Energy-Efficient Optimization in Wireless Sensor Networks Using Artificial Bee Colony Algorithms

Tarun K Y[1], Mrs. Shobha Chandra K[2], Sushmitha L K[3], Varshini M[4], and Suchitha K K[5]
[1,3,4,5] *Student, Malnad College of Engineering, Hassan*
[2]*Associate Prof, Malnad College of Engineering, Hassan*

***Abstract*—Wireless Sensor Networks (WSNs) face obstacles in energy efficiency, coverage optimization, and data collection reliability due to resource limitations. This study deploys Artificial Bee Colony (ABC) algorithms to improve network performance. The proposed approaches include clustering models for balanced energy consumption, routing algorithms for optimized data transmission, and scalable approaches to improve network coverage. By addressing difficulties such as cluster head selection, wireless node path optimization, and data latency, the algorithms significantly extend network lifetime, enhance data collection efficiency, and achieve superior coverage rates. Simulation results demonstrate the performance of ABC-based methods compared to traditional optimization algorithms**

***Index Terms*—Artificial Bee Colony, ABC, Optimization**

I. INTRODUCTION

Optimization problems are intrinsic to numerous fields, encompassing from engineering and computer science to operations research and logistics [16]. These problems, often characterized by high complexity, nonlinear interactions, and conflicting objectives, require innovative and efficient solutions. Among the multitude of approaches developed, nature-inspired algorithms have emerged as particularly effective tools. One such type of algorithm is, Artificial Bee Colony (ABC) algorithm, stands out for its comprehensibility, adaptability, and efficiency in solving a broad spectrum of optimization problems [7]. Based on the shared foraging patterns observed in honeybee colonies, the behavior is deeply merge in their collective dynamics, the ABC algorithm embodies the principles of self-organization, adaptability, and collaboration that define these natural systems [2].

The ABC algorithm was initially introduced by Karaboga in 2005 as a novel metaheuristic optimization technique [12]. The method imitates the intelligent foraging behavior of honeybee colonies, where individuals work collectively to discover and extract optimal food sources while maintaining a responsive search process [10]. In computational terms, the algorithm maps the biological behaviors of bees—such as food source discovery, resource sharing, and exploration—onto key stages of optimization. These resemblances allow the ABC algorithm to simulate a equitable balance between local exploration (intensification) and global search (diversification), making it particularly suitable for solving nonlinear, multimodal, and complex problems [13].

What makes the ABC algorithm unique is its elegant yet robust framework. Unlike many other metaheuristic algorithms that involve detailed mechanisms and a multitude of parameters, the ABC algorithm employs a minimalistic approach [13]. This comprehensibility not only makes it easy to implement but also enhances its flexibility and scalability across diverse application domains. In particular, the algorithm has demonstrated exceptional performance in areas such as engineering design, wireless sensor networks (WSNs), image processing, data mining, and scheduling. The adaptability of the ABC algorithm also lends itself to hybridization and integration with other optimization techniques, further broadening its scope and potential [10].

At its core, the ABC algorithm revolves around three primary roles within a virtual colony: employed bees, onlooker bees, and scout bees [7]. These roles correspond to distinct phases in the optimization process. Employed bees focus on leveraging known solutions and refining them, while onlooker bees evaluate and select promising solutions based on shared information. Meanwhile, scout bees explore new areas of the unexplored space to introduce diversity and stop the algorithm from stagnating in local optima [9]. This division of labor impersonates the natural dynamics of honeybee colonies, where effective communication and responsive role distribution are critical for survival and efficiency.

In optimization, the ”search space” constitutes the domain of all possible solutions to a given problem. Food origins correspond to these results, and their ”nectar amount” reflects their fitness or quality [9]. The algorithm begins by initializing a population of random solutions, after which it iteratively refines these solutions through cycles of exploration and exploitation [13]. By leveraging probabilistic selection mechanisms, neighborhood searches, and random exploration, the ABC algorithm effectively balances the trade-off between converging on promising regions and exploring uncharted areas.

One of the remarkable applications of the ABC algorithm lies in the realm of wireless sensor networks. These networks, composed of physically distributed sensor nodes, are vital in modern technologies such as environmental monitoring, smart cities, and industrial automation [6]. However, WSNs face significant challenges, including limited energy resources, communication constraints, and the need for efficient routing. The ABC algorithm addresses these challenges by enhancing key parameters such as cluster head selection, routing paths, and energy distribution. By doing so, it enhances network lifespan, reliability, and overall performance, demonstrating its potential in tackling real-world engineering problems [4].

Beyond WSNs, the ABC algorithm has been effortlessly applied to other domains, including engineering design, where it aids in optimizing structural configurations and component layouts. In image processing, the algorithm has been utilized for tasks such as segmentation, feature selection, and enhancement [5]. Its ability to navigate complex, high-dimensional search spaces makes it a precious tool in data mining, where uncovering meaningful patterns and insights from large datasets is paramount. Additionally, the ABC algorithm has proven effective in resource allocation, scheduling, and other combinatorial optimization problems, underscoring its versatility [13].

Despite its strengths, the ABC algorithm is not without limitations. In certain scenarios, particularly those involving highly rugged or deceptive search spaces, the algorithm may converge prematurely to suboptimal solutions. To mitigate this, researchers have proposed various enhancements, including adaptive parameter tuning, hybridization with other algorithms, and the incorporation of multi-strategy approaches. These improvements aim to enhance the algorithm’s convergence speed, robustness, and ability to escape local optima, further reinforcing its position as a leading optimization tool [7].

The growing popularity of the ABC algorithm can also be attributed to its alignment with current trends in computational intelligence. As fields such as artificial intelligence, big data analytics, and the Internet of Things continue to develop, the demand for efficient and adaptable optimization techniques is increasing. The ABC algorithm’s capability to adjust to different problem domains, coupled with its computational efficiency, positions it as a valuable asset in addressing these emerging challenges. Moreover, its simplicity and transparency make it accessible to researchers and practitioners across disciplines, fostering widespread adoption and innovation [8].

The potential of the ABC algorithm extends beyond traditional optimization problems. With ongoing advancements in algorithmic design and computational resources, new opportunities are emerging to integrate the ABC algorithm with machine learning, neural networks, and other cutting-edge technologies. For example, the algorithm can be employed for feature selection, hyperparameter optimization, and model tuning in machine learning applications. Its ability to handle multi-objective optimization problems also opens doors to tackling complex challenges involving conflicting objectives, such as balancing cost, efficiency, and sustainability [10].

II. RELATED WORKS

Mobile Wireless Sensor Networks (MWSNs) have emerged as a promising solution to address the energy depletion and data transmission bottlenecks observed in traditional wireless sensor networks. In particular, research on data collection strategies in MWSNs has gained significant attention due to the need for energy-efficient routing and network reliability. Numerous approaches have been suggested to reduce the common issues associated with fixed sensor nodes, such as the "energy hole" problem, communication overhead, and constraints in data transmission [15].

Early research primarily focused on optimizing the routing protocols for fixed sensor networks. In these systems, sensor nodes are static, and data is collected through a fixed routing path that leads to a central Sink node. However, as nodes closer to the Sink face a higher load, their energy drains faster, leading to early node failures and a reduced overall network lifetime. To address this issue, several studies have turned to mobile Sink nodes, which can traverse the network and adjust their routes dynamically, offering more flexibility in terms of load balancing and energy distribution.

The movement of the Sink node in MWSNs is one key aspect that has been explored. In particular, approaches such as those proposed by Zhang et al, Li et al, and Yang et al [14] focus on mobile Sink strategies that either move along fixed paths or adjust based on real-time network conditions. These strategies aim to reduce energy consumption by avoiding the concentration of traffic around the Sink and extending the network’s operational lifetime. However, these algorithms often overlook the need to balance multiple objectives, such as maximizing data collection efficiency, minimizing the mobile Sink's travel distance, and ensuring network reliability [14].

In response to these obstructions, some researchers have introduced bio-inspired optimization techniques, such as the Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) algorithms. These approaches are designed to solve NP-hard problems by comprehensibility natural processes. The ABC algorithm, in specific, has acquired attention for its simplicity and efficiency in exploring large solution spaces. Studies similar to those by Karaboga et [11] and others have exhibited the effectiveness of ABC in various optimization problems, including MWSN path planning. However, the need for an algorithm that can jointly optimize cluster head selection, data transmission paths, and mobile Sink routing remains a gap in the current literature.

The proposed study seeks to fill this gap by combining these concepts into a unified heuristic approach. By leveraging the artificial bee colony algorithm, the paper optimizes not only the Sink’s path but also the selection of cluster head nodes and data transmission routes, which significantly improves energy efficiency, network reliability, and overall data collection performance [15].

III. WORKING OF ABC ALGORITHM

The ABC algorithm translates the foraging behavior of bees into a computational framework to solve optimization problems. The algorithm operates through iterative cycles comprising the following key phases [7]:

A. Initialization: A population of food sources (candidate solutions) is randomly generated within the search space. In optimization problems, the "search space" talks about the set of all possible solutions that the algorithm explores to find the most optimal one. Each food source points to a potential result to the optimization problem.

B. Employed Bee Phase: Employed bees look for better solutions in the neighborhood of their current food sources. They utilize a greedy selection mechanism to determine whether to retain the original source or replace it with a newly discovered solution.

C. Onlooker Bee Phase: Onlooker bees assess the quality of food sources based on a probability proportional to their fitness. Using shared information from employed bees, they focus their exploration on promising areas of the search space.

D. Scout Bee Phase: Scout bees abandon food sources that have not improved over a predefined number of iterations and randomly explore the search space to identify new solutions. This step prevents the algorithm from stagnating at local optima.

E. Evaluation and Update: The best solution found so far is noted, until a termination criterion is met the cycle is repeated, such as a maximum number of iterations or convergence to a satisfactory solution, is met.

IV. BIOLOGICAL INSIPIRATION

The behavior of honeybees in locating and harvesting nectar from flowers provides the foundational inspiration for the ABC algorithm. In computational terms, "food sources" represent potential solutions to optimization problems, while "nectar amount" corresponds to the fitness value of these solutions. The roles of the employed, onlooker, and scout bees are allocated to different phases of solution survey and refinement, capturing the dynamic and interactive aspects of the algorithm. In a natural hive, bees are classified into three groups primarily: employed bees, onlooker bees, and scout bees. Each type of bee does a specific role in the colony's exploring process. Employed bees exploit known food origins, onlooker bees calculate the quality of food origins based on shared data, and scout bees explore new areas to discover potential resources. The division of tasks and dynamic communication between these groups enable the colony to optimize its energy disbursement and maximize nectar collection [16].

V. ARTIFICIAL BEE COLONY ALGORITHM AND METHODOLOGY

In 2005, Zhang et al [14], inspired by the foraging patterns of bee colonies, introduced a novel heuristic approach known as the Artificial Bee Colony (ABC) algorithm. Within this algorithm, the "colony" is divided into three different phases: employed bees, onlooker bees, and scout bees. Each bee corresponds to a location in the solution space, and the algorithm uses the collective behavior of these bees to discover the optimal path. Employed bees try to explore formerly visited food sources, onlooker bees select food sources based on the "dance" performed by employed bees, and scout bees randomly look for new food source\cite{yue2016abc}s. The food source locations represent potential results to the optimization problem, while the "nectar" quantity embodies the fitness of each result. The algorithm allocates the first half of the bee population as employed bees, with the remaining half acting as onlooker bees.

The ABC algorithm can be divided into four steps primarily [1]:

A. Initialization: Assume the population size is SN, where N represents the first set of food sources in the initial population Xi = {Xi1, Xi2, . . . , XiD } (i = 1, 2, . . . , N) , and D is the dimension of the optimization problem. The random initial population is generated using:
Xj = Xmin + rand(0, 1) · (Xmax − Xmin) [15].

B. Population Update: The initial food source locations are randomly given to employed bees. During each iteration, employed bees explore neighboring food sources of their current locations using: V1j = X1 + rand(−1, 1) · (Xij − Xxj)
Here, k ∈ {1, 2, . . . , SN }, j ∈ {1, 2, . . . , D}.

If the nectar quality of the new food source exceeds the previous one, the employed bee moves to it; or else, it stays at the current source [15].

C. Food Source Selection: Employed bees evaluate food sources based on their nectar quality (fitness value). Higher fitness results are more presumably to be chosen for further exploration using:

$$P_i = \frac{\mathrm{fit}\left(X_i\right)}{\sum_{n=1}^{SN}\mathrm{fit}\left(X_n\right)},$$

where fit(Xn) represents the fitness of solution Xn, proportional to the nectar quality of food source n. The fitness value is calculated as:

$$\mathrm{fit}\left(X_n\right) = \begin{cases} \frac{1}{f\left(X_n\right)}, & f\left(X_n\right) \geq 0 \\ 1 + \mathrm{abs}\left(f\left(X_n\right)\right), & f\left(X_n\right) < 0, \end{cases}$$ [15].

D. Population Elimination: If a solution stagnates after repeated updates, it is considered to be stuck in a local optimum and discarded. Corresponding onlooker bees transform into scouts, generating a new solution to replace the discarded one using:

Xij = Xmin + rand(0, 1) · (Xmax − Xmin)

Here,

j ∈ {1, 2, . . . , D},

rand(0,1) is a random value between 0 and 1, and Xmax and Xmin represent the maximum and minimum solution bounds, respectively [3].

ALGORITHM-CLUSTER SELECTION PROCESS[15]

1. Initialize the routing table.
2. While TRUE:
a. Listen for incoming packets.
b. If Receive Broad_Msg{0, 0, 0}:
i. Mark the node as a cluster.
ii. Send Broad_Msg{1, SNA, 0} (SNA is the current node's network address).
c. Else if Receive Broad_Msg{1, srcNetwAddr, hop}:
i. Check the routing table for destination srcNetwAddr.
ii. If no entry exists:
A. Add a new entry to the routing table: {Destination = srcNetwAddr, Metric = hop + 1}.
B. Broadcast Broad_Msg{1, srcNetwAddr, hop + 1}.
iii. Else if Metric > hop + 1:
A. Update the routing table: Metric = hop + 1.
B. Broadcast Broad_Msg{1, srcNetwAddr, hop + 1}.

iv. Else:

A. Ignore the current broadcast message.

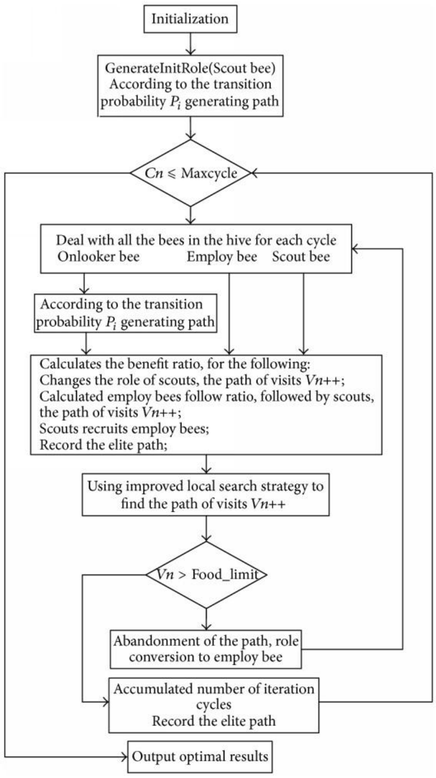

Fig. 1. Optimization- Based Artificial Bee Colony Algorithm for Data Collection Wireless Sensor Networks [15].

VI. REFERENCES

[1] W. Gao et al. "Improved Artificial Bee Colony algorithm based gravity matching navigation method". In: Sensors 14.7 (2014), pp. 12968–12989. ISSN: 14248220. DOI: 10 . 3390 / s140712968. URL: https : / / doi .org/10.3390/s140712968.

[2] "Improved FGMRP Routing Protocol to Solve Energy Hole Problem in Wireless Sensor Networks". In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). Date of Conference: 17-18 December 2021, Date Added to IEEE Xplore: 09 March 2022. IEEE, 2021. DOI: 10.1109/ICAC3N53548.2021.9725598.

[3] D. Karaboga, S. Okdem, and C. Ozturk. "Cluster based wireless sensor network routing using artificial bee colony algorithm". In: Wireless Networks 18.7 (2012), pp. 847–860. ISSN: 1572-8196. DOI: 10.1007/s11276-012-0438-z. URL: https://doi.org/10.1007/s11276-0120438-z.

[4] M. Krishnan, M. G. Yoon, and Y. Sangwoon. "An Improved Clustering with Particle Swarm Optimization-Based Mobile Sink for Wireless Sensor Networks". In: Tirunelveli: IEEE, 2018, pp. 1024–1028.

[5] P. Lalwani and S. Das. "Bacterial Foraging Optimization Algorithm for CH Selection and Routing in Wireless Sensor Networks". In: Dhanbad: IEEE, 2016, pp. 95–100.

[6] M.A. Latif and M.P. Saka. "Optimum design of tiedarch bridges under code requirements using enhanced artificial bee colony algorithm". In: Engineering Structures 199 (2019). Available online 6 June 2019, p. 109679. DOI: 10.1016/j.engstruct.2019.109679.

[7] Xianneng Li and Guangfei Yang. "Artificial bee colony algorithm with memory". In: Applied Soft Computing 41 (2016), pp. 362–372. DOI: 10.1016/j.asoc.2015.12.041.URL: https://doi.org/10.1016/j.asoc.2015.12.041.

[8] K. Navnath Dattatraya. "Hybrid-Based Cluster Head Selection for Maximising Network Lifetime and Energy Efficiency in WSN". In: Journal of King Saud University - Computer and Information Sciences (2019), pp. 1–11.

[9] "Performance Analysis of Wireless Sensor Networks Using Artificial Bee Colony Algorithm". In: 2018 Technologies for Smart-City Energy Security and Power (ICSESP). Date of Conference: 28-30 March 2018, Date Added to IEEE Xplore: 11 June 2018. IEEE, 2018, DOI: 10.1109/ICSESP.2018.8376711.

[10] Deepa S. R and Rekha D. "Cluster Optimization in Wireless Sensor Network Based on Optimized Artificial Bee Colony Algorithm". In: IET Networks (2021). First published: 21 March 2021, Citations: 4. DOI: 10.1049/ntw2.12023. URL: https://doi.org/10.1049/ntw2.12023.

[11] Weixing Su et al. "A novel comprehensive learning artificial bee colony optimizer for dynamic optimization biological problems". In: Saudi Journal of Biological Sciences 24.3 (2017), pp. 695–702. DOI: 10.1016/j.sjbs.2017.01.044.

[12] "Title of the Article Unknown". In: IEEE Transactions on Magnetics 49.8 (2013). Date of Publication: 18 January 2013, pp. 4811–4816. ISSN:00189464.DOI: 10.1109/TMAG.2013.2241447.

[13] Wen sheng Xiao et al. "A novel chaotic and neighborhood search-based artificial bee colony algorithm for solving optimization problems". In: Scientific Reports 13.20496 (2023). Open access. DOI:10.1038/s4159802328762-1.URL: https://www.nature.com/articles/s41598-023-28762-1.

[14] X. S. Yang, S. Deb, and A. Kar. "An Improved Artificial Bee Colony Algorithm for Optimal Design of Electromagnetic Devices". In: IEEE Transactions on Magnetics 49.8 (2013). Published in IEEE Transactions on Magnetics, Volume 49, Issue 8, August 2013, Date of Publication: 18 January 2013, pp. 4811–4816. DOI:10.1109/TMAG.2013.2241447.

[15] Yinggao Yue et al. "Optimization-Based Artificial Bee Colony Algorithm for Data Collection in Large-Scale Mobile Wireless Sensor Networks". In: International Journal of Distributed Sensor Networks 12.5 (2016), p. 7057490. DOI: 10.1155/2016/7057490.URL: https://doi.org/10.1155/2016/7057490.

[16] Shuliang Zhu et al. "An artificial bee colony algorithm with a balance strategy for wireless sensor network". In: Applied Soft Computing 136 (2023), p.110083. DOI:10.1016/j.asoc.2023.110083.