ARTIFICIAL INTELLIGENCE OPTIMIZATION TECHNIQUES, ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND ITS GLOBAL REFLECTIONS

Prof. Dr. Derviş Karaboğa
Erciyes University

Prof. Dr. Bahriye Akay
Erciyes University

Attribution: Karaboğa, D., & Akay, B. (2026). Artificial intelligence optimization techniques, Artificial Bee Colony (ABC) algorithm and its global reflections. In H. Arslan, A. Taşdelen, E. Kuzucu Hıdır, & O. A. Çetin (Eds.), *Artificial intelligence and national technology reflections* (pp. 599–619). Turkish Academy of Sciences. https://doi.org/10.53478/TUBA.978-625-6110-86-1.ch25

10.53478/TUBA.978-625-6110-86-1.ch25

ARTIFICIAL INTELLIGENCE OPTIMIZATION TECHNIQUES, ARTIFICIAL BEE COLONY (ABC) ALGORITHM AND ITS GLOBAL REFLECTIONS

Prof. Dr. Derviş Karaboğa
Erciyes University

Prof. Dr. Bahriye Akay
Erciyes University

Abstract

Optimization deals with finding the best solution within complex, high-dimensional, and often constrained search spaces. Optimization methods can be solved using traditional analytical methods which employ mathematical procedures to guarantee optimality but are usually limited in applicability to non-linear, non-differentiable or non-convex problems. To avoid these limitations, nature-inspired heuristic methods are utilized to generate acceptable solutions. Among nature-inspired meta-heuristics, Swarm Intelligence (SI)-based methods stand out for their scalability, flexibility, and applicability due to their decentralized and self-organizing abilities. Multiple interactions, positive and negative feedback mechanisms, stochastic fluctuations in SI-based systems lead to collective intelligence to effectively search discrete, high-dimensional and complex solution space. One of the prominent examples of SI-based algorithms, the Artificial Bee Colony (ABC) algorithm mimics the foraging behavior of honeybee colonies. In ABC, there are three distinct types of agents: employed bees which exploit discovered high-quality solutions and share their information by dancing to recruit other bees to promising regions; onlooker bees search the vicinity of solutions selected probabilistically based on the information transmitted through dancing; and scout bees which explore new food source regions to bring diversity to the population. By iteratively balancing exploration and exploitation through self-organization mechanisms, ABC has been applied to a wide range of complex problems successfully in engineering, computer science, and computational biology.

Keywords

Optimization Problems, Population-based Meta-Heuristics, Swarm Intelligence, Artificial Bee Colony (ABC) Algorithm

Introduction

Optimization is one of the fundamental tasks in modern science and engineering, including formulation and finding the best possible solution of decision variables under specified constraints. These problems arise in diverse domains, including industrial process design, control systems, logistics, finance, computational biology, and artificial intelligence, where finding optimal or near-optimal solutions can significantly enhance efficiency, performance, and resource utilization. Formally, an optimization problem involves minimizing or maximizing one or more objective functions subject to equality and inequality constraints and bound limitations on decision variables. The structural characteristics of these problems such as linearity and convexity have an impact on the complexity of the search space and drive the choice of solution methodologies.

Traditional optimization techniques utilize mathematical procedures to find exact optimal solutions. However, they often face difficulties in high-dimensional, non-linear, or non-convex landscapes and their performances depend on type of problems. To overcome these limitations of traditional methods, metaheuristic optimization algorithms have been proposed to produce high-quality solutions across a wide spectrum of problem types. These modern approaches can be categorized into deterministic heuristics and stochastic heuristics. Stochastic heuristics can be further classified as single-solution-based methods and population-based methods evolving a set of candidate solutions simultaneously.

Among population-based meta-heuristic approaches, Swarm Intelligence (SI) (Beni & Wang, 1989) emphasizes the collective behavior of social organisms without any central supervision. Division of labor and self-organization are key principles in SI and local interactions among self-organizing agents in SI lead to global-level intelligence. Self-organization has four mechanisms which make the population adapt to challenging conditions: positive and negative feedback, fluctuations, and multiple interactions. Within the community of SI frameworks, the Artificial Bee Colony (ABC) algorithm (Karaboga, 2005) is particularly notable for its biological fidelity and computational efficiency. ABC models honeybee intelligent foraging behavior through three agent types: employed, onlooker, and scout bees. The employed bees search in the vicinity of discovered areas, onlooker bees probabilistically reinforce the selection of superior solutions, and scout bees bring diversity to population by exploring new promising regions. Due to the balance between exploitation and exploration mechanisms, ABC performs efficient search in complex landscapes while avoiding premature convergence or getting trapped in a local minimum. The ABC algorithm has demonstrated efficient performance in a wide range of applications, and its simplicity and robust search capabilities underscore it as a general-purpose optimization tool.

This chapter is organized to provide an explanation about optimization problems, solution methods, and the ABC algorithm. The fundamental concepts of optimization and a brief discussion is given to highlight optimization methods' strengths, limitations. Following the principles of SI, the ABC algorithm's search phases are explained. Finally, practical applications of ABC are presented to demonstrate its effectiveness across various fields. This chapter aims to provide readers with both theoretical foundations and practical insights of the ABC algorithm.

1. Optimization Problems and Methods

Complex optimization problems are encountered in various fields of science including designing more efficient systems, training deep learning models. To appreciate the foundation of algorithms, it is important to understand the essentials of optimization which deals with finding the best possible solution within a defined search space, guided by a problem-specific objective function. In its most general form, an optimization problem can be expressed by Eq. 1.

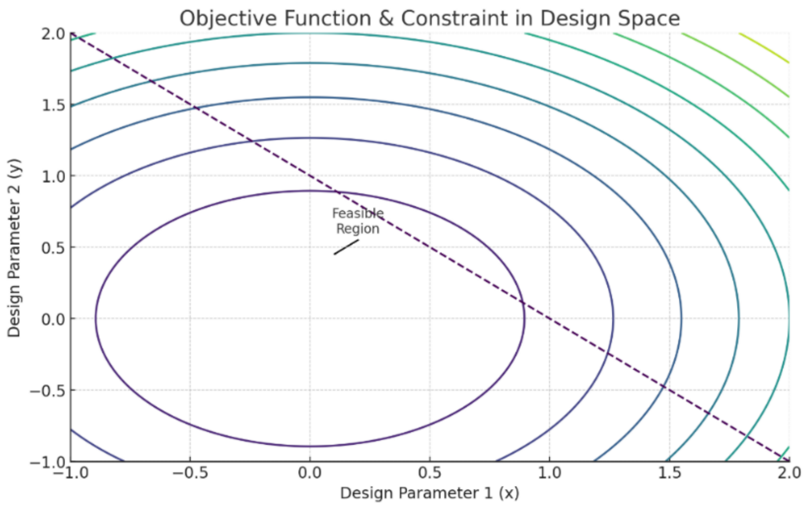
$$f(x)\ st.\ g_i(x) \leq 0, i = 1,2,\dots,m, h_j(x) = 0, j = 1,2,\dots,p, x_k^L \leq x_k \leq x_k^U, k = 1,2,\dots,D \tag{1}$$

where $f(x)$ is objective (target) function to be minimized, $x = [x_1, x_2, \dots, x_D]$ is decision vector of design variables (parameters), $g_i(x) \leq 0$ and $h_j(x) = 0$ are inequality and equality constraints, x_k^L, x_k^U are lower and upper bounds for each design variable (Figure 1).

Different types of optimization problems have been formulated in various fields and we should use appropriate methods according to their characteristics and complexity.

Figure 1

An Example for an Objective Function Constrained by Constraint Functions Defined by Decision Variables



1.1. Classification of Optimization Problems

Optimization problems can be divided into categories based on their structural characteristics of the cost function, the presence or absence of constraint functions, the topological properties of the search landscape. Such classification criteria provide a rigorous basis for understanding problem complexity and guide the selection or development of appropriate optimization methodologies.

Depending on Variables

- **Numerical vs. Combinatorial:**
  - *Numerical problems* involve finding the best values for a set of continuous or discrete variables.
  - *Combinatorial problems* focus on finding the optimal arrangement or combination of a set of objects.
- **Single-variable vs. Multi-variable:**
  - *Single-variable problems* optimize a function based on a single input variable.
  - *Multi-variable problems* involve optimizing a function with multiple interdependent variables.

Depending on Constraints

- **Unconstrained vs. Constrained:**
  - *Unconstrained problems* deal with finding the minimum or maximum of an objective function without any restrictions on the decision variables.
  - *Constrained problems* involve optimizing an objective function subject to equality and/or inequality constraints that limit the possible solutions that form the feasible region.

Depending on Objective Function

- **Linear vs. Non-linear:**
  - *Linear problems* are characterized by linear objective functions and constraints.
  - *Non-linear problems*, which are far more common in the real world, involve non-linear relationships.
- **Dynamic vs. Static:**
  - *Static problems* have a fixed objective function and search space over time.

- *Dynamic problems* involve an environment where the optimal solution may change over time.

- **Convex vs. Non-convex:**
  - *Convex problems* have a convex objective function, and the feasible region is defined by a convex set. Therefore, every local minimum is also a global minimum (unimodal).
  - *Non-convex problems* have either non-convex objective function or the feasible region (or both) resulting in the problem having multiple local minima and maxima, and finding the global optimum becomes challenging (multi-modal).

- **Single-objective vs. Multi-objective:**
  - *Single-objective problems* involve one objective function to be minimized or maximized.
  - *Multi-objective problems* involve two or more conflicting objective functions to be minimized or maximized simultaneously. Instead of a single "best" solution, it is typically associated with a Pareto front.

- **Separable vs. Non-Separable:**
  - *Separable problems* can be expressed as the sum of independent functions, each depending on only one variable or a disjoint subset of variables, each can be optimized independently.
  - *Non-Separable problems* cannot be broken down into a sum of individual functions because the variables are interdependent, which makes them more complex and computationally harder to optimize.

1.2. Classification of Optimization Methods

The optimization techniques for solving these problems can be broadly categorized into traditional methods and modern heuristic paradigms (Table 1). Traditional methods encompass analytical techniques, such as linear programming (Dantzig, 1951) and derivative-based algorithms, which exploit explicit mathematical properties of the objective function to guarantee optimality under well-defined conditions. Structural optimization methods, including dynamic programming (Bellman, 1954) and branch-and-bound (Land & Doig, 1960) procedures, systematically decompose the whole search space into sub-spaces to converge to globally optimal solutions. However, they need high computational resources especially for large-scale problems.

Table 1

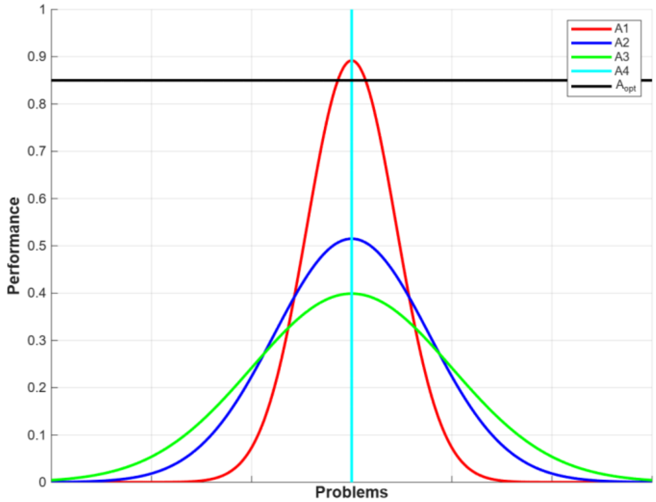
Classification of Optimization Methods

Traditional Methods	**Analytical Methods** (e.g., Linear Programming (Dantzig, 1951), Derivative methods) **Structural Methods** (e.g., Dynamic Programming (Bellman, 1954), Branch and Bound (Land & Doig, 1960))
Modern Heuristic Methods	**Deterministic Methods** (e.g., Tabu Search (Glover, 1989)) **Stochastic Methods** • *Single-solution-based* (e.g., Simulated Annealing (Kirkpatrick et al., 1983)) • *Population-based* (e.g., Evolutionary Algorithms (Fogel et al., 1966), Swarm Intelligence (Beni & Wang, 1989))

Although literature offers a wide range of traditional optimization algorithms whose performance may vary significantly across different classes of problems, the desirable scenario is the development of algorithms capable of consistently delivering acceptable and high-quality solutions over a broad spectrum of optimization tasks as shown in Figure 2. In Figure 2, A4 is a problem-tailored algorithm which produces the best result only for that problem. In other words, rather than designing methods tailored to specific problem types, it is desired to construct robust, general-purpose optimization frameworks that exhibit strong adaptability and reliable performance in diverse and potentially complex optimization landscapes. The limitations of traditional methods such as their reliance on derivative information or their tendency to become trapped in local optima created a clear need for more robust, explorative techniques.

Figure 2

Performance Spectrum of the Algorithms (A1, A2, A3, A4) on Various Optimization Tasks, While the Performance of the Desired Optimization Algorithm (A_{opt}) Does Not Deteriorate Significantly by the Problem Type



In contrast, modern heuristic optimization methods are designed to provide high-quality solutions within reasonable computational effort, particularly for complex, high-dimensional, or non-convex problem landscapes where traditional approaches become impractical. These methods can be systematically categorized into deterministic heuristics, such as tabu search (Glover, 1989) and stochastic heuristics. Stochastic heuristics can be further divided into single-solution-based approaches (e.g., simulated annealing (Kirkpatrick et al., 1983) and population-based approaches (e.g., evolutionary algorithms (Fogel et al., 1966) and swarm intelligence techniques (Beni & Wang, 1989) which evolve a set of candidate solutions instead of just one solution and perform more elaborate exploration in search space. Population-based methods can be examined through two main paradigms:

- **Evolutionary Algorithms (EAs):** By mimicking the natural evolution and survival of the fittest selection, these algorithms use mechanisms like selection, crossover (recombination) and mutation to guide the search toward an optimal state. Genetic Algorithm (GA) (Holland, 1975), Genetic Programming (GP) (Koza, 1992) and Differential Evolution (DE) (Storn & Price, 1997) fall into this group.
- **Swarm Intelligence (SI) Algorithms:** This paradigm is established on a cooperative model inspired by the collective behavior of social organisms. It yields a complex, global-level intelligence based on local social interaction and information sharing among agents, as seen in ant colonies, bird flocks, and bee colonies.

2. Swarm Intelligence (SI)

The fact that the researchers have been inspired by nature to design efficient algorithms led to rise of nature-inspired heuristics, particularly Swarm Intelligence (Beni & Wang, 1989) which models the collective problem-solving abilities of locally interacting agents (Bonabeau et al., 1999). This collective intelligence emerges from their local interactions with each other and their environment. Collective intelligence in a swarm exhibits six main characteristics (Beni & Wang, 1989):

- Flexibility: The system can adapt and produce necessary responses to internal and external threats or changes.
- Robustness: The group can successfully complete its tasks even if some individuals fail, ensuring the system is resilient to partial failures.
- Scalability: The system functions effectively whether it consists of a few individuals or millions, allowing it to adapt to problems of varying scales.
- Decentralized Control: There is no central supervision; control is distributed throughout the swarm.
- Self-organization: The system can adapt its structure and behavior without external guidance, creating order from local interactions.
- Parallelism: Individuals perform their tasks simultaneously, allowing for efficient, parallel processing of information and actions.

Self-organization is enabled by four core mechanisms:

- Positive Feedback encourages beneficial action patterns.
- Negative Feedback abandons inefficient action patterns.
- Fluctuations (Randomness) provide diversity to discover new patterns.
- Multiple Interactions enable information sharing in a collective manner.

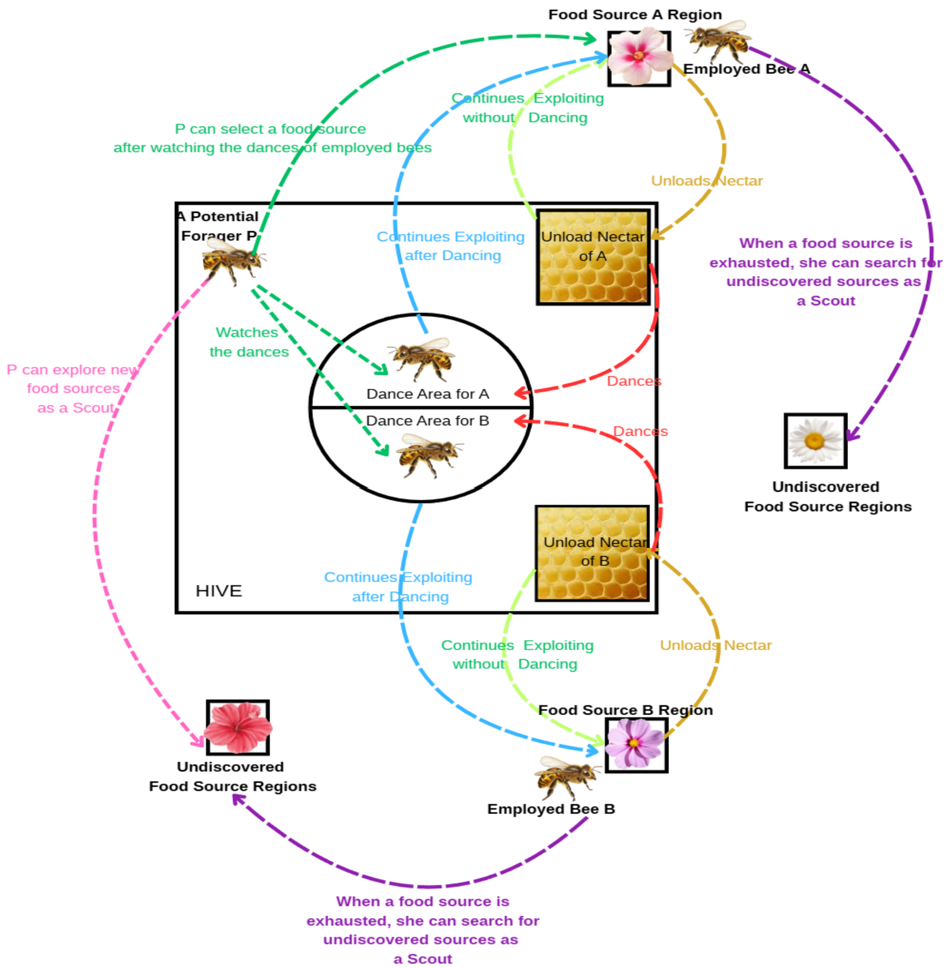
3. Swarm Intelligence in Honeybee Foraging

The foraging behavior of honeybees is a prominent example of swarm intelligence balancing the exploitation of known sources and exploration of new rich ones without any central supervision. This section explains the intelligence in foraging behavior of honeybees, exhibiting division of labor, communication strategies, and decision-making. These patterns in foraging make it a powerful base model for a robust optimization algorithm.

A honeybee colony performs many tasks efficiently in an efficient manner as a highly organized superorganism such as nest site selection, mating, hive organization, brood tending, navigation, communication and foraging (Karaboga & Akay, 2009). Considering the survival of a bee colony, the colony's fundamental problem is to maximize the nectar loaded to hive with the least energy consumption. This essential task is achieved by a high-level division of labor and self-organization based on intrinsic decision-making without any central supervision which are required to exhibit swarm intelligence ability. The honeybees assigned to foraging tasks are divided into three groups, which reflects the division of labor in foraging and they may adaptively change their role depending on the hive and environment conditions as shown in Figure 3.

Figure 3

Division of Labor in Honeybee Foraging



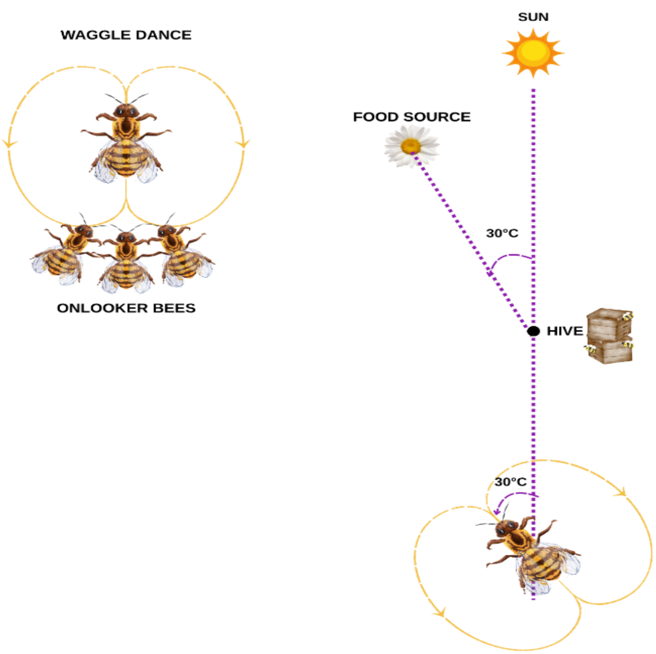
- Employed Bees: These bees exploit nectar of discovered food sources in their memory. They dance in the hive to transmit information about their sources including direction, distance, and nectar quality and to recruit onlooker bees towards potentially rich sources.
- Unemployed Bees: This group is not currently assigned to a particular food source and consists of two subtypes:

- Onlookers: These bees observe the dances of employed bees in the hive and use the transmitted information to fly a high-quality food source for exploitation.
- Scouts: They search the environment to discover new, potentially richer food sources.

Bees' communication system based on dancing is the main behavior pattern that leads to the emergence of collective intelligence. Employed bees associated with profitable food sources perform a dance to share their location with the onlooker bees observing the dances which encode direction and distance to the source. For sources more than 100 meters away, they use the waggle dance (Figure 4) while the round dance is performed for closer sources. The duration of the dance gives the distance of the food source while the angle of the dance is the angle between the food source and the sun.

Figure 4

Waggle Dance of Honeybees



This dance-based interaction enables a positive feedback mechanism, allowing the colony to recruit more foragers to the most promising locations. When the behavior of honeybees is examined, it is seen that four core mechanisms of self-organization are inherent in foraging honeybees (Seeley, 1995):

- Positive Feedback (Recruitment): When an employed bee finds a high-quality food source, its recruitment dance attracts more onlooker bees to that source.

- Negative Feedback (Abandonment): If a food source becomes depleted or is of poor quality, bees will eventually abandon it to be recruited to better sources.
- Fluctuations (Discovery): Scout bees are responsible for discovering entirely new rich food sources, ensuring the colony does not get stuck exploiting only known options.
- Multiple Interactions (Information Sharing): Employed bees share information about food source quality and location to onlooker bees to make collective decisions.

4. The Artificial Bee Colony (ABC) Algorithm

Artificial Bee Colony (ABC) algorithm (Karaboga, 2005) which is a prominent and powerful swarm intelligence algorithm, was introduced in 2005 at Erciyes University, Turkiye. This section will explore the phases of the algorithm and application of the ABC algorithm to demonstrate how the foraging behavior of honeybees can be simulated for solving optimization problems. Here, we define the core components of the ABC algorithm and associate its components with fundamental optimization concepts. The algorithm effectively and closely mimics the bees' foraging behavior to guide the search in the problem landscape (Karaboga & Basturk, 2007). The underlying metaphor of the ABC algorithm is the direct analogy between honeybees' foraging for food sources with high nectar amounts and an algorithm's search for an optimal solution. This mapping is crucial for clearly understanding the function of each component. The analogy between foraging in real honeybees’ colony and artificial bee colony algorithm is presented in Table 2. The ABC algorithm follows a simple, iterative structure that mimics the continuous foraging cycle of a honeybee colony. The process is repeated until a predefined stopping criterion (e.g., a maximum number of cycles) is met.

1. Initialization
2. REPEAT
3. Employed bees’ phase.
4. Onlooker bees’ phase.
5. Scout bees’ phase.
6. UNTIL (stopping criterion is met).

Table 2

Analogy Between Foraging in Real Honeybees' Colony and Artificial Bee Colony Algorithm

Optimization in foraging of real honeybees' colony	**Artificial bee colony (ABC) Optimization**
Working in continuous time	Discrete time
Having memory	Memory
Maximizing the honey inside a hive by exploring and exploiting rich food sources	**Max/Minimizing** objective function(s) for a problem
Accessible **food sources** around a hive	**Possible solutions** for an optimization problem
Number of accessible food sources around a hive	**Population size** of possible solutions in a search space
Nectar amount (profitability) of a food source	**Fitness** (quality) of a solution to the optimization problem
Five-dimensional search space	**n-dimensional** search space
Search Modes: Exploiting food sources Exploring new rich food sources	**Two search modes:** Local searches Global searches
Employed bees	**Local-global search** ability using her memory
Onlooker bees (positive feedback in SI)	**Local search** ability in the vicinity of possible good solutions
Scout bees (negative feedback and fluctuations in SI)	**Global search** ability by the abandonment mechanism and random search
Multiple interactions in SI i) Dance based ii) Pheromone based	**Dance based interaction**
Adaptation to new conditions in the hive and changes in the environment	**Balance** between exploration and exploitation based on "limit" parameter

- Initialization: In the ABC algorithm, each food source is represented by a vector of design parameters. In the initialization phase of the algorithm, the control parameter *limit* is assigned to identify the depleted sources in the scout bee phase, size of population (*SN*) is predefined, and a food source population of *SN* solutions is generated by Eq. 2.

$$x_{ij} = x_i^L + rand(0,1)(x_j^U - x_j^L) \quad (2)$$

where $i = 1, \ldots, SN$ and $j = 1, \ldots, D$, D is dimension of the search space, x_j^L and x_j^U are lower and upper bounds of jth parameter, respectively.

- Employed Bees' Phase (Exploitation and Exploration): In this phase, each employed bee, assigned to a food source (a solution), explores its immediate vicinity for a better option. It achieves this by generating a new candidate solution (v_i) in the neighborhood of its current one (x_i), modifying a single dimension of the solution vector based on information from another randomly selected solution by Eq. 3. If a better solution is found, the bee memorizes it and forgets the old one. Otherwise, the number of unsuccessful solution updates associated with the solution is incremented by 1.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (3)$$

where (x_k) is a randomly selected neighbor solution ($i \neq k$), j is a random dimension to be perturbated, φ_{ij} is a random number within the range [-1,1]. This phase represents the *exploitation* of known, promising areas of the search space.

- Onlooker Bees' Phase (Exploitation Reinforcement): After the employed bees have gathered information, they share it inside the hive. Onlooker bees evaluate the profitability (fitness) of all known food sources (p_i) by Eq. 4 and probabilistically choose one to exploit further. Higher-quality solutions have a higher probability of attracting onlookers, thus reinforcing the search effort in the most promising regions of the search space.

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (4)$$

- Scout Bees' Phase (Exploration): This phase is the algorithm's primary mechanism for avoiding local optima and enabling *exploration*. If the number of unsuccessful solution updates of a food source (solution) exceeds a predetermined number of attempts (a parameter known as the "limit"), it is considered exhausted and is abandoned. The employed bee associated with that source becomes a scout to discover new solution randomly generated in the search space by Eq. 1. This prevents the algorithm to get trapped in a local minima and ensures new regions are continuously explored.

4.1 Applications and Evolution of ABC

The search power of ABC algorithm which balances exploration and exploitation leads to practical results in both academic researches and real-world applications (Karaboga et al., 2014) (Akay et al., 2021) (Kaya et al., 2022) (Ibrahim et al., 2025) (Gorkemli et al., 2025). This section will evaluate the impact of the ABC algorithm by examining its competitive situation in the scientific community and its current evolution.

The ABC algorithm's simplicity and adaptability have led to its successful application across a vast range of domains. Its utility has been demonstrated in fields as diverse as engineering, computer science, and computational biology. Some examples are:

- Engineering: Electrical, Electronics, Industrial, Mechanical, Civil, Control, and Software Engineering.
- Computer Science: Wireless sensor networks, signal and image processing, data mining, clustering, classification, and feature extraction, cyber-security and training of artificial and deep neural networks.
- Computational Biology & Finance: Optimizing protein structures, drug design, financial forecasting models.

Variants of ABC: The ABC algorithm is a dynamic framework evolving continuously. Researchers have developed some versions to tackle different classes of problems including combinatorial, binary, parallel, and hybrid variants. Furthermore, in order to take advantage of their complementary strengths, ABC has been hybridized with other optimization methods including combinations with Tabu Search (Glover, 1989), Genetic Algorithms (GA) (Holland, 1975), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), and Differential Evolution (DE) (Storn & Price, 1997).

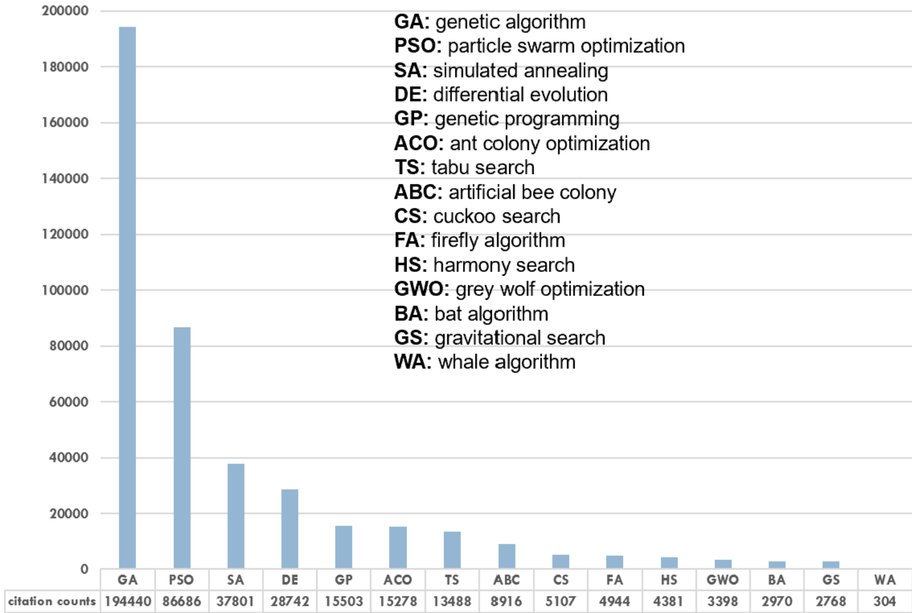
This proven performance, coupled with its remarkable flexibility, has firmly cemented the Artificial Bee Colony algorithm's place as a valuable and powerful tool for modern optimization.

4.2. Analyzing the ABC Algorithm's Academic Impact and Global Reflections

In the literature, researchers introduced so many algorithms inspired by nature and metaphors in nature (Campelo & Aranha, 2025).The significance of an algorithm within the global research community is often measured by its citation count, which reflects its adoption and influence and its scientific background. The citation metrics shown in Figure 5 from two major academic databases, Google Scholar and Web of Science, demonstrate that ABC's impact is highly competitive with foundational algorithms like SA (Kirkpatrick et al., 1983), TS (Glover, 1989), GA (Holland, 1975), GP (Koza, 1992), PSO (Kennedy & Eberhart, 1995), Ant Colony Optimization (Dorigo et al., 1996), DE (Storn & Price, 1997). This high level of academic interest is a strong indicator of its perceived effectiveness, robustness, and widespread adoption for solving complex optimization problems.

Figure 5

The Citation Counts of the Most-Used Meta-Heuristic Algorithms (Access Date: 01/06/2025)



Conclusion

The Artificial Bee Colony algorithm is a powerful example of approaches modelling the collective intelligence of natural systems to obtain robust solutions to complex computational problems. Inspired by the efficient foraging behavior of honeybees, it provides an effective framework balancing the exploration and exploitation processes in an optimization search. The widespread success and adoption of the ABC algorithm can be attributed to a combination of several key advantages:

1. Simplicity and Generality: It is derived from an efficient and powerful model of collective foraging, making it relatively easy to understand and implement for a wide variety of problems.

2. Few Control Parameters: Compared to many other heuristic algorithms, ABC requires the tuning of very few control parameters (e.g., colony size, limit), simplifying its application.

3. Robustness: The scout bee mechanism provides the ability to escape local minima and search the whole search space efficiently.

4. Flexibility and Adaptability: Its modular structure allows it to be easily modified for different types of optimization problems and hybridized with other algorithms to enhance its performance.

Ultimately, the Artificial Bee Colony algorithm is more than just an effective optimization tool; it is evidence of the power of biomimicry. Studying collective intelligence in natural systems from the complex dances of honeybees to the coordinated movements of bird flocks continues to inspire the development of innovative solutions for the most challenging computational problems.

References

Akay, B., Karaboga, D., Gorkemli, B., & Kaya, E. (2021). A survey on the artificial bee colony algorithm variants for binary, integer and mixed integer programming problems. *Applied Soft Computing, 106*, 107351.

Bellman, R. E. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society, 60*, 503-515.

Beni, G., & Wang, J. (1989). Swarm intelligence in cellular robotic systems. Tuscany, Italy.: Proceedings of the NATO Advanced Workshop on Robots and Biological Systems.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems.* Oxford, UK: Oxford University Press.

Dantzig, G. B. (1951). Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans (Ed.). *Activity analysis of production and allocation, John Wiley & Sons*, pp. 339–347.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26*(1), 29-41.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution.* New York: John Wiley & Sons.

Glover, F. (1989). Tabu Search–Part I. *ORSA Journal on Computing, 1*(3), 190-206.

Gorkemli, B., Kaya, E., Karaboga, D., & & Akay, B. (2025). A review on the versions of artificial bee colony algorithm for scheduling problems. *Journal of Combinatorial Optimization, 49*, 62.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence..* Ann Arbor, MI: University of Michigan Press.

Ibrahim, A., Elfadel, E., & Hashem, I. (2025). The Artificial Bee Colony Algorithm: A Comprehensive Survey of Variants, Modifications, Applications, Developments, and Opportunities. *Arch Computat Methods Eng, 32*, 3499–3533.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization.* Erciyes University.

Karaboga, D., & Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. 31, 61 (2009). *Artificial Intelligence Review, 31*, 61-85.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471.

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review, 42*, 21–57 .

Kaya, E., Gorkemli, B., Akay, B., & & Karaboga, D. (2022). A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. *Engineering Applications of Artificial Intelligence, 115*, 105311.

Kennedy, J., & Eberhart, R. (1995). A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MMHS'95).* 39-43.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671-680.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection.* Cambridge, MA: MIT Press.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica, 28*(3), 497–520.

Seeley, T. D. (1995). *The Wisdom of The Hive.* Harvard University Press.

Storn, R., & Price, K. (1997). Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341-359.

About the Authors

Prof. Dr. Dervis KARABOĞA
Erciyes University | karaboga[at]erciyes.edu.tr
ORCID: 0000-0003-1439-6969

Dervis KARABOGA received BSc degree from Electrical and Electronics Engineering Department of Erciyes University, Türkiye, in 1983; MSc degree from Electronics and Communication Engineering Department of Technical University of Istanbul, Türkiye, in 1988 and PhD degree from Systems Engineering Department of University of Wales, College of Cardiff, UK, in 1994. He currently works in Computer Engineering Department of Erciyes University, Türkiye. His general scientific study field is artificial intelligence. Karaboga particularly studies fuzzy systems, artificial neural networks, evolutionary computation and swarm intelligence-based optimization algorithms and their applications. He invented the artificial bee colony (ABC) optimization algorithm by modelling the intelligent foraging behavior of honeybee swarms in 2005 and has written two books on intelligent optimization techniques as well as a large number of scientific articles and papers. Due to his scientific studies, he achieved to be in the list of "highly cited researchers" in the years of 2016-2019. In 2023, he was awarded the TÜBİTAK Science Award and the TÜBA International Science Award.

Prof. Dr. Bahriye AKAY
Erciyes University | bahriye[at]erciyes.edu.tr
ORCID: 0000-0001-6575-4725

Bahriye AKAY received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from Erciyes University, Türkiye, in 2003, 2005, and 2009, respectively, and the Postdoctoral degree from the University of Birmingham, U.K., in 2011. She is currently a Professor with the Department of Computer Engineering, Erciyes University. She was selected as a Highly Cited Researcher by Clarivate Analytics, in 2018. Her research interests include artificial intelligence, evolutionary computation, swarm intelligence techniques, and software engineering. In 2022, she was awarded TÜBA Outstanding Young Scientist Award (GEBIP).