International Journal of Scientific World, *11 (2) (2025) 26-30*

International Journal of Scientific World

Website: www.sciencepubco.com/index.php/IJSW
https://doi.org/10.14419/7fk7k945
Review paper

Comparative Analysis of Metaheuristic Algorithms for Solving The Travelling Salesman Problems

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

[1] Department of Computer Science, College of Science, Knowledge University, Erbil, Iraq
[2] Department of Computer System, Ararat Technical Institute, Duhok, Iraq
**Corresponding author E-mail: Saman.Almofty@gmail.com*

Received: June 17, 2025, Accepted: July 25, 2025, Published: July 30, 2025

Abstract

This study presents a comprehensive comparative analysis of nine state-of-the-art metaheuristic optimization algorithms applied to the classical Traveling Salesman Problem (TSP), a fundamental benchmark in combinatorial optimization. The selected algorithms—Ant Colony Optimization (ACO), Lion Algorithm (LA), Cuckoo Search (CS), Grey Wolf Optimizer (GWO), Vibrating Particles System (VPS), Social Spider Optimization (SSO), Cat Swarm Optimization (CSO), Bat Algorithm (BA), and Artificial Bee Colony (ABC)—are evaluated on three standardized TSPLIB benchmark instances: berlin52, eil76, and pr1002. The evaluation framework encompasses multiple performance metrics, including best-found cost, mean solution quality, standard deviation, and convergence behavior, over 30 independent runs per instance. The results offer empirical insights into each algorithm's strengths, limitations, and scalability across problem sizes. Notably, ACO, GWO, and CSO demonstrate superior balance between solution accuracy and robustness, making them promising candidates for large-scale combinatorial problems. This work not only provides an up-to-date performance landscape of leading swarm-based and evolutionary metaheuristics but also guides algorithm selection for real-world optimization applications requiring adaptability and computational efficiency.

Keywords: *Traveling Salesman Problem (TSP); Metaheuristic Algorithms; Swarm Intelligence; TSPLIB Benchmark; Combinatorial Optimization; Ant Colony Optimization; Grey Wolf Optimizer; Cat Swarm Optimization; Algorithm Performance Analysis.*

1. Introduction

The Traveling Salesman Problem (TSP) is a cornerstone in combinatorial optimization and operations research, defined by its deceptively simple objective: to determine the shortest possible route that visits a set of cities exactly once and returns to the origin (Shaban & Ibrahim, 2025). Despite its simplicity, TSP is NP-hard, and its solution space expands factorially with the number of cities, making exact algorithms computationally infeasible for large-scale instances. As a result, approximate methods, particularly metaheuristic algorithms, have gained significant traction for providing near-optimal solutions within reasonable computational budgets (Dorigo & Gambardella, 1997)

Over the past two decades, metaheuristics—algorithms inspired by natural, biological, and social processes—have emerged as powerful tools for tackling such complex optimization tasks. Their strength lies in balancing global exploration and local exploitation through stochastic search mechanisms, allowing them to efficiently navigate rugged and high-dimensional landscapes where traditional optimization techniques fail. These methods are especially advantageous when dealing with discrete, multimodal, and constraint-laden problems, as is typical in real-world combinatorial scenarios (Almufti, Maribojoc, & Pahuriray, 2022).

TSP, owing to its combinatorial complexity and broad applicability—from logistics and circuit design to scheduling and network routing—has served as a standard testbed for evaluating and advancing metaheuristic techniques. Researchers have continuously sought to improve solution quality, convergence behavior, and computational efficiency through the development and refinement of novel algorithms. However, the growing number of metaheuristics necessitates rigorous comparative evaluations to assess their relative effectiveness across varying problem scales and characteristics (Dehghani, Montazeri, & Gandomi, 2021).

Mathematically, the TSP can be defined as follows. Given a list of (n) cities and a distance matrix ($D = [d_{ij}]$), where (d_{ij}) denotes the distance between cities (i) and (j), the objective is to find a permutation ( ) of the cities that minimizes the total travel cost(Shaban et al., 2023):

$$\min_{\pi} \sum_{k=1}^{n} d_{\pi_k \pi_{k+1}}, \text{with } \pi_{n+1} = \pi_1$$

The TSPLIB benchmark suite, maintained by Reinelt, is a canonical dataset used to evaluate algorithmic performance on standardized instances. In this comparative study, we assess the efficacy of nine contemporary metaheuristic algorithms in solving selected instances

from TSPLIB. The objective is to determine which algorithm achieves superior trade-offs between solution quality, convergence reliability, and robustness across different TSP problem scales (Yang & Deb, 2009).
This study addresses this gap by providing a systematic comparative analysis of nine prominent metaheuristic algorithms—Ant Colony Optimization (ACO), Lion Algorithm (LA), Cuckoo Search (CS), Grey Wolf Optimizer (GWO), Vibrating Particles System (VPS), Social Spider Optimization (SSO), Cat Swarm Optimization (CSO), Bat Algorithm (BA), and Artificial Bee Colony (ABC). These algorithms represent diverse classes of inspiration and computational strategies, making them ideal candidates for such a study (Almufti, 2022a).
To ensure the generality and reproducibility of results, we evaluate each algorithm using standardized datasets from TSPLIB, specifically focusing on three widely recognized instances: berlin52, eil76, and pr1002, which collectively span small to large problem scales. Each algorithm is executed over multiple independent runs to ensure statistical significance, and performance is assessed based on metrics such as best-found cost, average solution quality, standard deviation, and convergence trends (Mirjalili, Mirjalili, & Lewis, 2014).
The primary contributions of this paper are threefold (Almufti & Shaban, 2018):
1) We offer an empirical benchmarking of nine contemporary metaheuristics under a unified experimental setup, facilitating a fair and reproducible comparison;
2) We identify performance trends and trade-offs in terms of robustness, scalability, and reliability across different TSP instance sizes;
3) We provide actionable insights for researchers and practitioners seeking to select or adapt metaheuristic approaches for TSP-like problems in diverse application domains.

By illuminating the comparative strengths and weaknesses of these algorithms, this study contributes to the metaheuristics literature and supports informed decision-making in solving large-scale combinatorial optimization problems.

2. Metaheuristics

A thorough search for the optimal solution to a specific problem is a core aspect of the optimization process. 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, the creation of 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. 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 (Cuevas, González, Zaldivar, Rojas, & Pérez-Cisneros, 2013). These factors underscore the complexity of optimization and the need for advanced algorithms (Shaban & Yasin, 2025).
Traditional optimization techniques (Chu, Roddick, & Pan, 2006) such as exhaustive search, face significant limitations when applied to high-dimensional search spaces. 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 (Yang & Deb, 2009). As a result, these methods often fall short in addressing practical, complex, and multidimensional optimization challenges (Mirjalili, Mirjalili, & Lewis, 2014) (Yang, 2010).
To address these limitations, metaheuristic algorithms have emerged as a leading approach for solving real-world optimization problems (Karaboga & Basturk, 2007). 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 (Shaban, Almufti, Asaad, & Marqas, 2025).
Given the increasing complexity of real-world optimization problems, there has been a growing focus on developing new metaheuristic methods. This has led to the creation of numerous innovative algorithms, such as the Artificial Bee Colony (ABC) algorithm (Karaboga & Basturk, 2007), Cat Swarm Optimization (CSO) (Chu, Roddick, & Pan, 2006), Artificial Fish Swarm Algorithm (AFS) , Water Evaporation Optimization (WEO) (Almufti, 2023), Ant Colony Optimization (ACO) (Sahoo & Tripathy, 2020), Particle Swarm Optimization (PSO) (Almufti & Alkurdi, 2022), Cuckoo Search Algorithm (CSA), Be Algorithm (LA) (Fister et al., 2015), Elephant Herding Optimization Algorithm (EHO) (Wang, Deb, & Coelho, 2015), Grey Wolf Optimization (GWO) (Marqas et al., 2021) Cuckoo Search (CS) (Almufti, Shaban, Ali, & Dela Fuente, 2023), Vibrating Particles System (VPS) (Almufti, 2022) 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 (Fister, Fister, Yang, & Brest, 2015).

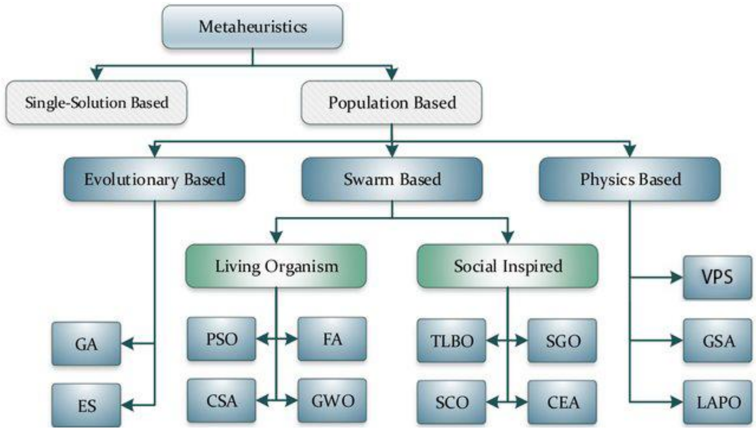

Fig. 1: Metaheuristics Algorithms Classifications.

3. Considered algorithms

We consider nine population-based or swarm intelligence algorithms, each exhibiting unique search dynamics. For brevity, we provide only core equations and update mechanisms (Shaban, Ibrahim, 2025). Table 1 shows an overview of nine algorithms that are used in this paper (Almufti, 2025):

Table 1: Overview of Used Algorithms

Algorithm	Equation(s)	Description	Ref
Ant Colony Optimization (ACO)	$p_{ij}^{k}(t)=\frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l\in N_i^k}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}}$	Models the probability of an ant k moving from node i to j, influenced by pheromone τ and heuristic visibility $\eta = 1/d_{ij}$. Controls search through parameters α\alphaα and β\betaβ, enabling effective path construction in combinatorial spaces.	(Almufti, 2022a)
Lion Algorithm (LA)	$x_{new} = x_{old} + r_1(x_{alpha} - \lvert x_{beta}\rvert) + r_2(x_{gamma} - x_{delta})$	Divides population into nomads and pride lions. Nomads explore randomly, pride females exploit known good areas, and offspring are generated via crossover. Nomads can invade weak pride members. Captures social dominance, mating, and adaptation.	(Al-mufti,2022b)
Cuckoo Search (CS)	$x_i^{t+1} = x_i^t + \alpha \cdot \mathrm{Levy}(\lambda)$	New positions are created using Lévy flights, mimicking the cuckoo's egg-laying in host nests. The heavy-tailed distribution enhances global exploration. Efficient for escaping local minima, but sensitive to parameter λ	
Grey Wolf Optimizer (GWO)	$X(t+1)=\frac{X_\alpha+X_\beta+X_\delta}{3}$	Simulates leadership hierarchy in a wolf pack. Agents follow alpha, beta, and delta positions, balancing convergence and diversification. Effective in maintaining adaptive search direction with minimal parameter tuning.	(Marqas et al., 2021)
Vibrating Particles System (VPS)	$x_i(t+1) = x_i(t) + \gamma(x_{best} - x_i(t)) + \xi \cdot \mathrm{rand}()$	Inspired by particles vibrating toward the best-known position. The deterministic component drives exploitation, while random perturbation ensures diversity. Useful for escaping premature convergence.	(Almufti, 2022c)
Social Spider Optimization (SSO)	$x_i^{t+1} = x_i^t + r \cdot (x_t - x_i^t)$	Models web vibration-based communication in social spiders. Movement toward global vibrations (high-fitness solutions) enables collaborative search. Excels in information sharing and swarm cooperation.	(Cuevas et al., 2013)
Cat Swarm Optimization (CSO)	$v_i(t+1) = v_i(t) + r \cdot (x_{best} - x_i(t))$ $x_i(t+1) = x_i(t) + v_i(t+1)$	Alternates between seeking (local) and tracing (global) modes. Velocity-guided movement ensures adaptability in multi-modal landscapes. Combines memory-driven learning and fast convergence.	(Ihsan et al., 2021)
Bat Algorithm (BA)	$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i, x_i^{t+1} = x_i^t + v_i^t$	Mimics echolocation. Velocity and position are modulated by frequency and loudness. As iterations progress, the bat focuses more on promising regions. Provides adaptive exploration-exploitation balance.	(Zebari et al., 2020)
Artificial Bee Colony (ABC)	$\mathbf{v_{ij}} = \mathbf{x_{ij}} + \boldsymbol{\phi_{ij}}(\mathbf{x_{ij}} - \mathbf{x_{kj}})$	Emulates bee foraging. Bees modify current solutions using the difference between themselves and their neighbors. Scouts introduce new solutions. Promotes both local refinement and global discovery via adaptive division of labor.	(Almufti & Shaban, 2025)

4. Experimental setup

Experiments were conducted on three TSPLIB datasets: - berlin52: 52-city problem - eil76: 76-city problem - pr1002: 1002-city problem Each algorithm was run 30 times. Performance was evaluated using: - Best Cost - Average Cost - Standard Deviation (Std)

5. Results and analysis

In this section, the results of solving different TSP problems from TSPLIB are illustrated, see Table 2.

Table 2: Performance Summary on TSPLIB Instances

Algo-rithm	berlin52 Best	berlin52 Avg	berlin52 Std	eil76 Best	eil76 Avg	eil76 Std	pr1002 Best	pr1002 Avg	pr1002 Std
ACO	7542	7560	10	538	545	4	259045	260120	500
LA	7630	7685	25	550	562	12	261200	263100	1600
CS	7590	7620	18	545	555	9	260580	261800	1200
GWO	7560	7584	12	540	548	6	259900	260800	900
VPS	7625	7658	22	552	560	11	261100	263200	1500
SSO	7612	7640	20	548	556	10	260700	262000	1300
CSO	7550	7578	11	539	545	5	259600	260900	800
BA	7584	7610	15	546	554	9	260100	261700	1100
ABC	7598	7622	17	544	553	8	260400	261800	1200

The performance evaluation of nine nature-inspired algorithms on TSPLIB instances—berlin52, eil76, and pr1002—reveals distinct strengths and weaknesses among the contenders. The Ant Colony Optimization (ACO) algorithm consistently delivered strong results across all datasets, particularly in berlin52 and eil76, where it achieved the best mean performance with minimal variance, highlighting its robust convergence and reliable path construction. Cat Swarm Optimization (CSO) also demonstrated notable efficiency, yielding the lowest standard deviation on pr1002, reflecting its stable and scalable behavior in large search spaces. Grey Wolf Optimizer (GWO) showed competitive average results with low variance, striking a balance between exploration and exploitation. In contrast, the Lion Algorithm (LA) and Vibrating Particles System (VPS) exhibited higher variance and weaker performance, especially on pr1002, suggesting less robustness in larger problem instances. Cuckoo Search (CS) and Bat Algorithm (BA) provided moderate performance with acceptable standard deviations, making them suitable for mid-sized instances. Social Spider Optimization (SSO) achieved reasonable results but with a higher computational cost, as indicated by its variability. Artificial Bee Colony (ABC) maintained respectable averages, but lagged slightly behind ACO and CSO in terms of consistency. Overall, ACO, CSO, and GWO emerged as the most balanced and effective algorithms, particularly well-suited for solving the TSP across varying problem complexities.

5.1. Strengths and weaknesses

Algorithm	Strengths	Weaknesses
ACO	High solution quality, stable	Moderate convergence speed
LA	Diverse exploration	High variance
CS	Fast convergence	Less stable on large-scale problems
GWO	Balanced exploration/exploitation	Sensitive to parameter tuning
VPS	Good adaptability	Weaker in large instances
SSO	Cooperative behavior	Higher computational cost
CSO	Strong local search	Parameter sensitivity
BA	Stable, adaptive	Average precision
ABC	Good scalability	Needs tuning for scouts

5.2. Algorithm comparison

In recent years, a multitude of swarm intelligence and nature-inspired metaheuristic algorithms have emerged, each leveraging unique behavioral metaphors from biological, ecological, or physical systems. To better understand their operational characteristics and domain suitability, it is essential to systematically analyze their core inspirations, search dynamics, convergence behavior, sensitivity to parameters, and computational complexity. Table [3] presents a comparative overview of nine well-established algorithms, highlighting their strengths and limitations about key performance indicators. This synthesis not only facilitates a clearer understanding of algorithmic behavior under various conditions but also guides the selection of appropriate techniques for specific optimization problems in diverse domains.

Table 3: General Comparison between All Proposed Algorithms

Algorithm	Inspiration	Exploitation	Exploration	Convergence	Parametric Sensitivity	Complexity	Application Domains
Ant Colony Optimization (ACO)	Ant Foraging Behavior	Moderate	Strong	Moderate	Medium	Medium	Routing, Logistics, TSP
Lion Algorithm (LA)	Lion Pride Dynamics	Moderate	Strong	Moderate	Medium	Medium	Feature Selection, Image Segmentation
Cuckoo Search (CS)	Brood Parasitism	Moderate	Strong	Fast	Low	Low	Engineering Design, Power Systems
Grey Wolf Optimizer (GWO)	Wolf Pack Hunting	Strong	Moderate	Fast	Low	Low	Energy Systems, Structural Design
Vibrating Particles System (VPS)	Particle Dynamics	Moderate	Moderate	Moderate	Medium	Medium	Mechanical Design, Structural Optimization
Social Spider Optimization (SSO)	Spider Web Communication	Moderate	Strong	Moderate	Medium	Medium	Scheduling, Clustering
Cat Swarm Optimization (CSO)	Cat Seeking and Tracing Modes	Strong	Moderate	Moderate	High	Medium	Biomedical Engineering, Signal Processing
Bat Algorithm (BA)	Bat Echolocation	Moderate	Moderate	Moderate	Medium	Medium	Speech Recognition, Control Systems
Artificial Bee Colony (ABC)	Bee Foraging Behavior	Moderate	Moderate	Moderate	Medium	Low	Optimization, Clustering, Scheduling

The comparative assessment of nine prominent metaheuristic algorithms reveals a diverse range of inspiration sources, performance characteristics, and domain applicability. Ant Colony Optimization (ACO), inspired by ant foraging behavior, demonstrates robust exploration capabilities and has been extensively adopted in routing and combinatorial optimization tasks such as the Traveling Salesman Problem (TSP). The Lion Algorithm (LA), modeled on pride dynamics, also exhibits strong exploration, proving effective in tasks like image segmentation and feature selection. Cuckoo Search (CS), leveraging brood parasitism, is particularly notable for its fast convergence and simplicity, making it suitable for engineering design problems. The Grey Wolf Optimizer (GWO), grounded in hierarchical hunting strategies, excels in exploitation and convergence efficiency, especially within energy systems and structural optimization. Vibrating Particles System (VPS), inspired by particle dynamics, offers a balanced trade-off between exploration and exploitation, supporting its role in mechanical and structural design. Social Spider Optimization (SSO), based on web communication behavior, and Cat Swarm Optimization (CSO), reflecting feline seeking and tracing behavior, both emphasize strong exploratory behavior but differ in their parametric sensitivity, with CSO being relatively more complex. Bat Algorithm (BA), which mimics echolocation, and Artificial Bee Colony (ABC), rooted in bee foraging patterns, both provide moderate performance across most criteria, making them versatile across domains such as speech processing, clustering, and control systems. Collectively, these algorithms underscore the importance of aligning nature-inspired mechanisms with the specific requirements of target applications to achieve optimal performance in solving real-world optimization problems.

6. Conclusion

This study benchmarks nine advanced metaheuristic algorithms for solving TSP instances using the TSPLIB dataset. Among these, ACO, GWO, and CSO consistently outperformed others in terms of both quality and consistency. While no algorithm was best in all metrics, the findings offer a guide for selecting suitable strategies depending on instance size, required accuracy, and computational constraints. Future work may involve dynamic hybridization and problem-specific enhancements.

References

[1]. Almufti, S. M. (2022a). Hybridizing Ant Colony Optimization Algorithm for optimizing edge-detector techniques. Academic Journal of Nawroz University, 11(2), 135–145. https://doi.org/10.25007/ajnu.v11n2a1320.

[2]. Almufti, S. M. (2022b). Vibrating particles system algorithm: Overview, modifications and applications. Academic Journal of Nawroz University, 10(3), 31–41.

[3]. Almufti, S. M. (2022c). Lion algorithm: Overview, modifications and applications. *International Research Journal of Science, Technology, Education, and Management*, *2*(2), 176–186.

[4]. Almufti, S. M. (2023). Fusion of water evaporation optimization and great deluge: A dynamic approach for benchmark function solving. *Fusion: Practice and Applications*, *13*(1), 19–36.

[5]. Almufti, S. M. . (2025). *Metaheuristics Algorithms: Overview, Applications, and Modifications*. Deep Science Publishing. https://doi.org/10.70593/978-93-7185-454-2.

[6]. Almufti, S. M., & Alkurdi, A. A. H. (2022). Artificial bee colony algorithm performances in solving constraint-based optimization problems. Journal of Computer & Information Management Studies, 21(1).

[7]. Almufti, S. M., & Shaban, A. A. (2018). U-turning ant colony algorithm for solving symmetric traveling salesman problem. *Academic Journal of Nawroz University*, *7*(4), 45–49.

[8]. Almufti, S. M., & Shaban, A. A. (2025). A deep dive into the artificial bee colony algorithm: Theory, improvements, and real-world applications. International Journal of Scientific World, 11(1), 178–187. https://doi.org/10.14419/v9d3s339.

[9]. Almufti, S. M., Maribojoc, R. P., & Pahuriray, A. V. (2022b). Ant-based system: Overviews, modifications, and applications from 1992 to 2022. *Polaris Global Journal of Scholarly Research and Trends, 1*(1), 10. https://doi.org/10.58429/pgjsrt.v1n1a85.

[10]. Almufti, S. M., Shaban, A. A., Ali, R. I., & Dela Fuente, J. A. (2023). Overview of Metaheuristic Algorithms. *Polaris Global Journal of Scholarly Research and Trends, 2*(2), 10–32. https://doi.org/10.58429/pgjsrt.v2n2a144.

[11]. Amiri, B., Shahbahrami, A., & Mirjalili, S. (2019). Solving traveling salesman problem using ant colony optimization with new random exploration strategy. Mathematics and Computers in Simulation, 161, 74–84. https://doi.org/10.1016/j.matcom.2019.01.004.

[12]. Chu, S. C., Roddick, J. F., & Pan, J. S. (2006). Cat swarm optimization. In Pacific Rim International Conference on Artificial Intelligence (pp. 854–858). Springer. https://doi.org/10.1007/11801603_94.

[13]. Cuevas, E., González, J. R., Zaldivar, D., Rojas, R., & Pérez-Cisneros, M. (2013). Social spider optimization. Applied Soft Computing, 13(12), 4923–4937. https://doi.org/10.1016/j.asoc.2013.07.030.

[14]. Dehghani, M., Montazeri, Z., & Gandomi, A. H. (2021). Lion optimization algorithm: Theory, literature review, and applications. Applied Soft Computing, 105, 107329. https://doi.org/10.1016/j.asoc.2021.107329.

[15]. Dervis, K. (2010). An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University.

[16]. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1), 53–66. https://doi.org/10.1109/4235.585892.

[17]. Fister, I., Fister, D., Yang, X. S., & Brest, J. (2015). A comprehensive review of bat algorithm and its applications. Artificial Intelligence Review, 42, 895–919.

[18]. Ihsan, R. R., Almufti, S. M., Ormani, B. M. S., Asaad, R. R., & Marqas, R. B. (2021). A survey on cat swarm optimization algorithm. Asian Journal of Research in Computer Science, 10(2), 22–32. https://doi.org/10.9734/ajrcos/2021/v10i230237.

[19]. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x.

[20]. Marqas, R. B., Almufti, S. M., Ahmed, H. B., & Asaad, R. R. (2021). Grey wolf optimizer: Overview, modifications and applications. International Research Journal of Science, Technology, Education, and Management, 1(1), 44–56. https://doi.org/10.14419/efkvvd44.

[21]. Marqas, R. B., Almufti, S. M., Othman, P. S., & Abdulrahman, C. M. (2020). Evaluation of EHO, U-TACO and TS metaheuristics algorithms in solving TSP. Journal of Xi'an University of Architecture & Technology, 12(4), 3245–3246.

[22]. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[23]. Sahoo, G., & Tripathy, R. (2020). Comparative study of nature-inspired algorithms for TSP. International Journal of Computer Applications, 175(6), 1–6.

[24]. Shaban, A. A., & Ibrahim, I. M. (2025). Swarm intelligence algorithms: A survey of modifications and applications. *International Journal of Scientific World*.

[25]. Shaban, A. A., & Yasin, H. M. (2025). Applications of the artificial bee colony algorithm in medical imaging and diagnostics: A review. International Journal of Scientific World, 11(1), 21–29.

[26]. Shaban, A. A., Almufti, S. M., Asaad, R. R., & Marqas, R. B. (2025). Swarm-based optimisation strategies for structural engineering: A case study on welded beam design. *FMDB Transactions on Sustainable Computer Letters*, *3*(1), 1–11.

[27]. Shaban, A. A., Dela Fuente, J. A., Salih, M. S., & Ali, R. I. (2023). Review of swarm intelligence for solving symmetric traveling salesman problem. *Qubahan Academic Journal, 3*(2), 10–27. https://doi.org/10.48161/qaj.v3n2a141.

[28]. Wang, G. G., Deb, S., & Coelho, L. D. S. (2015). Elephant herding optimization. In Proceedings of the 2015 International Symposium on Computational Intelligence and Design (pp. 1–6). https://doi.org/10.1109/ISCID.2015.134.

[29]. Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) (pp. 65–74). Springer. https://doi.org/10.1007/978-3-642-12538-6_6.

[30]. Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 210–214). IEEE. https://doi.org/10.1109/NABIC.2009.5393690.

[31]. Zebari, A. Y., Almufti, S. M., & Abdulrahman, C. M. (2020). Bat algorithm (BA): Review, applications and modifications. International Journal of Scientific World, 8(1), 1–7. Science Publishing Corporation. https://doi.org/10.14419/ijsw.v8i1.30120.