Advanced and Sustainable Technologies
ASET
E-ISSN 2976-2294

Volume 4, No 1, June 2025 [195-202]

Exploring the Evolution of Artificial Bee Colony Algorithms: Emphasis on Semi-Greedy Strategies

MFMA Hamzas[1*] and H Radhwan[1]

[1]Faculty of Mechanical Engineering & Technology, Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia.

Received 16 April 2025, Revised 5 May 2025, Accepted 13 May 2025

ABSTRACT

The Artificial Bee Colony (ABC) algorithm has emerged as a prominent metaheuristic technique for solving complex optimization problems due to its simplicity, robustness, and bio-inspired behavior. However, standard ABC suffers from limitations such as slow convergence and premature stagnation. To address these issues, numerous variants have been developed, among which the Semi-Greedy Artificial Bee Colony (SGABC) algorithm introduces a significant advancement by incorporating heuristic-driven yet probabilistic decision strategies. This review provides a comprehensive analysis of the evolution of ABC algorithms, with particular emphasis on semi-greedy strategies. It categorizes key modifications, compares SGABC with standard ABC and other metaheuristics, and highlights its superior performance in problems such as Two-Sided Assembly Line Balancing (2SALB). The paper also explores SGABC's industrial applications, identifies current research gaps, and proposes future directions including adaptive control, multi-objective frameworks, and real-time optimization. SGABC is positioned as a robust and scalable optimization framework with strong potential for further theoretical development and industrial deployment.

Keywords: Two-sided assembly line balancing, Artificial bee colony, Semi-greedy algorithm, Metaheuristic, Optimization.

1. INTRODUCTION

Metaheuristic algorithms inspired by natural processes have become indispensable tools in solving complex optimization problems across engineering, manufacturing, logistics, and artificial intelligence domains. These algorithms are especially valuable in dealing with NP-hard problems where exact methods become computationally impractical. Among the most widely studied nature-inspired techniques is the Artificial Bee Colony (ABC) algorithm, introduced by Karaboga in 2005 [1], which draws on the intelligent foraging behavior of honey bee swarms. The ABC algorithm has since gained significant attention due to its simple structure, minimal parameter tuning requirements, and ease of adaptation to various problem types.

ABC is a population-based algorithm that simulates the roles of three types of bees: employed, onlooker, and scout bees. These artificial agents collectively balance global exploration and local exploitation in the search space. The algorithm has been successfully applied to a wide array of applications, including function optimization, scheduling, clustering, image segmentation, and notably, assembly line balancing problems (ALBPs), a core focus of this paper [2, 3].

*Corresponding author: fidzwan@unimap.edu.my

Despite its effectiveness, the standard ABC algorithm suffers from several inherent limitations. These include slow convergence rates, especially in the later stages of optimization, and premature convergence where the search becomes trapped in local optima. Moreover, the random nature of neighborhood solution generation limits the algorithm's ability to exploit known good solutions effectively [4]. These challenges have prompted numerous researchers to propose various improvements and hybridizations over the years.

One particularly promising direction involves the integration of heuristic-based selection mechanisms into the ABC framework. Known as the Semi-Greedy Artificial Bee Colony (SGABC) algorithm, this enhanced variant seeks to retain the explorative nature of ABC while improving its exploitation capabilities through the incorporation of semi-greedy strategies. The semi-greedy approach, inspired by the Greedy Randomized Adaptive Search Procedure (GRASP), utilizes heuristic information to build a restricted candidate list (RCL) from which selections are made probabilistically. This allows the algorithm to focus on promising regions of the solution space without sacrificing population diversity [5].

The SGABC algorithm has shown promising results in various domains, particularly in discrete and constrained optimization problems such as the Two-Sided Assembly Line Balancing (2SALB) problem. Recent studies have demonstrated that SGABC not only improves convergence speed but also delivers more stable and higher-quality solutions compared to both standard ABC and other swarm-based metaheuristics [6, 7].

This review paper provides a comprehensive overview of the evolution of the ABC algorithm, with a particular emphasis on semi-greedy modifications. It aims to categorize the major variants of ABC, highlight the role of SGABC in enhancing optimization performance, evaluate its comparative effectiveness, and outline its application potential in industrial settings. Furthermore, it identifies existing research gaps and proposes future directions to improve the scalability, adaptiveness, and industrial relevance of SGABC-based optimization frameworks.

2. BACKGROUND AND ALGORITHMIC FOUNDATIONS

The Artificial Bee Colony (ABC) algorithm is a population-based metaheuristic developed by Dervis Karaboga in 2005. Inspired by the foraging behavior of honey bees, the algorithm simulates the collective intelligence of a bee colony in locating, evaluating, and exploiting food sources. In the context of optimization, these food sources represent candidate solutions, and the goal is to iteratively improve them to identify the most optimal solution. The division of labor among employed bees, onlooker bees, and scout bees enables a dynamic balance between exploration of the global search space and exploitation of known high-quality solutions [1]. Due to its biologically inspired structure, the ABC algorithm is both intuitive and computationally efficient. It has become a widely adopted technique for solving a broad range of continuous and discrete optimization problems, including scheduling, routing, clustering, and manufacturing systems. Its popularity stems from its simplicity of implementation, minimal reliance on algorithm-specific parameter tuning, and strong global search capabilities that make it well-suited for tackling complex, nonlinear, and multimodal problems [8].

To address the aforementioned limitations, several modifications and hybrid approaches have been proposed. Table 1 summarizes key enhancements to the ABC algorithm, highlighting their objectives, techniques, and representative studies.

Table 1: Key enhancements to the ABC Algorithm.

Enhancement Type	Objective	Example Techniques / Variants	Key References
Modified Search Operators	Improve local search and convergence speed	Adaptive step size, Lévy flight, opposition-based learning	Zhang et al. (2023) [2]
Hybrid Metaheuristics	Combine ABC with other algorithms for enhanced performance	ABC-GA, ABC-PSO, ABC-DE	Cao et al. (2023) [9]
Elitism and Memetic Strategies	Retain the best solutions and apply local refinements	Memetic ABC, elitist replacement	Bansal et al. (2020) [7]
Self-Adaptive / Parameter-Free ABC	Reduce manual tuning of parameters during execution	Dynamic population size, fuzzy logic controllers	Huang et al. (2023) [10]
Heuristic-Guided (Semi-Greedy) ABC	Guide solution construction using informed probabilistic choices	Restricted Candidate List (RCL), biased selection	Feo & Resende (2020) [5]

One notable enhancement to the Artificial Bee Colony (ABC) algorithm is the integration of semi-greedy strategies into its core framework. Originally introduced in the context of Greedy Randomized Adaptive Search Procedures (GRASP) by Feo and Resende (1995) [11], semi-greedy methods involve constructing a Restricted Candidate List (RCL) based on heuristic evaluations, from which selections are made probabilistically. This hybrid approach offers a practical balance between the intensification of greedy methods and the diversification offered by stochastic search, making it well-suited for large and complex search spaces [5].

In the context of ABC, these strategies can be embedded during both the employed bee and onlooker bee phases, where bees use task-specific heuristics such as precedence weight, processing time, or positional priority to make more informed search decisions. Such integration enhances the algorithm's ability to escape local optima and accelerates convergence, especially in constrained combinatorial environments [2, 12].

The performance limitations of standard ABC, particularly in discrete, multi-constraint settings, have driven the evolution of the Semi-Greedy Artificial Bee Colony (SGABC) algorithm. SGABC combines the exploratory power of ABC with heuristic-guided exploitation, resulting in improved convergence speed and solution quality. It effectively maintains search diversity through probabilistic selection while directing efforts toward promising regions in the solution space.

Recent empirical findings confirm the algorithm's practical value. For instance, Amin Hamzas et al. (2023) [6] applied a hybrid SGABC model to solve the Two-Sided Assembly Line Balancing (2SALB) problem and reported a significant reduction in the number of workstations used and better convergence stability compared to standard ABC, PSO, and GA. These results validate the SGABC algorithm as a robust and scalable optimization method for solving real-world discrete and constraint-driven problems.

3. DEVELOPMENT OF SEMI-GREEDY STRATEGIES

The Semi-Greedy Artificial Bee Colony (SGABC) algorithm was introduced to address fundamental weaknesses of the standard Artificial Bee Colony (ABC) framework, particularly its susceptibility to premature convergence, limited exploitation ability, and performance instability in discrete or constrained problem domains. While the original ABC algorithm excels at global exploration, it often lacks the capability to refine promising solutions effectively, especially in complex combinatorial spaces. This shortcoming primarily arises from its heavy dependence on random neighbor generation and greedy selection mechanisms, which can cause the search to stagnate in local optima [2, 4].

To counteract this, SGABC integrates a semi-greedy selection mechanism that balances heuristic guidance with controlled randomness. The strategy involves constructing a Restricted Candidate List (RCL), a concept rooted in Greedy Randomized Adaptive Search Procedures [5], during each solution update. Rather than deterministically choosing the best candidate or blindly sampling the neighborhood, SGABC evaluates multiple potential solutions based on heuristics such as task time, positional weights, or precedence constraints, and includes only the most promising candidates in the RCL. One solution is then probabilistically selected from this pool. This procedure is embedded in both the employed bee and onlooker bee phases, ensuring that the algorithm emphasizes promising areas of the search space while maintaining diversity to avoid local traps.

A defining strength of SGABC lies in its adaptive greediness control. Some versions utilize a static threshold to determine RCL composition, while others dynamically adjust the RCL size or threshold throughout the search based on convergence behavior, population diversity, or fitness distribution. For example, larger RCLs are used in the early search phase to promote broad exploration, while smaller, greedier selections are favored as convergence nears. This dynamic strategy enhances convergence speed and stability, especially in multi-objective or time-sensitive optimization problems [12].

To further boost its effectiveness, SGABC is often hybridized with trajectory-based local search methods. Examples include 2-opt for combinatorial optimization, greedy repair for infeasible solutions, or path relinking techniques. These hybridizations marry the global search power of swarm intelligence with the fine-tuning accuracy of local refinement, resulting in improved convergence quality and reduced stagnation risks. Moreover, researchers have recently explored learning-based SGABC variants, integrating fuzzy decision-making or reinforcement learning to tune heuristic weightings on the fly in uncertain or dynamic problem settings [2].

Recent comparative studies underscore SGABC's superiority over traditional metaheuristics. Hamzas et al. (2023) [6] applied SGABC to benchmark datasets of the Two-Sided Assembly Line Balancing (2SALB) problem and reported consistent improvements in both convergence speed and solution quality over standard ABC, PSO, and GA. Specifically, the study achieved up to 15% reduction in the number of workstations and 30% faster convergence, while maintaining 100% feasibility across multiple runs. Similarly, Liu et al. (2024) [13] demonstrated SGABC's ability to solve high-dimensional scheduling problems with constrained resources more effectively than ACO and DE by leveraging its balanced search strategy.

In summary, SGABC represents a significant evolutionary leap in swarm-based optimization algorithms. Its semi-greedy structure enables informed yet flexible decision-making, allowing it to solve constrained, discrete, and dynamic optimization problems more robustly than traditional ABC. With its adaptability, hybridization capability, and successful deployment in real-world scenarios, SGABC is poised to serve as a cornerstone in the next generation of intelligent optimization tools. Table 2 below indicates the key SGABC developments.

Table 2: Key of SGABC Developments.

Study / Author	Application Domain	SGABC Feature	Key Enhancements	Performance Outcome
Hamzas et al. (2023) [6]	2SALB (Assembly Line Balancing)	Semi-greedy selection + static RCL	Hybridized with assignment slot encoding	15% fewer workstations, faster convergence
Singh et al. (2022) [12]	Multi-objective job scheduling	Dynamic RCL adaptation	Time-based greediness control	Improved Pareto front coverage, fewer iterations
Liu et al. (2024) [13]	Constrained resource scheduling	Heuristic-driven candidate ranking	Precedence + urgency-based heuristics	30% better task balancing under resource constraints
Zhang et al. (2023) [2]	Uncertain task assignment	Learning-based RCL generation	Reinforcement learning for heuristic tuning	High stability under uncertainty, adaptive search depth
Bansal et al. (2020) [7]	General engineering design	Semi-greedy local search hybridization	2-opt local refinements post-selection	Better convergence quality, robust against noise

4. COMPARATIVE PERFORMANCE ANALYSIS

This section presents a comprehensive comparative evaluation of the Semi-Greedy Artificial Bee Colony (SGABC) algorithm against the standard ABC algorithm and other well-established metaheuristic approaches, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). SGABC is specifically designed to overcome performance limitations found in traditional ABC, and its effectiveness is analyzed here through a set of widely recognized performance indicators. These indicators include solution quality, convergence speed, feasibility rate, and robustness, all of which provide a holistic perspective on algorithm performance in complex optimization environments [3, 4].

The analysis draws from recent empirical studies and benchmark applications, particularly within the domain of discrete and combinatorial optimization problems such as the Two-Sided Assembly Line Balancing (2SALB) problem, a well-known NP-hard challenge in manufacturing systems optimization. SGABC has demonstrated a notable ability to produce higher-quality solutions by minimizing workstation usage, optimizing cycle time, and achieving better load distribution among stations [6]. In terms of convergence behavior, SGABC consistently achieves near-optimal solutions in fewer iterations compared to standard ABC and GA, owing to its adaptive greedy mechanism that biases selection toward high-potential regions of the search space while maintaining diversity [13].

Feasibility and robustness are also central to this evaluation. In studies involving constraint-heavy scheduling and task allocation problems, SGABC maintained a near-100% feasibility rate across multiple test instances, indicating its strong capacity to navigate search spaces with precedence constraints, side assignments, and resource limitations [12]. Furthermore, the algorithm has shown high robustness, with low variance in results across multiple independent runs, highlighting its reliability for industrial and real-time applications.

By consolidating findings across multiple studies, this comparative section aims to establish SGABC's superiority over standard ABC and demonstrate its competitiveness with other hybrid metaheuristics. The next sub-sections provide detailed quantitative comparisons and performance charts reinforcing these conclusions.

4.1 Evaluations Metric

To ensure a fair and comprehensive comparison of optimization algorithms, particularly the Semi-Greedy Artificial Bee Colony (SGABC) algorithm against standard ABC and other metaheuristics, a well-defined set of performance metrics is adopted. These metrics not only assess the final quality of the solutions produced but also provide insights into the algorithm's computational behavior, adaptability, and reliability. By evaluating key dimensions such as effectiveness, efficiency, feasibility, and stability, these metrics allow for a holistic assessment that is both practically relevant and scientifically rigorous. The selected indicators are widely recognized in optimization literature and are especially critical in evaluating algorithms applied to complex, discrete, and constrained problems such as two-sided assembly line balancing (2SALB). The following four metrics serve as the primary benchmarks in this comparative study: ensure a fair and comprehensive comparison. The following metrics are used:

- **Solution Quality**: This refers to the optimality of the solutions produced, often quantified by the number of workstations used, cycle time, or objective function value in optimization problems. High solution quality reflects the algorithm's capability to find near-optimal or optimal solutions in structured problem spaces. SGABC has demonstrated superior solution quality in recent studies, particularly in discrete manufacturing optimization [6, 13].
- **Convergence Speed**: This measures how quickly the algorithm approaches high-quality solutions, typically by counting the number of iterations or the computational time required. Algorithms with faster convergence reduce processing costs and time-to-decision, which is critical in real-time industrial settings. SGABC's hybrid semi-greedy mechanism has shown improved convergence behavior in constrained optimization scenarios [12].
- **Feasibility Rate**: Defined as the proportion of algorithm runs that produce valid, constraint-satisfying solutions, this metric is vital for real-world applications where infeasible outputs can render an algorithm unusable. Studies have reported that SGABC consistently maintains near 100% feasibility in problems involving precedence constraints and side allocations [3].
- **Robustness**: This refers to the algorithm's ability to produce consistent results across multiple independent runs, indicating its stability and reliability. Robust algorithms show minimal performance variation under different initial conditions or problem instances. SGABC has exhibited strong robustness across benchmark datasets, reinforcing its practical applicability [2, 4].

4.2 SGABC vs Standard ABC

Numerous studies have shown that SGABC significantly improves upon the limitations of the standard ABC algorithm. By incorporating semi-greedy selection during the solution construction phase, SGABC demonstrates faster convergence and produces higher-quality solutions, particularly in structured problems such as task scheduling and assembly line design.

SGABC consistently demonstrates reduced idle time in workstations and more efficient balancing outcomes when applied to 2SALB instances, outperforming the standard ABC in terms of both effectiveness and reliability. Table 3 show the metric evaluations of Standard ABC vs SGABC.

Table 3: Metric Evaluations.

Metric	Standard ABC	SGABC
Solution Quality	Moderate	High
Convergence Speed	Slower	Faster
Feasibility Rate	Variable	High and consistent
Robustness	Sensitive to parameters	Stable across multiple runs

5. CONCLUSION AND FUTURE DIRECTION

The evolution of Artificial Bee Colony algorithms reflects the growing demand for flexible and effective optimization methods across multiple domains. The integration of semi-greedy strategies into ABC represents a substantial advancement by improving intensification without sacrificing exploratory potential. SGABC offers a compelling balance between randomness and heuristic guidance, making it suitable for solving discrete, constrained, and combinatorial problems such as two-sided assembly line balancing.

This review has highlighted the strengths of SGABC in terms of convergence rate, robustness, and practical applicability. However, challenges remain in parameter tuning, scalability to multi-objective scenarios, and real-time responsiveness. Future research should prioritize adaptive and self-regulating frameworks, integration with reinforcement learning, and deployment in Industry 4.0 environments via parallel and real-time optimization mechanisms. Overall, SGABC holds considerable promise as a next-generation optimization tool, capable of bridging theoretical metaheuristics and practical decision support in complex systems.

ACKNOWLEDGEMENTS

The author sincerely appreciates the Faculty of Mechanical Engineering & Technology’s crucial support and resources during this research.

REFERENCES

[1] Karaboga, D.. Artificial bee colony algorithm. scholarpedia, vol 5, issue 3 (2010) p. 6915.
[2] Zhang, J., Liu, S., & Hou, J.. A learning-based semi-greedy ABC algorithm for adaptive task assignment under uncertainty. Expert Systems with Applications, vol 213 (2023) p. 119098.
[3] Akyol, S., Avci, E., & Kalayci, C. B.. A comprehensive review of artificial bee colony algorithm and its applications. Journal of Computational Science, vol 52 (2021) p. 101401.
[4] Ahmad, N., Ali, M., Rehman, M. U., & Alshahrani, S.. Adaptive artificial bee colony algorithm with elite guidance for complex engineering design. Mathematics, vol 10, issue 17 (2022) p. 3085.
[5] Feo, T. A., & Resende, M. G. C.. Greedy randomized adaptive search procedures. Journal of Global Optimization, vol 6, issue 2 (2020) pp. 109-133.
[6] Amin Hamzas, M. F. M., Bareduan, S. A., Zakaria, M. Z., & Sin, T. C.. An assignment slots solution approach for the two-sided assembly line balancing by using a hybrid semi-greedy artificial bee colony algorithm. AIP Conference Proceedings, vol 2544, issue 1 (2023) p. 040018.
[7] Bansal, J. C., Sharma, H., & Arya, K. V.. Memetic artificial bee colony algorithm. Soft Computing, vol 23 (2020) pp. 10529-10554.
[8] Karaboga, D., & Basturk, B.. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, vol 39 (2007) pp. 459-471.

[9] Cao, W., Xu, J., Zhang, Y., Zhao, S., Xu, C., & Wu, X.. A hybrid discrete artificial bee colony algorithm based on label similarity for solving point-feature label placement problem. ISPRS International Journal of Geo-Information, vol 12, issue 10 (2023) p. 429.
[10] Huang, Y., Yu, Y., Guo, J., & Wu, Y.. Self-adaptive Artificial Bee Colony with a Candidate Strategy Pool. Applied Sciences, vol 13, issue 18 (2023) p. 10445.
[11] Feo, T. A., & Resende, M. G.. Greedy randomized adaptive search procedures. Journal of global optimization, vol 6 (1995) pp. 109-133.
[12] Singh, R., Sharma, V., & Bansal, J. C.. A dynamic semi-greedy ABC algorithm for multi-objective production scheduling. Engineering Applications of Artificial Intelligence, vol 112 (2022) p. 104998.
[13] Liu, Q., Yang, X., & Zhang, Y.. Semi-greedy artificial bee colony algorithm for constrained scheduling: Balancing exploration and exploitation. Applied Soft Computing, vol 148 (2024) p. 110948.

Conflict of interest statement: The authors declare no conflict of interest.

Author contributions statement: Conceptualization; SGABC Methodology; Algorithms; Computational experiments; Performance; Writing & Editing, Mohd Fidzwan Md Amin Hamzas & H Radhwan.