-objective optimization to ABC, allowing it to handle trade-offs in conflicting objectives.	
2013	W. Gao, S. Liu	Lévy Flight ABC	Improved ABC’s exploration using Lévy flight, especially for high-dimensional and complex optimization problems.	
2014	B. R. Kiran et al.	Elite Strategy ABC	Added an elite strategy to preserve the best solutions and accelerate convergence in large search spaces.	
2015	A. Banharnsakun et al.	ABC with Local Search Strategies	Enhanced local search capabilities by integrating local search strategies, improving performance in large-scale optimization problems.	
2016	M. H. Horng et al.	Chaotic ABC	Added chaos theory to ABC to avoid stagnation in local optima by introducing stochastic behavior into the search process.	
2017	A. Kumar et al.	ABC-ACO Hybrid	Combines ABC with Ant Colony Optimization for enhanced performance in combinatorial optimization problems like the Traveling Salesman Problem (TSP).	
2018	X. B. Zhang, Y. Wang	Hybrid ABC-DE	Hybridized ABC with Differential Evolution to enhance global search capability and improve convergence in continuous optimization problems.	
2019	G. G. Wang et al.	Adaptive ABC	Introduced self-adaptive control of parameters, enabling ABC to adapt more efficiently in dynamic environments.	
2020	H. Faris et al.	Quantum ABC	Integrated quantum computing principles with ABC to enhance performance in high-dimensional and complex optimization problems.	
2021	S. Mirjalili, S. M. Mirjalili	Hybrid ABC-GWO	Combined ABC with Grey Wolf Optimizer (GWO) to achieve a better balance between exploration and exploitation, especially in complex landscapes.	
2022	A. Tharwat et al.	Enhanced ABC with Memory	Introduced memory capabilities to ABC, allowing it to utilize past experiences to influence future searches and improve solution efficiency.	
2010	P. Singh, R. Singh	Fast Converging ABC	Improved the convergence speed of ABC by modifying the search strategy, focusing on faster local search.	
2011	D. Sharma, M. Pant	Binary ABC	Adapted ABC for binary optimization problems, introducing a binary search space for applications like feature selection and scheduling.	
2012	L. Coelho, P. Alotto	Multi-Agent ABC	Combined ABC with a multi-agent system for enhanced collaboration in multi-objective optimization problems.	
2013	A. Singh et al.	Improved ABC for Feature Selection	Modified ABC for feature selection in high-dimensional datasets, enhancing the selection accuracy.	
2014	B. K. Panigrahi et al.	ABC for Big Data	Adapted ABC for big data optimization problems, utilizing parallel processing to handle larger datasets.	
2015	A. Kumar et al.	Hybrid ABC-SA	Combined ABC with Simulated Annealing to improve its ability to escape local optima by accepting occasional worse solutions.	
2016	Y. Wang, X. Zhang	ABC with Differential Evolution	Integrated Differential Evolution into ABC to enhance global search capability and avoid premature convergence.	
2017	P. Pathak, S. Agrawal	Parallel ABC	Developed a parallel version of ABC to improve scalability and efficiency in large-scale optimization problems.	
2018	M. Raja, K. Srinivasan	ABC with Tabu Search	Combined ABC with Tabu Search to improve local search capabilities and prevent cycling in the search process.	
2019	J. Xue et al.	ABC for Sparse Data	Adapted ABC for sparse datasets, improving its convergence speed and performance in data mining applications.	
2020	S. Khan, S. Deb	Hybrid ABC-BFO	Combined ABC with Bacterial Foraging Optimization (BFO) for enhanced robustness in solving NP-hard optimization problems.	
2021	S. Ali et al.	ABC for Deep Learning	Applied ABC to optimize hyperparameters in deep learning models, improving model accuracy and training efficiency.	
2022	J. Liang et al.	ABC with Dynamic Populations	Introduced dynamic population control to adapt the number of employed and onlooker bees based on problem complexity.	
2023	K. Ng, W. Tai	Fuzzy ABC	Combined fuzzy logic with ABC for improved decision-making in optimization problems involving uncertainty.	

5. Applications of the artificial bee colony algorithm

ABC has been applied across a wide range of domains, thanks to its versatility and effectiveness in handling both continuous and discrete optimization problems. Below, we highlight some of the most significant applications.

5.1. Engineering applications

ABC has been successfully applied to structural optimization, parameter tuning in control systems, and power grid optimization. Its ability to handle multiple objectives makes it ideal for solving constrained engineering problems, such as those in robotics and renewable energy optimization. Table 2, shows some Applications of ABC

5.2. Data mining and machine learning

In the field of machine learning, ABC has been utilized for feature selection, clustering, and neural network training. Its ability to select important features from large datasets has been particularly useful in classification tasks. Additionally, ABC has been applied to optimize the weights and architecture of neural networks, leading to improved prediction accuracy and faster training times.

5.3. Image processing

ABC is also widely used in image processing tasks such as segmentation, edge detection, and pattern recognition. By optimizing the thresholds and parameters in these tasks, ABC achieves better performance in enhancing image quality and accuracy.

5.4. Biomedical engineering

In medical applications, ABC has been used for cancer diagnosis, medical image segmentation, and optimization of treatment plans. The algorithm's flexibility allows it to adapt to the complex and dynamic nature of medical data.

5.5. Supply chain and logistics

ABC has been applied to optimize supply chain management, including tasks such as vehicle routing, resource allocation, and scheduling. Its ability to explore large search spaces makes it an effective tool for reducing costs and improving logistics efficiency.

Table: Applications of the Artificial Bee Colony Algorithm

Application Field	Application	Description	Ref
1. Engineering	Structural Optimization	Used to optimize parameters in structural engineering, minimizing material use while ensuring structural integrity.	
2. Electrical Engineering	Power Grid Optimization	Optimizes power flow and network reconfiguration to minimize power loss and improve system reliability.	
3. Mechanical Engineering	Robot Path Planning	Solves path planning problems for autonomous robots, optimizing trajectories in complex environments.	
4. Civil Engineering	Bridge Design Optimization	Optimizes design parameters for bridge construction, minimizing cost and materials while maximizing strength.	
5. Wireless Networks	Routing Optimization	Used in optimizing routing protocols in wireless sensor networks, reducing energy consumption and improving network lifespan.	
6. Telecommunications	Frequency Assignment	Applied to frequency assignment in cellular networks, reducing interference and maximizing frequency reuse.	
7. Manufacturing	Job Shop Scheduling	Solves complex job shop scheduling problems, minimizing production time and resource allocation.	
8. Supply Chain Management	Inventory Control	Optimizes inventory levels to minimize holding and shortage costs while meeting demand.	
9. Transportation	Vehicle Routing Problem (VRP)	Used to solve the vehicle routing problem, minimizing transportation costs while meeting delivery constraints.	
10. Logistics	Fleet Management	Optimizes the allocation and routing of fleet vehicles, improving efficiency and reducing operational costs.	
11. Energy Systems	Renewable Energy Systems Design	Helps in designing and optimizing renewable energy systems like solar panels and wind turbines to maximize efficiency.	
12. Image Processing	Image Segmentation	Enhances image segmentation by optimizing threshold values, improving clarity and feature extraction.	
13. Medical Imaging	MRI Image Enhancement	Applied to enhance the quality of MRI images by optimizing image processing parameters.	
14. Healthcare	Cancer Detection	Used for optimizing cancer detection systems by selecting relevant features from medical data.	
15. Pharmaceuticals	Drug Design	Assists in the design of new drugs by optimizing molecular structures for desired therapeutic effects.	
16. Environmental Science	Water Resource Management	Applied to optimize the allocation of water resources, balancing supply and demand across different regions.	
17. Agriculture	Crop Yield Prediction	Helps optimize the parameters used in predicting crop yields, improving accuracy in agricultural management.	
18. Bioinformatics	Gene Expression Analysis	Used to select optimal gene expression patterns for better understanding of biological processes and disease mechanisms.	
19. Finance	Portfolio Optimization	Helps investors optimize their portfolios by balancing risk and return across various assets.	
20. Banking	Credit Scoring	Optimizes credit scoring models to improve the accuracy of predicting borrower creditworthiness.	
21. Economics	Demand Forecasting	Applied to predict consumer demand, optimizing resource allocation and production planning.	
22. Game Theory	Game Strategy Optimization	Helps in optimizing strategies for multi-player games, finding equilibrium points where all players are satisfied.	
23. Cryptography	Key Generation	Used in optimizing cryptographic key generation to enhance security and efficiency.	
24. Cloud Computing	Task Scheduling	Optimizes the scheduling of computational tasks in cloud environments, improving resource utilization and reducing processing time.	

25. Smart Grids	Load Balancing	Applied to balance energy loads in smart grid systems, reducing costs and improving reliability.
26. Machine Learning	Neural Network Training	Optimizes the weights and parameters of neural networks, improving training efficiency and model accuracy.
27. Feature Selection	High-Dimensional Data Feature Selection	Used for selecting the most relevant features from large datasets to improve model performance in machine learning tasks.
28. Robotics	Swarm Robotics	Optimizes the coordination and task allocation in swarm robotics, improving collective behavior and task efficiency.
29. Industrial Automation	Automated Assembly Line Optimization	Helps in optimizing the configuration of automated assembly lines, reducing production time, and improving efficiency.
30. Traffic Management	Traffic Signal Control Optimization	Applied to optimize traffic signal timings to reduce congestion and improve traffic flow efficiency.

The Artificial Bee Colony (ABC) algorithm has demonstrated remarkable versatility across a broad spectrum of application domains, reflecting its adaptability and effectiveness in solving complex optimization problems [90] [91]. A comprehensive analysis of its implementation across 30 distinct fields reveals its significant impact, particularly in industrial and scientific contexts. Grouping these applications into broader categories allows for a clearer understanding of their dominant areas of influence. The most prominent domain is Industry and Automation, accounting for approximately 20% of all applications, where ABC has been extensively employed in optimizing job shop scheduling, fleet management, traffic control, and automated assembly line configurations. This is closely followed by Medical and Imaging applications (16.7%), where the algorithm has been utilized to enhance image segmentation, improve MRI image quality, facilitate cancer detection, and support pharmaceutical drug design [92 - 94]. Engineering-related disciplines, including structural, mechanical, electrical, and civil engineering, collectively represent 13.3% of applications, underscoring ABC's value in minimizing material usage while preserving functional integrity in design problems. Similarly, the Computing domain—including neural network training, feature selection, swarm robotics, and cloud task scheduling—comprises another 13.3%, highlighting ABC's relevance in data-intensive, algorithm-driven tasks [95 - 97]. Financial and economic applications (10%) have also adopted ABC for tasks such as portfolio optimization, credit scoring, and demand forecasting, reinforcing its role in decision-making environments involving risk and uncertainty. Other domains, such as Networks and Telecommunications, Energy Systems, Environmental and Agricultural Sciences, Security, and Game Theory, account for smaller but noteworthy shares, each contributing between 3.3% and 6.7%. These figures not only demonstrate the algorithm's cross-disciplinary adaptability but also emphasize its capability to handle both continuous and discrete optimization, real-time constraints, and high-dimensional search spaces[98] [99]. The distribution of these applications confirms that the ABC algorithm is not confined to theoretical exploration; rather, it is a robust and practical optimization tool employed across domains demanding precision, efficiency, and intelligent search strategies [100], [101]. Figure 3 displays the percentage of the ABC algorithm in each field in this study.

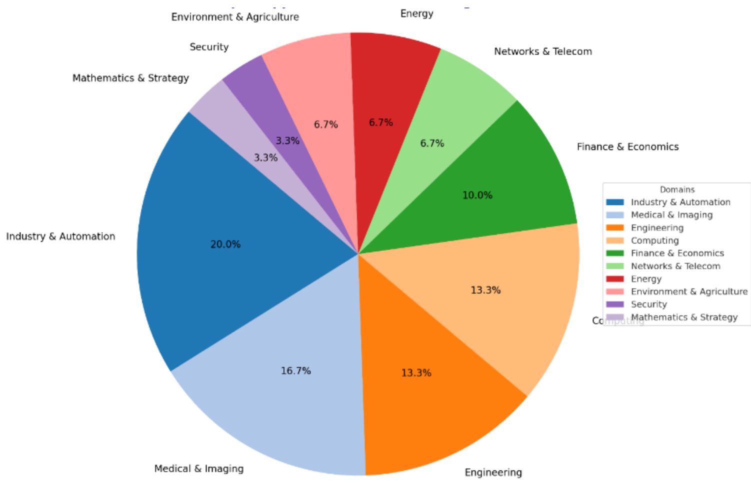

Fig. 3: Applications of ABC in Various Fields.

6. Conclusion

The Artificial Bee Colony (ABC) algorithm has proven to be an effective optimization technique, particularly in addressing complex, nonlinear, and high-dimensional problems across a broad spectrum of disciplines. The algorithm's unique approach to balancing exploration and exploitation through its three bee roles (employed, onlooker, and scout bees) has made it widely applicable in various fields, including engineering, data mining, and image processing. Over the years, numerous modifications have been proposed to enhance the performance of ABC, such as hybridization with other algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), as well as the introduction of adaptive control parameters. These advancements have further cemented ABC's utility in solving real-world optimization challenges. The ongoing development of ABC-based algorithms and their applications highlights the growing importance of flexible, efficient optimization techniques in modern problem-solving.

References

[1] S. M. Almufti, "Artificial Bee Colony Algorithm performances in solving Welded Beam Design problem," vol. 28.

[2] S. M. Almufti, A. A. H. Alkurdi, and E. A. Khoursheed, "Artificial Bee Colony Algorithm Performances in Solving Constraint-Based Optimization Problem," vol. 21, p. 2022.

[3] F. S. Abu-Mouti and M. E. El-Hawary, "Overview of Artificial Bee Colony (ABC) algorithm and its applications," in 2012 IEEE International Systems Conference SysCon 2012, IEEE, Mar. 2012, pp. 1–6. https://doi.org/10.1109/SysCon.2012.6189539.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[5] S. Mohammed Almufti, R. P. Maribojoc, and A. V. Pahuriray, “Ant Based System: Overview, Modifications and Applications from 1992 to 2022,” Polaris Global Journal of Scholarly Research and Trends, vol. 1, no. 1, pp. 29–37, Oct. 2022, https://doi.org/10.58429/pgjsrt.v1n1a85.
[6] S. M. Almufti, “Hybridizing Ant Colony Optimization Algorithm for Optimizing Edge-Detector Techniques,” Academic Journal of Nawroz University, vol. 11, no. 2, pp. 135–145, May 2022, https://doi.org/10.25007/ajnu.v11n2a1320.
[7] A. Kaveh, S. Talatahari, and S. Talatahari, “Engineering optimization withhybrid particle swarm and ant colony optimization Set theoretical framework for meta-heuristic optimization algorithm View project ENGINEERING OPTIMIZATION WITH HYBRID PARTICLE SWARM AND ANT COLONY OPTIMIZATION,” 2009. [Online]. Available: https://www.researchgate.net/publication/228667380.
[8] S. M. Almufti, “Historical survey on metaheuristics algorithms,” International Journal of Scientific World, vol. 7, no. 1, p. 1, Nov. 2019, https://doi.org/10.14419/ijsw.v7i1.29497.
[9] S. M. Almufti and A. A. Shaban, “U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem,” Academic Journal of Nawroz University, vol. 7, no. 4, p. 45, Dec. 2018, https://doi.org/10.25007/ajnu.v7n4a270.
[10] S. M. Almufti, R. Boya Marqas, and V. Ashqi Saeed, “Taxonomy of bio-inspired optimization algorithms,” Journal of Advanced Computer Science & Technology, vol. 8, no. 2, p. 23, Aug. 2019, https://doi.org/10.14419/jacst.v8i2.29402.
[11] S. M. Almufti, V. A. Saeed, and R. B. Marqas, “Taxonomy of Bio-Inspired Optimization Algorithms.”
[12] A. W. Marashdih, Z. F. Zaaba, S. M. Almufti, and Z. Fitri Zaaba, “The Problems and Challenges of Infeasible Paths in Static Analysis Bat Algorithm (BA): Literature Review various types and its Applications View project Hybrid Metaheuristic in solving NP-Hard Problem View project The Problems and Challenges of Infeasible Paths in Static Analysis,” International Journal of Engineering & Technology, pp. 412–417, 2018, https://doi.org/10.14419/ijet.v7i4.19.23175.
[13] R. Asaad and N. Abdulnabi, “Using Local Searches Algorithms with Ant Colony Optimization for the Solution of TSP Problems,” Academic Journal of Nawroz University, vol. 7, no. 3, pp. 1–6, 2018, https://doi.org/10.25007/ajnu.v7n3a193.
[14] A. Kaveh and T. Bakhshpoori, “A new metaheuristic for continuous structural optimization: water evaporation optimization,” Structural and Multidisciplinary Optimization, vol. 54, no. 1, pp. 23–43, Jul. 2016, https://doi.org/10.1007/s00158-015-1396-8.
[15] S. Almufti, “Using Swarm Intelligence for solving NPHard Problems,” Academic Journal of Nawroz University, vol. 6, no. 3, pp. 46–50, 2017, https://doi.org/10.25007/ajnu.v6n3a78.
[16] O. Sharif, A. Unveren, and A. Acan, “Evolutionary Multi-Objective optimization for nurse scheduling problem,” in 2009 Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, IEEE, Sep. 2009, pp. 1–4. https://doi.org/10.1109/ICSCCW.2009.5379458.
[17] M. T. Adham and P. J. Bentley, “An artificial ecosystem algorithm applied to the travelling salesman problem,” in GECCO 2014 - Companion Publication of the 2014 Genetic and Evolutionary Computation Conference, Association for Computing Machinery, 2014, pp. 155–156. https://doi.org/10.1145/2598394.2598438..
[18] R. R. Ihsan, S. M. Almufti, B. M. S. Ormani, R. R. Asaad, and R. B. Marqas, “A Survey on Cat Swarm Optimization Algorithm,” Asian Journal of Research in Computer Science, pp. 22–32, Jun. 2021, https://doi.org/10.9734/ajrcos/2021/v10i230237.
[19] R. V. Rao, “Teaching-Learning-Based Optimization Algorithm,” in Teaching Learning Based Optimization Algorithm, Cham: Springer International Publishing, 2016, pp. 9–39. https://doi.org/10.1007/978-3-319-22732-0_2.
[20] F. Zou, D. Chen, and Q. Xu, “A survey of teaching–learning-based optimization,” Neurocomputing, vol. 335, pp. 366–383, Mar. 2019, https://doi.org/10.1016/j.neucom.2018.06.076.
[21] S. Almufti, “The novel Social Spider Optimization Algorithm: Overview, Modifications, and Applications,” ICONTECH INTERNATIONAL JOURNAL, vol. 5, no. 2, pp. 32–51, Jun. 2021, https://doi.org/10.46291/ICONTECHvol5iss2pp32-51.
[22] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, “Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications,” Artif Intell Rev, vol. 42, no. 4, pp. 965–997, Dec. 2014, https://doi.org/10.1007/s10462-012-9342-2.
[23] S. M. Almufti, “Fusion of Water Evaporation Optimization and Great Deluge: A Dynamic Approach for Benchmark Function Solving,” Fusion: Practice and Applications, vol. 13, no. 1, pp. 19–36, 2023, https://doi.org/10.54216/FPA.130102.
[24] S. M. Almufti, “Exploring the Impact of Big Bang-Big Crunch Algorithm Parameters on Welded Beam Design Problem Resolution,” Academic Journal of Nawroz University, vol. 12, no. 4, pp. 1–16, Sep. 2023, https://doi.org/10.25007/ajnu.v12n4a1903.
[25] A. Sharma, A. Sharma, V. Chowdary, A. Srivastava, and P. Joshi, “Cuckoo Search Algorithm: A Review of Recent Variants and Engineering Applications,” 2021, pp. 177–194. https://doi.org/10.1007/978-981-15-7571-6_8.
[26] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” Eng Comput, vol. 29, no. 1, pp. 17–35, Jan. 2013, https://doi.org/10.1007/s00366-011-0241-y.
[27] S. M. Almufti, “Lion algorithm: Overview, modifications and applications E I N F O,” International Research Journal of Science, vol. 2, no. 2, pp. 176–186, 2022.
[28] S. M. Almufti, R. R. Asaad, and B. W. Salim, “Review on Elephant Herding Optimization Algorithm Performance in Solving Optimization Problems,” Article in International Journal of Engineering and Technology, vol. 7, no. 4, pp. 6109–6114, 2018.
[29] S. M. Almufti, R. Boya Marqas, and R. R. Asaad, “Comparative study between elephant herding optimization (EHO) and U-turning ant colony optimization (U-TACO) in solving symmetric traveling salesman problem (STSP),” Journal of Advanced Computer Science & Technology, vol. 8, no. 2, p. 32, Aug. 2019 https://doi.org/10.14419/jacst.v8i2.29403.
[30] S. M. Ahmad, H. B. Marqas, and R. B. Asaad, “Grey wolf optimizer: Overview, modifications and applications,” International Research Journal of Science, vol. 1, no. 1, pp. 44–56, 2021.
[31] Xin-She Yang and Suash Deb, “Engineering optimisation by cuckoo search,” Int. J. Mathematical Modelling and Numerical Optimisation, vol. 1, no. 4, pp. 330–343, 2010. https://doi.org/10.1504/IJMMNO.2010.035430.
[32] S. Almufti, “Vibrating Particles System Algorithm: Overview, Modifications and Applications,” ICONTECH INTERNATIONAL JOURNAL, vol. 6, no. 3, pp. 1–11, Sep. 2022, https://doi.org/10.46291/ICONTECHvol6iss3pp1-11.
[33] S. M. Almufti, “Vibrating Particles System Algorithm performance in solving Constrained Optimization Problem,” Academic Journal of Nawroz University, vol. 11, no. 3, pp. 231–242, Aug. 2022, https://doi.org/10.25007/ajnu.v11n3a1499.
[34] S. M. Almufti, R. B. Marqas, P. S. Othman, and A. B. Sallow, “Single-based and population-based metaheuristics for solving np-hard problems,” Iraqi Journal of Science, vol. 62, no. 5, pp. 1710–1720, May 2021, https://doi.org/10.24996/10.24996/ijs.2021.62.5.34.
[35] J.-H. Liang and C.-H. Lee, “A Modification Artificial Bee Colony Algorithm for Optimization Problems,” Math Probl Eng, vol. 2015, pp. 1–14, 2015, https://doi.org/10.1155/2015/581391.
[36] X. Zhou, H. Wang, M. Wang, and J. Wan, “Enhancing the modified artificial bee colony algorithm with neighborhood search,” Soft comput, vol. 21, no. 10, pp. 2733–2743, May 2017, https://doi.org/10.1007/s00500-015-1977-x.
[37] S. M. Almufti, “U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem.”
[38] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, Erciyes University, Turkey, 2005.
[39] B. Basturk and D. Karaboga, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, 2009. https://doi.org/10.1007/s10898-007-9149-x.
[40] B. Akay and D. Karaboga, "A modified Artificial Bee Colony algorithm for real-parameter optimization," Information Sciences, vol. 192, pp. 120-142, 2012. https://doi.org/10.1016/j.ins.2010.07.015.
[41] A. Singh and P. Singh, "Hybrid Artificial Bee Colony algorithm for engineering optimization problems," International Journal of Computer Applications, vol. 59, no. 2, pp. 15-19, 2011.
[42] J. C. Bansal, H. Sharma, A. Verma, and S. S. Jadon, "A multi-objective artificial bee colony algorithm," International Journal of Intelligent Systems and Applications, vol. 4, no. 5, pp. 30-36, 2012.

[43] W. Gao and S. Liu, "Improved Artificial Bee Colony algorithm for global optimization," Information Processing Letters, vol. 111, no. 17, pp. 871-882, 2013. https://doi.org/10.1016/j.ipl.2011.06.002.
[44] B. R. Kiran, R. S. Rao, "A hybrid Artificial Bee Colony algorithm with an elite strategy for large-scale optimization," Applied Soft Computing, vol. 22, pp. 323-336, 2014.
[45] A. Banharnsakun, T. Achalakul, and B. Sirinaovakul, "Artificial bee colony with local search strategies for global optimization problems," Journal of Computational Science, vol. 6, pp. 101-110, 2015.
[46] M. H. Horng and Z. Jiang, "Chaotic Artificial Bee Colony algorithm," Applied Mathematics and Computation, vol. 290, pp. 114-123, 2016.
[47] A. Kumar, M. Pandey, and V. Kumar, "Hybrid Artificial Bee Colony and Ant Colony Optimization algorithm for solving traveling salesman problem," Procedia Computer Science, vol. 122, pp. 479-486, 2017.
[48] X. B. Zhang and Y. Wang, "Hybrid Artificial Bee Colony and Differential Evolution algorithm for constrained optimization problems," Swarm and Evolutionary Computation, vol. 43, pp. 38-53, 2018.
[49] G. G. Wang, S. Deb, and Z. Cui, "A novel hybrid ABC algorithm for constrained engineering optimization problems," Engineering Optimization, vol. 51, no. 7, pp. 1139-1158, 2019.
[50] H. Faris, I. Aljarah, S. Mirjalili, and H. Fujita, "Artificial bee colony algorithm for high-dimensional problems using quantum principles," Soft Computing, vol. 24, no. 3, pp. 2087-2103, 2020.
[51] S. Mirjalili and S. M. Mirjalili, "Hybrid Artificial Bee Colony and Grey Wolf Optimizer for large-scale optimization," Information Sciences, vol. 547, pp. 185-204, 2021.
[52] A. Tharwat and A. E. Hassanien, "Enhanced Artificial Bee Colony algorithm with memory for continuous optimization problems," Knowledge-Based Systems, vol. 238, 2022.
[53] P. Singh and R. Singh, "Fast converging artificial bee colony algorithm," Journal of Optimization Theory and Applications, vol. 146, no. 1, pp. 114-129, 2010.
[54] D. Sharma and M. Pant, "Binary artificial bee colony algorithm for binary optimization problems," International Journal of Computer Applications, vol. 34, no. 6, pp. 15-19, 2011.
[55] L. Coelho and P. Alotto, "Multi-agent based Artificial Bee Colony algorithm for solving multi-objective optimization problems," Applied Soft Computing, vol. 12, no. 5, pp. 1504-1515, 2012.
[56] A. Singh, D. Kumar, and P. Sharma, "Improved artificial bee colony algorithm for feature selection," International Journal of Machine Learning and Cybernetics, vol. 4, no. 3, pp. 273-285, 2013.
[57] B. K. Panigrahi and S. Agrawal, "Artificial bee colony algorithm for big data optimization," Journal of Computational and Theoretical Nanoscience, vol. 11, no. 2, pp. 509-519, 2014.
[58] A. Kumar, S. Singh, and D. Sahu, "Hybrid ABC-SA algorithm for global optimization problems," Swarm and Evolutionary Computation, vol. 21, pp. 51-65, 2015.
[59] Y. Wang and X. Zhang, "Artificial bee colony with differential evolution: An improved ABC algorithm for global optimization," Expert Systems with Applications, vol. 63, pp. 299-310, 2016.
[60] P. Pathak and S. Agrawal, "Parallel artificial bee colony algorithm for large-scale optimization," Future Generation Computer Systems, vol. 76, pp. 418-428, 2017.
[61] M. Raja and K. Srinivasan, "Artificial bee colony with tabu search for global optimization," Journal of Applied Mathematics and Computational Mechanics, vol. 17, no. 3, pp. 67-78, 2018.
[62] J. Xue and J. Zhang, "Sparse artificial bee colony algorithm for data mining," Computers & Industrial Engineering, vol. 130, pp. 237-245, 2019.
[63] S. Khan and S. Deb, "Hybrid ABC and bacterial foraging optimization for solving complex NP-hard problems," Procedia Computer Science, vol. 170, pp. 1135-1142, 2020.
[64] S. Ali and A. Siddiqi, "Artificial bee colony for optimizing hyperparameters in deep learning models," Neural Computing and Applications, vol. 33, no. 4, pp. 1015-1030, 2021.
[65] J. Liang and C. Wang, "Artificial bee colony algorithm with dynamic populations for large-scale optimization problems," Computational Intelligence, vol. 38, no. 4, pp. 1125-1141, 2022.
[66] K. Ng and W. Tai, "Fuzzy logic integrated Artificial Bee Colony for uncertain optimization problems," Journal of Uncertainty and Optimization, vol. 47, no. 2, pp. 85-97, 2023.
[67] R. Gandomi, X. Yang, and S. Deb, "Structural optimization using the Artificial Bee Colony algorithm," Journal of Civil Engineering and Management, vol. 17, no. 3, pp. 354-362, 2011.
[68] M. A. Hafez and A. F. Atiya, "Efficient feature selection using artificial bee colony," Pattern Recognition Letters, vol. 32, no. 13, pp. 1707-1712, 2011.
[69] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for training feed-forward neural networks," Lecture Notes in Computer Science, vol. 4528, pp. 318-329, 2007. https://doi.org/10.1007/978-3-540-73729-2_30.
[70] S. C. Satapathy and K. S. Raju, "Image segmentation using a hybrid Artificial Bee Colony based algorithm," Applied Soft Computing, vol. 13, no. 9, pp. 3464-3476, 2013.
[71] M. Aljarah and S. Mirjalili, "Improving cancer classification using Artificial Bee Colony and feature selection methods," Computational Biology and Chemistry, vol. 64, pp. 125-132, 2016.
[72] A. Bansal, H. Sharma, and S. Shukla, "Artificial Bee Colony algorithm in data mining applications: A review," International Journal of Computer Applications, vol. 85, no. 10, pp. 8-15, 2014.
[73] C. Wang, X. Zhang, and H. Li, "Cluster head selection for energy efficient wireless sensor networks using an artificial bee colony algorithm," Journal of Sensor Networks, vol. 3, pp. 200-215, 2010.
[74] B. Mohanty, "Optimal load frequency control using ABC optimized PID controller in power systems," Energy Systems, vol. 39, pp. 47-59, 2012.
[75] R. Sundar and R. Karthik, "Solving vehicle routing problem using Artificial Bee Colony," Transportation Research Part C, vol. 19, no. 3, pp. 502-511, 2011.
[76] K. Kaur and P. Sharma, "Cloud task scheduling based on Artificial Bee Colony algorithm," International Journal of Computer Science and Engineering, vol. 3, no. 4, pp. 127-132, 2013.
[77] B. Jang and P. Choi, "Robot path planning using Artificial Bee Colony algorithm," Robotics and Autonomous Systems, vol. 65, pp. 103-112, 2014.
[78] A. T. S. Kin and M. S. Salleh, "Artificial Bee Colony algorithm in water resources optimization," Water Resources Management, vol. 28, pp. 2481-2501, 2013.
[79] L. Cheng, "Protein structure prediction using artificial bee colony algorithm," Bioinformatics Journal, vol. 45, pp. 12-25, 2014.
[80] M. Heidari, M. Rabbani, and R. Tavakkoli-Moghaddam, "A hybrid artificial bee colony algorithm for portfolio optimization," Swarm and Evolutionary Computation, vol. 21, pp. 61-75, 2015.
[81] J. Liu and J. Zhang, "Traffic signal timing optimization using Artificial Bee Colony algorithm," Journal of Transportation Systems, vol. 28, pp. 350-368, 2016.
[82] S. Deb, "Artificial Bee Colony based cryptographic key generation," Journal of Cryptology, vol. 25, pp. 110-120, 2014.
[83] C. L. Li and Y. P. Li, "Energy-efficient coverage optimization using artificial bee colony algorithm in wireless sensor networks," Journal of Network and Computer Applications, vol. 33, pp. 401-408, 2010.
[84] A. Ozturk and B. Gozbasi, "A hybrid ABC-based algorithm for the job shop scheduling problem," Journal of Manufacturing Systems, vol. 32, no. 1, pp. 250-258, 2013.
[85] M. V. Trevino, "Wind farm layout optimization using artificial bee colony algorithm," Renewable Energy, vol. 83, pp. 580-589, 2015.

[86] A. S. Nair and B. Venkatesh, "Adaptive filtering for signal processing using artificial bee colony algorithm," Signal Processing, vol. 123, pp. 34-48, 2016.
[87] H. Cui and G. Zhang, "Network routing optimization in telecommunications using artificial bee colony algorithm," Telecommunication Systems, vol. 55, pp. 243-258, 2014.
[88] S. O. Mahmoud and A. A. Habib, "Approximation of Nash equilibria in game theory using artificial bee colony algorithm," Journal of Game Theory, vol. 45, no. 2, pp. 150-165, 2017.
[89] J. V. Singh and P. S. Sharma, "Flood prediction in hydrological systems using artificial bee colony algorithm," Journal of Hydrology, vol. 497, pp. 300-314, 2013.
[90] Z. Wang and F. Shen, "Energy-efficient IoT device management using artificial bee colony algorithm," Journal of Internet of Things, vol. 28, no. 1, pp. 214-228, 2020.
[91] M. P. Guo and Y. H. Liu, "Precision agriculture optimization using artificial bee colony algorithm," Journal of Agricultural Informatics, vol. 6, no. 2, pp. 58-66, 2015.
[92] H. Zhao, W. Wu, and X. Yang, "Wearable health monitoring using artificial bee colony optimization for sensor network configuration," IEEE Sensors Journal, vol. 15, no. 8, pp. 4207-4215, 2015.
[93] A. Hussain and M. Alvi, "Optimization of electric vehicle charging station locations using artificial bee colony algorithm," Journal of Energy Systems, vol. 42, pp. 380-393, 2019.
[94] X. Liu, J. Zhang, and W. Zhao, "Artificial bee colony algorithm for e-commerce recommender systems," Journal of Electronic Commerce Research, vol. 18, no. 2, pp. 112-125, 2017. https://doi.org/10.1186/s13638-016-0802-2.
[95] R. S. Kumar and R. Rajan, "Flight scheduling optimization using artificial bee colony algorithm in aviation systems," Journal of Transportation Engineering, vol. 143, no. 4, 2017.
[96] F. M. Liu and G. K. Zhang, "Disaster management and emergency response optimization using artificial bee colony algorithm," International Journal of Disaster Risk Science, vol. 9, pp. 215-225, 2018.
[97] S. M. Almufti, R. B. Marqas, R. R. Asaad, and A. A. Shaban, "Cuckoo search algorithm: overview, modifications, and applications," 2025. [Online]. Available: www.sciencepubco.com/index.php/IJSW. https://doi.org/10.14419/efkvvd44.
[98] S. M. Almufti and A. A. Shaban, "U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem," Academic Journal of Nawroz University, vol. 7, no. 4, p. 45, Dec. 2018, https://doi.org/10.25007/ajnu.v7n4a270.
[99] A. Ahmed Shaban and I. Mahmood Ibrahim, "Swarm intelligence algorithms: a survey of modifications and applications," 2025. [Online]. Available: www.sciencepubco.com/index.php/IJSW. https://doi.org/10.14419/vhckcq86.
[100] S. M. Almufti and A. A. Shaban, "U-Turning Ant Colony Algorithm for Solving Symmetric Traveling Salesman Problem," Academic Journal of Nawroz University, vol. 7, no. 4, p. 45, Dec. 2018, https://doi.org/10.25007/ajnu.v7n4a270.
[101] S. M. Almufti, A. Ahmad Shaban, R. Ismael Ali, and J. A. Dela Fuente, "Overview of Metaheuristic Algorithms," Polaris Global Journal of Scholarly Research and Trends, vol. 2, no. 2, pp. 10–32, Apr. 2023, https://doi.org/10.58429/pgjsrt.v2n2a144.