Expert Systems

WILEY

Expert Systems

ORIGINAL ARTICLE OPEN ACCESS

Improving Safety and Efficiency of Industrial Vehicles by Bio-Inspired Algorithms

Eduardo Bayona[1,2] | J. Enrique Sierra-García[1] | Matilde Santos Peñas[3]

[1]Department of Digitalization, University of Burgos, Spain | [2]UBU-MICHELIN Joint Research Unit in Automation and Smart Industry, Arainnov Michelin Aranda, Spain | [3]Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain

Correspondence: Eduardo Bayona (ebayona@ubu.es)

Received: 3 September 2024 | **Revised:** 27 December 2024 | **Accepted:** 2 January 2025

Keywords: AGV | bio-inspired algorithms | industry 4.0 | optimization

ABSTRACT

In the context of industrial automation, optimising automated guided vehicle (AGV) trajectories is crucial for enhancing operational efficiency and safety. They must travel in crowded work areas and cross narrow corridors with strict safety and time requirements. Bio-inspired optimization algorithms have emerged as a promising approach to deal with complex optimization scenarios. Thus, this paper explores the ability of three novel bio-inspired algorithms: the Bat Algorithm (BA), the Whale Optimization Algorithm (WOA) and the Gazelle Optimization Algorithm (GOA); to optimise the AGV path planning in complex environments. To do it, a new optimization strategy is described: the AGV trajectory is based on clothoid curves and a specialised piece-wise fitness function which prioritises safety and efficiency is designed. Simulation experiments were conducted across different occupancy maps to evaluate the performance of each algorithm. WOA demonstrates faster optimization providing suitable safety solutions 4 times faster than GOA. Meanwhile, GOA gives solutions with better safety metrics but demands more computational time. The study highlights the potential of bio-inspired approaches for AGV trajectory optimisation and suggests avenues for future research, including hybrid algorithm development.

1 | Introduction

In the rapidly evolving landscape of industrial automation and logistics, the optimization of AGV trajectories represents a frontier of significant scientific and practical interest (Sierra-Garcia and Santos 2024). As the core of material handling in manufacturing, warehousing and port operations, AGVs play a significant part in improving operational efficiency and reducing human error (Zhang et al. 2023). However, the dynamic nature of their operating environments poses unique challenges for path planning (Oyekanlu et al. 2020).

AGVs are transforming the material transportation sector across multiple industries by increasing efficiency, reducing labour costs and increasing safety standards (Chaudhry et al. 2022). Their application affects operational efficiency significantly and can be found frequently from logistic hubs to factory lines (Sodiya et al. 2024). Nonetheless, there is some concern when it comes to path planning for AGVs in dynamic and complicated situations. The complex task of avoiding obstacles and guaranteeing effective transportation calls for a practical combination of strategies (Dao et al. 2024). AGVs are also relevant in the field of industrial logistics, where the smooth transportation of containers depends on the effective scheduling and route planning of these vehicles. Here, the emphasis on optimising AGV trajectories is important for both increasing efficiency and guaranteeing safety in the usually crowded dock area. (Chen et al. 2023).

The complexity of AGV path planning is further accentuated when considering the dynamic nature of their operating environments (Yang et al. 2023). Traditional algorithms, while foundational, often fall short in addressing the multifaceted objectives of AGV operations, such as energy conservation, minimising travel time and avoiding collisions (Li et al. 2023). This gap has paved the way for innovative approaches that meld traditional path-planning methods with cutting-edge technologies and methodologies (Wang, Luo, and Tan 2021).

Metaheuristic algorithms, inspired by natural phenomena, have emerged as a promising solution, offering the ability to handle high-dimensional and nonlinear problems with remarkable adaptability (Peñacoba et al. 2023). Bio-inspired algorithms are a subset of metaheuristic algorithms that take their inspiration from biological processes observed in nature, such as the social behaviour of swarms, the foraging strategies of animals and evolutionary dynamics. These bio-inspired approaches have been extensively used for their robustness and efficiency in solving complex optimization problems, particularly in dynamic and uncertain environments (Soria, Schiano, and Floreano 2022).

The choice of bio-inspired algorithms for this study is particularly relevant given that as prior research has shown, they are quite successful in examining complicated solution spaces (Rani et al. 2023). Bio-inspired algorithms are a good alternative for AGV trajectory planning by leveraging their inherent stochastic nature, which promotes a comprehensive exploration of the solution space (Liang et al. 2023). This variability in the search process enables these algorithms to effectively navigate the complex landscape of potential solutions, making them particularly well-suited for problems with multiple optimal paths (Zhang et al. 2024).

In the context of Industry 4.0, AGVs are essential for automating logistics operations, increasing process flexibility and reducing reliance on human intervention in manufacturing and smart warehousing. However, effective path planning for these industrial vehicles remains a challenging task, especially in complex, dynamic and obstacle-rich environments. AGVs must navigate these scenarios efficiently and safely, avoiding collisions, minimising travel time and meeting operational demands. Conventional path-planning methods often struggle to balance these goals, especially as environments become increasingly intricate.

This study addresses the need to address these diverse objectives in AGV path planning using advanced optimization techniques, in particular by exploring the potential of biologically inspired optimization algorithms. The Bat algorithm (BA), Whale Optimization Algorithm (WOA) and Gazelle Optimization Algorithm (GOA) offer adaptive and flexible search methods, that could effectively meet the complex demands of AGV navigation in real-world environments. In this research, these three metaheuristic algorithms are applied and compared within a clothoid-based AGV path-planning framework. The ultimate goal is to identify efficient and reliable solutions for these industrial vehicles that align with Industry 4.0 standards and challenges, thereby improving automation, safety and operational performance. The growing literature on these methods highlights their potential for continuous improvement and adaptation to increasingly complex problems (Yao and Deng 2023; Zellagui, Belbachir, and El-Sehiemy 2023; Chandrasekaran et al. 2022).

The main contribution of this paper is to demonstrate how biologically inspired optimization algorithms can effectively address the challenges of AGV trajectory optimization, combining diverse requirements. Various metrics are used to evaluate and compare those that respond to the demands of modern industrial environments where safety, efficiency and adaptability make the difference (Khamis, Hussein, and Elmogy 2015).

The three bio-inspired strategies, BA, WOA and GOA, have shown potential in various robotic path-planning tasks. BA is known for its effectiveness in navigating complex paths (Yu, Zhu, and Lv 2023); WOA, inspired by the cooperative hunting behaviour of humpback whales, is widely applied in path optimization for robotic systems (Vinaykumar, Babu, and Frnda 2023); and GOA, which emulates the agile and evasive movements of gazelles, offers an innovative approach adaptable to a variety of applications, such as lithium-ion battery optimization in smart grids (Hasanien et al. 2023). However, despite their potential, their application in clothoid-based AGV path planning within the context of Industry 4.0 has not been previously explored.

This study adapts these algorithms for AGV trajectory optimization and introduces a rigorous set of evaluation methods and metrics to analyse and contrast their effectiveness on complex occupational maps. By doing so, it provides new insights into how these algorithms perform in practical, real-world scenarios, offering guidance for selecting the most suitable algorithms for AGV systems in dynamic and complex environments typical of Industry 4.0.

This paper is organised as follows: Section 2 provides a comprehensive review of the literature on AGV path planning, highlighting the strengths and limitations of existing algorithms. Section 3 details the methodology, including the selection criteria for the bio-inspired algorithms and the development of the novel fitness function. Section 4 presents the experimental setup and evaluation criteria, while Section 5 discusses the results, comparing the performance of the algorithms. Finally, Section 6 concludes the paper with key findings and proposes directions for future research.

2 | Related Works

Bio-inspired algorithms have found extensive application in trajectory generation and path planning, offering innovative solutions to navigate through complex environments in various disciplines (Lu et al. 2023; Hason and Al-Darraji 2023), highlighting the emergence of new methods in the recent literature, including those studied in this paper (Gonçalves, Souza, and Fernandes 2022).

Previous studies have demonstrated the effectiveness of algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in trajectory planning for autonomous systems due to their similar computational frameworks (Ziadi, Njah, and Charfi 2020). However, both face limitations in complex multi-objective tasks, such as AGV navigation. PSO's rapid convergence can hinder exploration in highly constrained environments, while GA's crossover and mutation mechanisms, though robust, impose significant computational demands

(Chevallier and Clairmont 2024). Bayona et al. compared GA, PSO and pattern search for automated guided vehicles, revealing their computational intensity and the need for careful fitness function design to prevent premature convergence and ensure feasible solutions in highly constrained spaces (Bayona, Sierra-García, and Santos 2024). In contrast, emerging bio-inspired algorithms, such as those used in this study, offer unique advantages for Industry 4.0 applications. For instance, the WOA outperforms traditional methods in exploration and solution quality as shown previously (Mohajeri et al. 2022), while approaches like the BA stand out for their versatility and superior performance across multiple domains (Johnvictor et al. 2022).

Recent studies have explored various bio-inspired metaheuristic algorithms for optimising trajectory generation in AGVs and mobile robots. The authors (Bayona, Sierra-García, and Santos 2024) compared genetic algorithms, PSO and pattern search for AGV path planning, highlighting their relative efficiencies. Savsani et al. evaluated seven metaheuristics for robotic arm trajectory optimization (Savsani, Jhala, and Savsani 2016), finding teaching-learning-based optimization, artificial bee colonies and cuckoo search algorithms to be particularly effective. Miljković & Petrović reviewed and compared (Miljković and Petrović 2020) biologically inspired algorithms for mobile robot scheduling, like PSO, chaotic PSO and WOA.

The BA was developed by Yang (2010), leveraging the echolocation mechanism of bats has shown success in urban planning and civil engineering, particularly in the layout optimization of road networks and the design of pedestrian traffic (Srivastava and Sahana 2019). It has been also used in a previous study (Roberge, Tarbouchi, and Labonte 2013) where the BA is compared with other optimization techniques in the context of real-time UAV path planning. it demonstrated that the BA was effective in finding optimal paths for UAVs, showcasing its potential for trajectory optimization. Also, the BA has been used by the authors in the study of Perez et al. (2018) to optimise trajectories for a unicycle mobile robot, noting its rapid convergence compared to other metaheuristics.

The WOA, drawing inspiration from the foraging behaviour of humpback whales, was presented by Mirjalili & Lewis in a previous work (Mirjalili and Lewis 2016). It has been used by the authors in a previous work (Koyama et al. 2009) to conduct a comparison of different decoding algorithms in the context of brain-computer interface control. The results indicated that the Whale algorithm performed competitively, highlighting its potential for trajectory optimization tasks. This technique has also been used by Yan et al. (2022) and applied in autonomous vehicle navigation, improving the accuracy and efficiency of path planning in unpredictable terrains. Du et al., in a previous work (Du et al. 2023), introduced an inertia weight factor and variable neighbourhood search to enhance global and local optimization capabilities for manipulator trajectory planning and Loucif et al. also applied WOA to tune PID controllers for robot manipulator trajectory tracking in previous work (Loucif, Kechida, and Sebbagh 2020), demonstrating superior performance compared to other optimization methods.

The GOA, which draws inspiration from the movement patterns and running behaviour of gazelles, was introduced by Agushaka et al. in a previous work (Agushaka, Ezugwu, and Abualigah 2022) as a novel bio-inspired metaheuristic optimizer. It has been applied to optimise parameters in smart grid applications in a previous work (Y. Zhang et al. 2023) and it was enhanced in a study focusing on the optimization of microgrid operations, including power-to-gas equipment. This improved GOA showcases better global search accuracy and faster convergence by incorporating strategies like logistic mapping initialization and Gaussian mutation coefficient (Wu et al. 2024). A trajectory generation and optimization algorithm for quadrotor UAVs, utilising time-parametrized polynomials and addressing numerical stability issues has been proposed by Alqudsi et al. (Alqudsi, Kassem, and El-Bayoumi 2021) and Aadithya et al. introduced a decentralised online trajectory optimization algorithm for multi-robot systems, incorporating obstacle detection, local shape-based maps and inter-robot communication for collision avoidance (Aadithya et al. 2018).

These examples from various disciplines demonstrate the adaptability and potential of bio-inspired algorithms in optimising complex systems and demonstrating their robustness in various optimization challenges. Focusing on comparative studies that highlight their performance across various optimization challenges, recent research has examples like the comparison of three state-of-the-art algorithms by Morrel et al. (Morrell et al. 2018), finding that different approaches excel in various scenarios, such as slow conservative trajectories or high-speed flight in tight spaces. Salamat & Tonello proposed a stochastic trajectory generation method using PSO with quintic B-splines in a previous work (Salamat and Tonello 2017), which satisfies UAV configuration constraints and allows for obstacle avoidance and speed limitations. Also, the authors (Rehberg et al. 2023) evaluated four nonlinear trajectory optimization solvers for energy-optimal quadrotor motion planning, addressing the challenges of nonconvex constraints in complex and highly dynamic scenarios.

While existing comparative studies, like the study of Khalife et al. in a previous work (Khalife et al. 2022), there has been a lack of direct evaluation studies specifically focusing on the application of these algorithms to clothoid-based AGV trajectory optimization. This paper fills this gap with a comprehensive comparative study of bio-inspired algorithms using previous literature guidelines (Molina et al. 2020).

To further clarify the methodological distinctions between our proposed approach and existing methods, we provide a comparative summary in Table 1. This table highlights the primary differences in application contexts, trajectory modelling, optimization objectives and evaluation metrics across various studies.

3 | Bio-Inspired Techniques

This study evaluates the performance of three different biologically inspired algorithms for optimising AGV navigation in complex environments: BA, GOA and WOA. These methods were selected for the AGV path-planning problem based on their adaptability, convergence properties, efficiency and suitability for multi-objective optimization in constrained environments (Dalla Vedova, Berri, and Re 2019). These algorithms leverage

TABLE 1 | Comparative analysis of optimization methodologies.

Reference	Application	Problem modelling	Optimization objective & technique	Evaluation metrics
This study	AGV trajectory optimization for safety and efficiency in Industry 4.0 environments	Clothoid-based trajectory model with a specialised piece-wise fitness function	Minimise collision risk, ensure safety compliance and optimise trajectory length with Bat, Whale and Gazelle algorithms	Non-collision time, initial safe path time, minimum and average distances to obstacles, trajectory travel time
(Savsani, Jhala, and Savsani 2016)	Robotic arm trajectory optimization for a three-revolut robotic arm	Discrete trajectory model with multiple design variables for joint angles and velocities	Minimise joint travel time, travel distance and Cartesian lengths simultaneously with the Bat algorithm	Best solution, mean solution, standard deviation, computational effort, convergence and statistical significance
(Wu et al. 2024)	Microgrid operation optimization including power-to-gas and hybrid energy storage systems	System model with integrated P2G and energy storage components	Minimise operation cost and carbon emission penalty cost with the Gazelle algorithm	Carbon emission reduction, operation cost, comparative performance with single energy storage systems
(Morrell et al. 2018)	High-speed trajectory planning for quadrotors near obstacles	Dynamic feasibility model to ensure safe, trackable flight	Minimise collision risk and maximise tracking accuracy at high speeds with the ASTRO algorithm	Collision-free path rate, trajectory tracking accuracy
(Salamat and Tonello 2017)	Trajectory generation for UAVs in various flight scenarios	Quintic B-splines ensure smooth second-order derivatives	Multi-objective optimization to satisfy UAV constraints and minimise trajectory length with the PSO algorithm	Execution time, minimum-length trajectory effectiveness, obstacle avoidance compliance
(Rehberg et al. 2023)	Nonlinear trajectory optimization for quadrotor flight control	High-dimensional state space with nonlinear constraints	Minimise effort and improve tracking accuracy in complex environments with the KOMO algorithm	Solution quality, tracking error, computational efficiency
(Khalife et al. 2022)	Diesel engine combustion modelling for performance and emission estimation	Empirical model based on fuel type and engine load	Maximise thermal efficiency and minimise fuel consumption and emissions with Grey Wolf, Ant Lion and Grasshopper optimization Algorithms	Correlation coefficient, Mean Square Error, prediction accuracy for engine performance and emissions

distinct nature-inspired search mechanisms that effectively balance exploration and exploitation, which is important for finding AGV trajectories in complex industrial settings (Jakšić et al. 2023).

On the other hand, compared to other metaheuristic optimization algorithms, BA, WOA and GOA offer a novel approach that has not yet been explored for industrial autonomous vehicles and that aligns more closely with the demands of AGV path planning in Industry 4.0. BA's local search is suitable for efficiently exploring constrained trajectories; WOA's adaptive behaviours provide a balanced approach to exploring spaces with many obstacles and GOA's predator pursuit response offers effective safety management in highly constrained scenarios. Therefore, these three algorithms BA, WOA and GOA offer a representative sample of adaptive optimization techniques suitable for intelligent AGV path planning in industrial environments.

A maximum of 60 iterations, chosen empirically as a stopping criterion, was applied across the three algorithms to balance computational cost and solution quality. Preliminary testing indicated that each algorithm achieved substantial improvement within this range, with minimal gains observed from additional iterations. This choice reflects a balance between computational efficiency and solution quality, capturing the essential convergence benefits without unnecessary computational costs. Each algorithm is configured as follows.

3.1 | BA

BA is a nature-inspired optimization technique developed by Xin-She Yang in a previous work (Yang and Hossein Gandomi 2012), inspired by the echolocation behaviour of bats. It belongs to the swarm intelligence family and has shown effectiveness in various fields, including engineering, business and transportation (Okwu and Tartibu 2021).

In this study, the BA initialization phase starts by generating random output angles within predefined constraints, simulating the initial location of bats within the search space. The algorithm uses a population size of 50 individuals, empirically chosen to balance computational cost and solution quality, mirroring a colony of bats and harnesses the principle of echolocation. Bats use sound waves to navigate and locate prey, a mechanism replicated in the BA through varying frequencies and pulse emissions.

- *Frequency (f)*: Determines how often bats explore the search space, with a range [0,2], allowing them to adjust their movement dynamically based on their proximity to the optimal solution. This parameter enables a balance between global and local search capabilities, enabling bats to navigate efficiently towards promising areas of the search space.
- *Pulse Rate (r)*: Controls the emission rate of sound pulses. A higher pulse rate encourages exploration, while a lower rate focuses on exploitation, guiding bats towards promising regions in the search space.
- *Loudness (A)*: Used as a measure of the bat's eagerness to explore. It decreases as the bat moves closer to its prey, simulating the decreasing loudness of echolocation sounds as bats approach their target. This gradual reduction in loudness ensures that bats focus their search efforts on areas around the discovered optimal solutions, enhancing the exploitation phase of the algorithm.

BA achieves a balance between exploration and exploitation through dynamically adjusted parameters, specifically frequency f and pulse rate r, which drives its convergence behaviour.

During each iteration, the positions and velocities of the bats are updated based on their current positions, the best-known solutions and randomization factors influenced by frequency and loudness. Bats with better positions (solutions) emit fewer pulses and lower loudness. In each iteration, the position x_{i+1} of a bat (or agent) is updated based on its current position x_i and velocity v_i:

$$x_{i+1} = x_i + v_i \tag{1}$$

where x_{i+1} is the updated position of the bat and v_i is the velocity, allowing each bat to move towards promising solutions. The velocity v_i is influenced by the frequency f, helping BA adjust its search radius around the best-known solution:

$$v_i = v_{i-1} + f_i \cdot \left(x_i - x_{best}\right) \tag{2}$$

Here, x_{best} represents the current best solution and the frequency f, which controls the search radius around the best solution, is bounded within Q_{min} and Q_{max}. This velocity adjustment allows bats to exploit local areas based on the best solution found so far. By adjusting f, BA adapts its search range, enabling both wide exploration and focused local search.

The pulse rate $r_i = r_0\left(1 - e^{-\gamma i}\right)$ increases over time, making exploration less likely as iterations progress. Concurrently, loudness $A(t)$ decays exponentially, following $A_{i+1} = \alpha \cdot A_i$, where $\alpha \in (0,1)$ is a decay factor. This dual adjustment in pulse rate and loudness is designed to ensure that BA increasingly focuses on the exploitation of the best-known solutions. As t increases, $r_i \approx r_0$ (the maximum pulse rate), while $A_i \approx 0$, effectively reducing random exploration and narrowing the search radius around promising solutions.

To ensure convergence, $r(t)$ and $A(t)$ adapt dynamically: $r(t)$ increases, encouraging local search around optimal areas, while $A(t)$ decreases, restricting the bats' movement range and refining their focus near high-quality solutions. Although a formal convergence proof for BA is complex, literature suggests that such gradual parameter decay adjustments in pulse rate and loudness facilitates convergence by balancing exploration and exploitation throughout the algorithm's iterations.

BA has been successfully applied to solve complex problems such as manufacturing cell design, where it efficiently distributes machines in production centres to minimise travel during the manufacturing process (Soto et al. 2016). Recently, BA has been implemented in swarm robotics, with both physical prototypes and computational simulations developed to replicate microbat behaviour for tasks like finding target locations in unknown indoor environments (Suárez, Iglesias, and Gálvez 2019). The value of the parameters used in this study are summarised in Table 2.

3.2 | WOA

WOA is a nature-inspired metaheuristic optimization algorithm that mimics the bubble-net hunting behaviour of humpback whales, where they create spiral-shaped bubbles to encircle and capture prey (Okwu and Tartibu 2021). WOA achieves a balance between exploration and exploitation through adaptive encircling and spiral updating mechanisms, which guide the algorithm's convergence.

- *Encircling Prey*: Whales (solutions) encircle the prey (optimal solution) by moving towards the best-known position in the search space. This behaviour ensures that the population gradually converges towards the optimal solution, similar to how whales encircle their prey.
- *Spiral Updating Position*: Whales follow a spiral path as they close in on their prey, which is mathematically represented by a spiral equation. This spiral motion introduces a stochastic component, enhancing the diversity of the search and preventing premature convergence.

In each iteration, WOA updates the positions of the whale agents relative to the best-known solution x_{best}, employing a shrinking encircling mechanism and a spiral updating equation. These mechanisms allow WOA to adaptively focus on promising solutions, narrowing the search space as iterations progress. The position x_{i+1} of a whale is updated based on x_{best} and the current distance D between the whale and the best solution, using the following spiral updating formula:

$$x_{i+1} = x_{best} + D \cdot e^{bl} \cdot \cos(2\pi l) \tag{3}$$

Here, $D = | x_{best} - x_i |$ is the distance between the whale's current position x_i and the best solution x_{best}, while b is a constant controlling the logarithmic spiral's shape and l is a random number in the range $[-1, 1]$. This formula creates a shrinking spiral pattern around x_{best}, allowing WOA to perform local searches near high-quality solutions while still exploring the space adaptively.

As iterations progress, the distance D between each whale and x_{best} shrinks, reducing the search radius around optimal areas and promoting convergence. The gradual reduction of D effectively narrows the search space, ensuring that WOA increasingly concentrates around x_{best}. This adaptive spiral updating mechanism helps WOA transition from exploration to exploitation as iterations increase, enhancing its ability to converge towards optimal solutions. To maintain diversity and prevent the algorithm from getting stuck in local optima, WOA includes a random selection of solutions from the population. This process injects variability into the population, ensuring a comprehensive exploration of the search space.

In this study, WOA uses a population of 50 whales. The random selection combined with the shrinking encircling mechanism allows WOA to efficiently search both globally and locally, making it particularly effective for complex, multi-modal optimization problems.

With the trajectory approach of this paper, WOA optimises the output angles of the intermediate points as tuning variables, with initial output angles randomly determined within predefined constraints. Additionally, the algorithm selects individuals for the next generation using a mechanism that considers both the current best solution and random selection among the existing solutions. This method promotes a diverse population, preventing premature convergence and encouraging thorough exploration of the search space. Parameters and their role in the algorithm's operation are summarised in Table 3.

WOA has shown competitive performance compared to other state-of-the-art algorithms in solving various optimization problems. WOA's strengths lie in its convergence speed and ability to balance exploration and exploitation (Mohammed, Umar, and Rashid 2019).

3.3 | GOA

GOA draws inspiration from the social behaviour and evasive movements of gazelles when fleeing predators. It models the swift, unpredictable movements of gazelles as they evade predators, translating these behaviours into mathematical functions that guide the search process. Each gazelle in the population represents a potential solution and their movements are influenced by a combination of social hierarchy, random exploration and gradient following (Abualigah, Diabat, and Zitar 2022). GOA achieves convergence by balancing exploration and exploitation through adjustments in step size, perception range and a Predator Search Response (PSR) mechanism, which enables the algorithm to adaptively shift its focus as needed.

TABLE 2 | BA parameters.

Parameter	Value
Bats population size	50
Number of iterations	60
Search space boundaries	[0, 360]
Frequency (f)	$[Q_{min}, Q_{max}] = [0, 2]$
Loudness (A)	0.95
Pulse rate (r)	0.1

TABLE 3 | Whale optimization algorithm parameters.

Parameter	Value
Whale population size	50
Number on iterations	60
Search space boundaries	[0, 360]

- *Perception Range*: The gazelles' perception range controls the scope of their exploration, allowing them to assess their surroundings and decide on their next move. This is analogous to the crossover operation in genetic algorithms, but GOA's approach is more dynamic, adjusting the search direction based on the perceived landscape.
- *PSR*: PSR is a key mechanism in GOA that modulates the gazelles' behaviour, increasing the probability of moving towards high-quality solutions as iterations progress. This probability, p_{PSR}, allows GOA to gradually shift from broad exploration to focused exploitation, adapting to promising solutions over time.

GOA's convergence is achieved by modulating key parameters that guide the balance between broad search and focused exploitation. Specifically, GOA dynamically adjusts the perception range, incorporates random exploration and gradient following and uses a PSR mechanism to influence gazelles' likelihood of moving towards high-quality solutions. These mechanisms collectively manage the convergence rate, ensuring that GOA can adapt its search focus over time.

The perception range P_i of each gazelle initially allows for broad exploration, enabling each agent to assess its surroundings and select promising directions. This perception range is analogous to the crossover operation in genetic algorithms, but GOA's approach is more dynamic, reducing the perception range over time. Mathematically, this is represented by:

$$P_{i+1} = P_i \cdot \beta \tag{4}$$

where $\beta \in (0, 1)$ is a decay factor. As P_i decreases, gazelles concentrate their search around local regions, which accelerates convergence by narrowing the search radius and emphasising local refinement near high-quality solutions.

GOA further balances exploration and exploitation through a combination of random movements and gradient-based adjustments, which help each gazelle explore new areas of the search space while moving towards promising gradients. Additionally, GOA includes a mutation mechanism that simulates sudden, unpredictable directional changes seen in gazelles when evading predators. This randomness adds diversity to the search, reducing the risk of premature convergence. At the same time, an elite selection strategy ensures that the most successful gazelles guide the population, turning the search towards more optimal solutions.

A key component in GOA's adaptive strategy is the PSR mechanism, which adjusts the gazelles' movement based on external signals. The probability p_{PSR} of moving towards the best-known solution increases over time, helping GOA transition from exploration to exploitation. The probability is defined by:

$$p_{\mathrm{PSR}} = 1 - e^{-\delta i} \tag{5}$$

where δ controls the rate of increase in p_{PSR} and i is the iteration count. As iterations progress, p_{PSR} approaches 1, directing the search towards high-quality solutions and minimising unnecessary exploration.

The primary position update for each gazelle incorporates an adaptive step size c_i and an exponential decay factor, gradually decreasing the search radius as iterations progress. In each iteration, the position x_{i+1} of a gazelle is updated based on:

$$x_{i+1} = x_i + c_i \cdot d \cdot e^{(-\alpha i)} \tag{6}$$

where c_i is the initial step size, d is a direction vector, α controls the decay rate and i is the iteration count. This formulation ensures that as i increases, $c_i \cdot e^{(-\alpha i)}$ approaches zero, narrowing the search area around optimal solutions and supporting steady convergence by refining the search radius.

By combining these mechanisms, GOA effectively balances exploration and exploitation, which facilitates a controlled convergence rate. The early iterations allow for broad exploration, while the decay in perception range, step size and the increase in p_{PSR} gradually narrow the search focus, supporting local refinement around promising solutions.

In this study, the GOA is configured with a population of 50 gazelles, chosen by previous parameter tuning. GOA tunes the output angles of the intermediate points, which serve as parameters in the optimization process. The angles are initially generated randomly within the bounds defined by the task's constraints, setting the stage for the algorithm's operation. All these parameters and their role in the algorithm's operation are summarised in Table 4.

This algorithm is particularly effective in dynamic and highly constrained environments as it has shown promising performance on benchmark functions and engineering problems, demonstrating its potential as an effective optimization tool (Abdollahzadeh et al. 2022). Recent studies have proposed enhancements to improve its performance. The authors of a previous work (Qin et al. 2024) introduced a multi-strategy approach combining PSO with GOA, demonstrating superior convergence accuracy and stability across benchmark functions and engineering problems. Ekinci et al. developed in a previous work (Ekinci, Izci, and Hussien 2024) a hybrid gazelle-Nelder–Mead algorithm, which outperformed other optimization approaches in parameter extraction for solar photovoltaic systems. These studies show the versatility and effectiveness of GOA-based algorithms in solving complex optimization challenges across diverse domains.

3.4 | Algorithms Common Parameter Analysis

A comparative analysis of the three bio-inspired algorithms employed in this study reveals that, despite their inherent differences in terms of characteristics and mechanisms, they exhibit a number of common fundamental parameters that play analogous roles.

- The exploration and exploitation parameters of each algorithm facilitate a balance between exploration and exploitation within the search space. The BA incorporates frequency (f) and pulse rate (r) as modulating factors in the exploration and exploitation of the search space. WOA employs a shrinking encircling mechanism and random selection to adjust the search intensity over iterations. GOA uses the perception range and step size coefficient to dynamically adjust the search scope and direction of gazelles.

- The three algorithms feature randomization mechanisms, which introduce stochasticity to maintain diversity within the population and prevent premature convergence. The aforementioned mechanisms, whether it be the pulse rate in BA, the spiral updating position in WOA, or the mutation in GOA, ensure that the search space is sufficiently explored, thereby reducing the risk of becoming trapped in a local optimum.
- The selection of the optimal solutions is a fundamental aspect of all three algorithms. They incorporate a mechanism to prioritise the best solutions that have been identified within the search space. In BA, the positions of bats are updated based on the best solution and their current velocity. WOA involves encircling behaviour to move towards the best solution. GOA employs elite selection to propagate the best solutions to the next generation.

TABLE 4 | GOA parameters.

Parameter	Value
Gazelle population size	50
Number on iterations	60
Search space boundaries	[0, 360]
Probabilistic selection rate	0.34
Step size coefficient	0.88

The similarities in the structure and operation of these algorithms, despite their distinct inspirations and specific implementations, are indicative of underlying commonalities. The acknowledgement of the parallels between these parameters enables a more detailed comprehension of the ways in which these algorithms can be successfully deployed in analogous scenarios.

4 | Optimization Methodology

This research presents a comparative analysis of three bio-inspired algorithms that aim to optimise the trajectory planning of AGVs in complex operational environments. We evaluate the algorithms in different simulation environments. Firstly, a detailed discussion of the evaluation methodology, the approach for optimization and the characterisation of key elements within the study are provided. The methodological framework is illustrated in Figure 1.

The problem definition is the first step in identifying the AGV navigation challenge, which is the basis for the optimisation

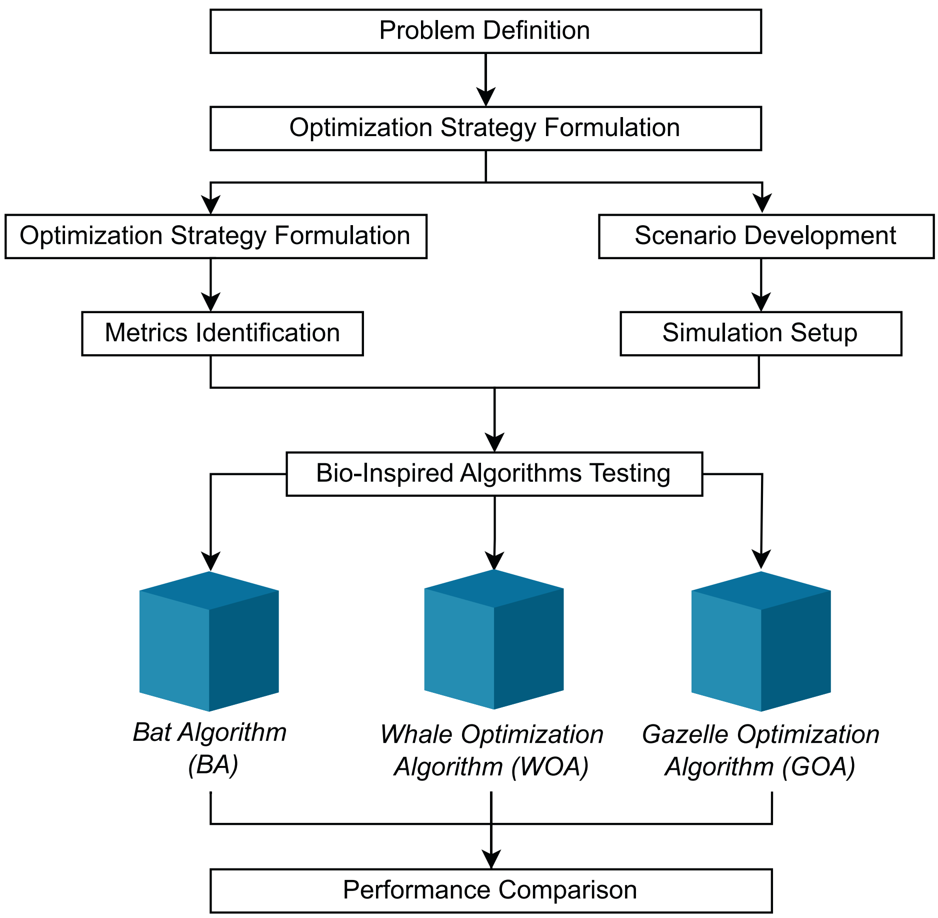

FIGURE 1 | Optimization methodology.

process. Two parallel aspects are then developed: the optimisation strategy formulation designed to be independent yet adaptable to each heuristic method and the scenario development. Within the optimisation strategy formulation, key performance metrics are identified to evaluate the performance of each algorithm. At the same time, in the scenario development phase, different occupancy maps are designed to test the algorithms under different conditions. These scenarios provide a structured environment in which the performance of the algorithms can be evaluated. Both the identified metrics and the developed scenarios lead to the algorithmic testing phase. Here the specific configurations of the bio-inspired algorithms are outlined, setting the stage for experimental testing. Each algorithm is then subjected to the developed scenarios and its performance is carefully recorded.

In the next stage, the results of the algorithmic testing are analysed and compared in the performance evaluation phase. This analysis highlights the strengths and weaknesses of each algorithm. Indicators such as time to reach a safe trajectory and efficiency in determining optimal paths are measured, providing a comprehensive assessment of the algorithms' capabilities.

To ensure a comprehensive evaluation of the different methods, it is essential to establish a clear mathematical framework for modelling the trajectories generated throughout the process. For the sake of clarity, the problem modelling is elaborated in Section 4.1 and the proposed optimization strategy is outlined in Section 4.2. This strategy remains independent of the chosen optimization technique. In addition, it is necessary to define the scenarios in which the trajectories will be compared, as described in Section 4.4. To provide a solid evaluation, a set of meaningful metrics is defined and recorded at each iteration of the optimization methods. These metrics, that are explained in Section 4.5, include the time to reach a collision-free solution and the time to reach the optimal solution. Section 4.3 discusses the fitness function used in the optimization methods, being an important part of the optimal solution-searching process. Once the overall framework is established, the metaheuristic techniques are configured as described in Section 3 and then tested according to the process described in Section 4.6. After each run, a log file containing the evaluation metrics is generated. This information is then used to compare the techniques and the results are discussed in Section 5.

4.1 | Problem Modelling

Figure 2 presents the approach for creating trajectories that are customised to meet the specific requirements of logistics applications. A computer-aided design (CAD) layout, details the dimensions of the work area, obstacles, start and end points and their corresponding angles. In order to achieve accurate docking at stations, it is important to accurately specify these angles. In their previous work (Bayona, Sierra-García, and Santos 2024), the authors presented an automated technique for processing the occupancy map to identify obstacles. The identified obstacles are delineated by polylines marking their perimeters. This technique simplifies the process of measuring the distance between the vehicle and the obstacles, thereby increasing the efficiency of the optimisation algorithm's fitness function computation.

The optimization tool's functionality is aided by users specifying a series of waypoints. Adjusting the angles at these waypoints is, therefore, the main focus of the optimisation routine. The system is provided with the obstacle list, the coordinates of the start and end points, including their angles and the user-defined waypoints. Iterative processes are used to determine the fastest trajectory, avoiding collisions and maintaining a safe distance from obstacles. These trajectories can then be deployed to AGVs or managed via a fleet control system.

Clothoid curves have several applications in path planning and generation for autonomous vehicles or robots. They have been selected in this study due to their unique ability to ensure continuous curvature transitions. This feature is essential for AGVs, which operate in restricted environments, where smooth changes in direction help prevent abrupt movements that could lead to instability when carrying loads. Other path-planning approaches, such as B-splines, were considered as they can also provide smooth paths, but clothoid curves inherently provide that G^2 continuity (continuous curvature), making them ideal for applications that need smooth and gradual curvature changes. Achieving similar behaviour with B-splines would require additional calculations to adjust and maintain curvature at control points, which can complicate real-time adjustments in dynamic environments.

In this study, clothoid curves are used for trajectory planning for autonomous vehicles and modelled using the Frenet formula within an occupancy map. An algorithm inspired by Bertolazzi and Frego (Bertolazzi and Frego 2015) solves the G^1 Hermite interpolation problem to generate a clothoid curve connecting two points with specific tangent vectors on a plane. Solutions for this optimization problem are defined as a set of elements: $\theta(N_w)$, where N_w is the number of intermediate points and θ is the output angle of the trajectory at those intermediate points. Therefore, the solution will have as many elements as intermediate points. The angle range is between 0 and 2π. These conditions are formally expressed in (7).

$$\theta_i \in [0, 2\pi]; \quad i \in \{\mathbb{N} \leq N_w\} \tag{7}$$

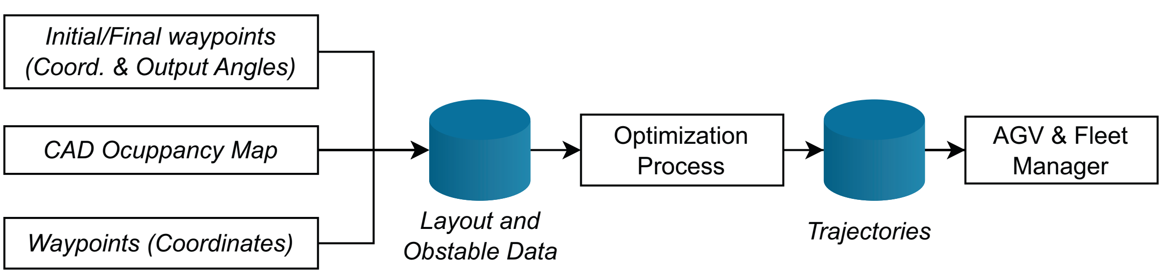

FIGURE 2 | Trajectory generation process.

The distance between obstacles and the vehicle is calculated using the vehicle's vertices and the segments representing the map's obstacles. While the AGV follows its trajectory, the vertices' positions in the inertial frame S_0 change, but their positions in the vehicle's coordinate frame S_R remain constant. The vertices in S_R are represented as $V_R = (W/2, L/2), (-W/2, L/2), (W/2, -L/2), (-W/2, -L/2)$, with W and L being the vehicle's width and length, respectively, as shown in Figure 3.

4.2 | Optimization Strategy

To enhance the AGV's trajectory, we propose a dynamic iterative computational approach, as shown in Figure 4. This method applies metaheuristic bio-inspired methods to optimise the output angles at intermediate waypoints θ_i. These angles, θ_i, along with the coordinates of the waypoints (x_i, y_i) and the starting and ending points' coordinates and angles (x_s, y_s, θ_s) and (x_e, y_e, θ_e), are the main factors in constructing the AGV's path, as described in Section 4.1. The trajectory is tested against a predefined list of obstacles to measure proximity to any obstacle D and perform collision checks D_c, as shown in a previous work (Bayona, Sierra-García, and Santos 2024). The results of these tests are used to evaluate the fitness function f_c, which assesses the performance of the trajectory. The metaheuristic algorithm then adjusts the angles θ_i based on the fitness value, refining the trajectory at each iteration for optimal performance.

The main parameters of this optimization problem are summarised in Table 5.

This optimization strategy can be synthesised and formalised with the Algorithm 1.

n is the number of points of the trajectory, N_o is the number of obstacles in the occupancy map, N_w is the number of intermediate points, D_{ijk} represents the distance from the vehicle's vertex k in the trajectory point i to the obstacle j when there is no collision and$D_{c_{ijk}}$ is the distance when a collision occurs. Dc is the sum of distances to all obstacles in a collision. $list_{obs}$ is the set of the obstacles, f_{opt} denotes the function executed by the metaheuristic algorithm, $t_{frenet} \in \mathbb{R}^{3n}$ denotes the Frenet trajectory, get_{dis} denotes the function to calculate the distances to the obstacles and get_{col} indicates the function which computes the collision distances.

4.3 | Fitness Function

Efficient and safe trajectory planning for AGVs requires a well-designed fitness function to guide the optimization process. This study presents a current fitness function structured as a piece-wise function with three distinct components, designed for optimising trajectories for AGVs. The approach is inspired by the methodology detailed in previous work (Bayona et al. 2024), with a particular focus on the Piece-Wise Average (PWAVG) method, which emphasises balancing safety and navigational efficiency.

The fitness function used in this optimization process, defined in Equation (8), is made up of three components, each of which addresses different aspects of the AGV path optimization. These components ensure a balance between safety and efficiency:

$$f_c = \begin{cases} 1 + \sum_{i=1}^{n} \min_{j \in N_o, k \in 1..4} D_{c_{ijk}} & \text{if } D_c \neq 0 \\ 0.5 + \dfrac{\left(SD - \min\left(\min_{j \in N_o, k \in 1..4} D_{ijk}\right)\right)}{2 \cdot SD} & \text{if } D_c = 0 \wedge \min\left(\min_{j \in N_o, k \in 1..4} D_{ijk}\right) \leq SD \\ \dfrac{TTT}{C_t} & \text{if} \min\left(\min_{j \in N_o, k \in 1..4} D_{ijk}\right) > SD \end{cases} \tag{8}$$

where n is the number of points of the trajectory, N_o is the number of obstacles in the occupancy map, D_{ijk} represents the distance from the vehicle's vertex k in the trajectory point i to the obstacle j when there is no collision and $D_{c_{ijk}}$ is that distance when a collision occurs. Dc is the sum of distances to all obstacles when there is a collision. Thus, when there is no collision ($Dc = 0$), the fitness function aims to maximise the distance to obstacles. On the other hand, when a collision occurs ($D_c \neq 0$), the fitness function is designed to minimise the length of the trajectory inside the obstacles. Distance calculations are based

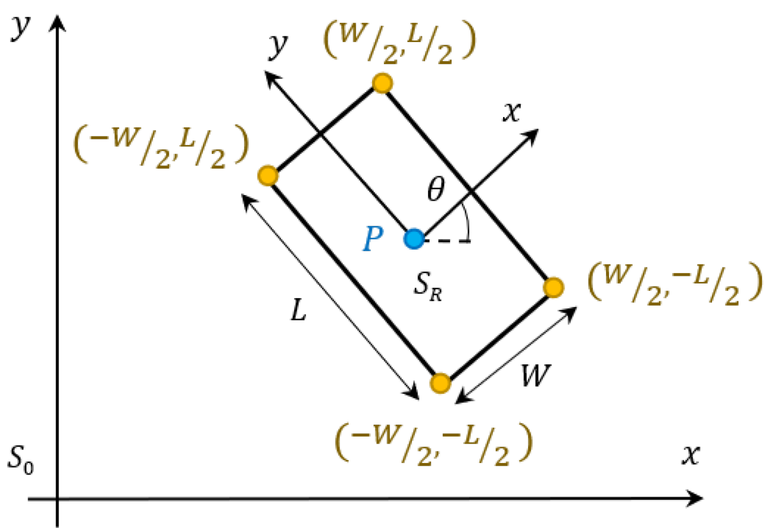

FIGURE 3 | Vehicle singular points representation in general and relative reference frames (Bayona, Sierra-García, and Santos 2024).

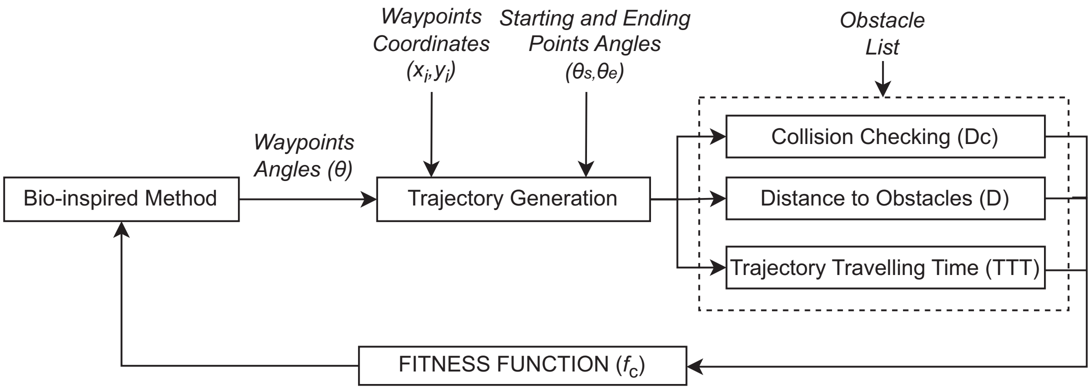

FIGURE 4 | Optimization strategy.

TABLE 5 | Summary of constraints, fixed parameters and optimization variables.

Constraints and fixed parameters	
Coordinates of the start point	(x_s, y_s)
Output angle of the start point	θ_s
Coordinates of the endpoint	(x_e, y_e)
Input angle of the endpoint	θ_e
Coordinates of the waypoints	$(x_i, y_i) \in \mathbb{R}^2,\ \ i \in \{\mathbb{N} \leq N_w\}$
Physical dimensions of the AGV	—
Optimization variables	
Output angles of the waypoints	$\theta_i \in [0, 2\pi],\ \ i \in \{\mathbb{N} \leq N_w\}$

on the mathematical description in a previous work (Bayona, Sierra-García, and Santos 2024). SD is the safety distance term, a constant term defined by the user to adjust the obstacle clearance. This proximity safety term is essential for preventing near-miss scenarios and ensuring that trajectories maintain an appropriate gap around obstacles.

Finally, TTT refers to the Trajectory Travelling Time metric described in 4.5 and C_t is a normalisation coefficient for continuity purposes in the piece-wise fitness function. The coefficient C_t is adjusted to ensure that the value of the third row of the fitness function is less than 0.5, thereby maintaining continuity across the transitions between different segments of the function.

- *Collision Penalty Component*:

$$1 + \sum_{i=1}^{n} \min_{j \in N_o, k \in 1.4} D_{c_{ijk}} \tag{9}$$

The collision penalty component described in Equation (9) applies when the AGV's path intersects with obstacles ($D_c \neq 0$). It penalises the trajectory based on the total invaded distance within the collision space, quantified by $D_{c_{ijk}}$, which measures the minimum distance between the AGV's vertices and obstacles during collision. This ensures that paths with significant overlap in collision zones are deprioritized, enhancing operational safety.

- *Proximity Safety Component*:

$$0.5 + \frac{\left(SD - \min\left(\min_{j \in N_o, k \in 1..4} D_{ijk}\right)\right)}{2 \cdot SD} \tag{10}$$

When there are no collisions ($D_c = 0$) but the trajectory does not comply with the minimum safety distance (SD), this term is shown in Equation (10) penalises proportionally paths that come too close to obstacles, seeking to maintain a minimum safety zone. The constant $SD = 100$ (Safe Distance) enforces a minimum clearance, which can be adjusted as needed. The initial safe path (ISP) represents the first collision-free route identified by the algorithm that maintains the required safety clearance (SD) throughout the entire trajectory. When this happens the condition to use the third row of the fitness function is triggered.

- *Efficiency Component*:

$$\frac{TTT}{C_t} \tag{11}$$

This component is activated when the path keeps a distance greater than the safety threshold $\left(\min\left(\min_{j \in N_o, k \in 1..4} D_{ijk}\right) > SD\right)$. It minimises the Total Travel Time (TTT) metric shown in Section 4.5, promoting efficient trajectories. The normalisation constant C_t ensures the fitness function remains continuous across transitions.

The balance between exploration and exploitation is a central aspect of the search strategy of the selected algorithms and the fitness function components are designed to support this balance in the context of AGV path planning. The proximity safety component encourages exploration by guiding the search process towards routes that maintain safe distances to obstacles, allowing the algorithms to explore multiple feasible paths without prematurely committing to suboptimal routes. The collision penalty component reinforces exploitation by penalising routes that intersect with obstacles, focusing the search on refining safe, collision-free routes. Finally, the efficiency component balances these elements by minimising the travel time, prompting the algorithm to prioritise routes that are both safe and operationally efficient.

ALGORITHM 1 | Optimization algorithm.

Require: $N_w \geq 0, n > 0, |list_{obs}| = N_o \geq 0, W > 0, L > 0$
$f_c \leftarrow \infty$
while $iter \leq iter_{MAX}$ **do**
$\{\theta_i\} \leftarrow f_{opt}(f_c), \ \ i \in \{\mathbb{N} \leq N_w\}$
$t_{Clothoid} \leftarrow f_{clothoid}(\theta_i, (x_e, y_e), (x_s, y_s), \{(x_i, y_i)\}), \ \ i \in \{\mathbb{N} \leq N_w\}$
$\{D_{ijk}\} \leftarrow get_{dis}(t_{clothoid}, list_{obs}, W, L), \ \ i \in \{\mathbb{N} \leq n\}, j \in \{\mathbb{N} \leq N_o\}, k \in \{\mathbb{N} \leq 4\}$
$\left\{D_{c_{ijk}}\right\} \leftarrow get_{col}(t_{clothoid}, list_{obs}, W, L), \ \ i \in \{\mathbb{N} \leq n\}, j \in \{\mathbb{N} \leq N_o\}, k \in \{\mathbb{N} \leq 4\}$
$D_c \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{N_o} \sum_{k=1}^{4} D_{c_{ijk}}$
if $D_c \neq 0$ **then**
$f_c \leftarrow 1 + \sum_{i=1}^{n} min_{j \in N_o, k \in 1..4} D_{c_{ijk}}$
else
if $D_c = 0 \wedge \min(\min_{j \in N_o, k \in 1..4} D_{ijk}) \leq SD$ **then**
$f_c \leftarrow 0.5 + \frac{(SD - \min(\min_{j \in N_o, k \in 1..4} D_{ijk}))}{2 \cdot SD}$
else
$f_c \leftarrow \frac{TTT}{C_t}$
end if
end if
end while

The combination of components ensures that the optimization process maintains a dynamic balance between exploring potential routes and exploiting good-quality solutions. To ensure that each component of the cost function contributes proportionally, min-max normalisation is applied to scale each metric into an equal range, maintaining a balanced impact on safety, collision avoidance and operational efficiency objectives. This normalisation technique provides smooth transitions between components and maintains stable behaviour across different scenarios, ensuring a consistent evaluation framework that accurately reflects the trade-offs between collision avoidance, safe distances and efficiency.

4.4 | Evaluation Scenarios

Two simulation environments have been designed to evaluate the performance of cutting-edge optimization algorithms. Each scenario covers dimensions of 13,500 by 11,000 mm and is differentiated by its complexity level, which is determined by the density and strategic placement of obstacles within the environment. This presents varied challenges for trajectory planning. Figure 5 shows the revised scenarios, including the start and end points of each trajectory and the associated output angles. The advanced challenge scenario has more obstacles than the initial basic challenge design, requiring a differentiated approach to navigate effectively.

Regarding design philosophy, the environments are created to simulate various challenges that one may face during real-world navigation. This provides a comprehensive platform to evaluate the adaptability and efficiency of the algorithms. Figure 5 displays the visualisation of these environments, with blue polygons representing obstacles and white spaces indicating navigable paths. The use of start and end points, accompanied by directional arrows, provides a clear representation of the test conditions. This enables a thorough analysis of the performance of each algorithm under varying levels of complexity.

This study presents two novel metrics, the Navigability Ratio (NR) and the Critical Point Density (CPD), to assess the complexity of occupancy maps. These metrics serve as important criteria for assessing the complexity of the designed simulation environments, improving the understanding of the challenging nature of different levels of complexity and their impact on the performance of the selected optimisation algorithms.

The NR is determined by calculating the percentage of the total area within the simulation environment that remains free from obstacles, allowing AGVs to navigate. This metric is important for understanding the available space for trajectory planning. A lower ratio indicates a more complex environment with limited navigable paths, while a higher ratio suggests a less crowded environment with more straightforward navigation. The ratio is calculated by dividing the total environmental area by the difference between the total environmental area and the area occupied by obstacles as shown in Equation (12). A higher percentage indicates a more navigable environment with fewer obstacles, simplifying trajectory planning. Conversely, a lower percentage signifies a highly complex environment with limited navigable paths, posing significant navigation challenges.

$$NR(\%) = 100\left(\frac{A_{total} - A_{Obs}}{A_{total}}\right) \tag{12}$$

where A_{total} represents the total area of the occupancy map and A_{Obs} is the area occupied by obstacles.

The CPD metric identifies areas within the environment that present significant navigational challenges, despite being

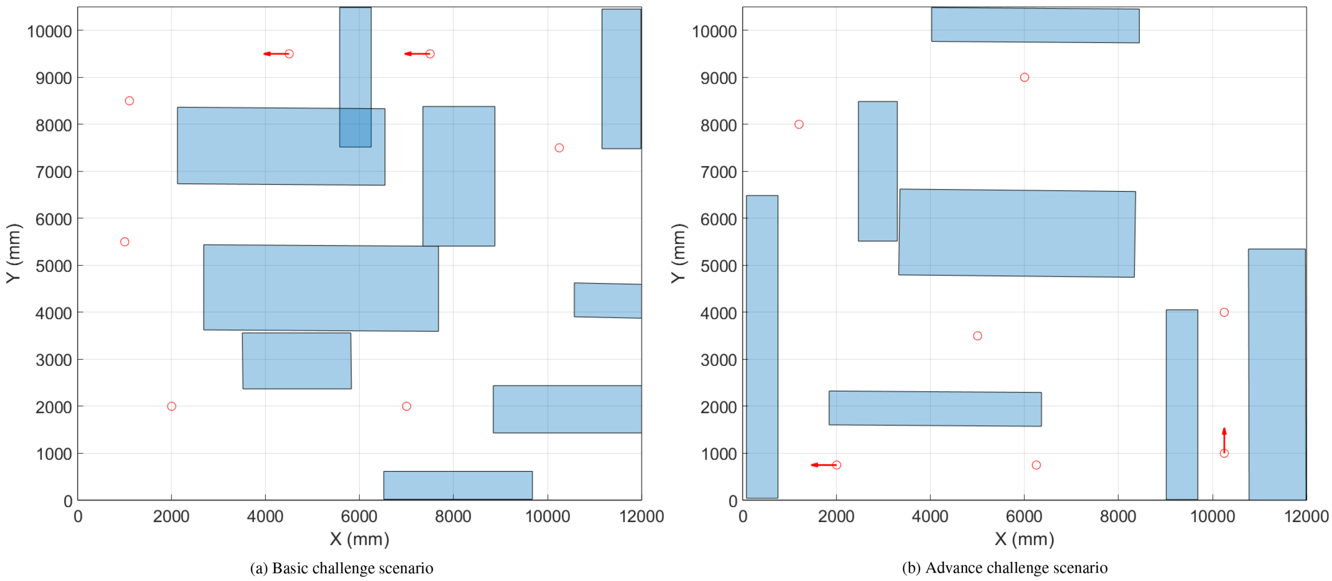

FIGURE 5 | Graphical representation of the considered scenarios for optimization experiments.

technically navigable. Critical points are defined as areas with a space greater than 1.1 times the width of the AGV but less than 3 times this width, representing narrow passages that require precise manoeuvring. The metric is calculated by identifying critical points or areas using spatial analysis techniques, then counting their occurrence and normalising this count by the total area of the map. This density value provides insight into the concentration of navigational challenges. It highlights environments that are mostly navigable but contain numerous sections that require advanced pathfinding solutions and adaptability from optimization algorithms.

$$CPD = \frac{NCP}{A_{total}} \tag{13}$$

where NCP represents the number of critical points in the map.

By incorporating these metrics into our evaluation framework, we obtain a comprehensive understanding of environmental complexity. Figure 5 illustrates not only the physical layout of obstacles and navigable paths but also provides a framework for the application of these quantitative measures. The use of the NR and the CPD metrics as criteria for complexity assessment ensures a comprehensive analysis of each simulation environment. The study aims to identify the most suitable algorithms for different levels of environmental complexity.

Two different occupancy maps are used to evaluate the effectiveness of the chosen optimisation algorithms in the generation of trajectories for AGVs. The NR and the CPD metrics are used to quantify and compare the complexity of these scenarios. They are designed to represent different levels of navigational complexity within the environment.

- Basic challenge scenario

$$NR(\%) = 74.13$$

$$CPD = 0.007$$

- Advance challenge scenario

$$NR(\%) = 76.04$$

$$CPD = 0.03$$

The Basic Challenge Scenario is defined by a smaller number of obstacles distributed across the map. This simplified design aims to evaluate algorithms' ability to generate efficient trajectories in a less complex environment. The NR metric is expected to be high in this scenario, indicating a broad availability of obstacle-free space for AGV navigation due to the lower obstacle density present in the map. The CPD metric is expected to be low, indicating a reduced number of narrow passages that require precise navigation manoeuvres. This is due to the strategic and limited placement of obstacles.

In contrast, the Advance Challenge Scenario is a more demanding environment with a higher obstacle density and strategic distribution, designed to simulate complex navigational challenges, including narrow passages and densely populated obstacle areas. This scenario is expected to have a high percentage of navigable area, reflecting the available free space even to a lesser area of obstacles. Unlike the Basic Challenge Scenario, the density of critical points is expected to be higher. The ability to plan accurate and flexible routes is challenged by the presence of many narrow sections and areas.

Using these metrics to evaluate the complexity of basic and advanced challenge scenarios not only allows for quantitative comparison but also provides valuable insights into the robustness and versatility of the optimization algorithms being studied.

4.5 | Evaluation Metrics

The evaluation metrics used in this study assess both the safety and efficiency dimensions, which are critical to AGV path optimization. Each metric reflects a unique aspect of performance, facilitating a comparison of algorithmic effectiveness:

- *Non-collision time (NCT)*: NCT measures the time required by each algorithm to identify a viable path that avoids all obstacles. This metric provides insight into the convergence speed and efficiency of each algorithm, with faster NCT values indicating algorithms that can quickly adapt to complex environments.
- *Initial Safe Path time (ISPT)*: ISPT quantifies the time required for each algorithm to identify the first collision-free, safety-compliant path. This measure is particularly useful for assessing the initial exploration capabilities of the algorithms under varying environmental complexities.
- *Minimum distance to obstacles (MDO)*: The MDO metric evaluates the closest distance between any point of the AGV's path and the nearest obstacle. The calculation involves determining the MDO for each vertex of the vehicle along the trajectory and selecting the smallest value from this set of values. This metric assesses the safety margin provided by each algorithm's solution, with higher MDO values corresponding to increased anti-collision safety. When collisions are present, MDO is set to zero, distinguishing between feasible and non-feasible solutions. The formula for MDO, given by Equation (14), reflects this spatial safety criterion quantitatively.

$$MDO = min_{i\in 1\ldots n,\ k\in 1\ldots 4}\left(min_{j\in N_o} D_{ijk}\right) \tag{14}$$

  Being n the number of trajectory points and N_o the number of obstacles in the occupancy map. D_{ijk} denotes the distance from the vehicle's vertex k in the trajectory point i to the obstacle j when there is no collision.

- *Average Distance to Obstacles (ADO)*: The ADO metric calculates the average of the distances from all points along the AGV's path to the nearest obstacles, providing a holistic view of safety along the entire trajectory. High ADO values indicate that the AGV maintains a safe distance from obstacles along the route, allowing for effective collision avoidance. Equation (15) formalises this metric.

$$ADO = \frac{\sum_{i=1}^{n} min_{j\in N_o, k\in 1..4} D_{ijk}}{n} \tag{15}$$

- *TTT:* This metric calculates the time needed by an AGV to complete the planned route. It is calculated by dividing the route length by the set AGV speed. TTT serves as an efficiency indicator as it directly correlates to time. Lower TTT values imply more time-efficient routes, in line with the industrial vehicle's operational performance objectives. In this work, the time is calculated by dividing the total distance of the trajectory by the AGV's constant velocity ($v_{AGV} = 1m/s$).

4.6 | Experimental Setup

The structure of the experiments was designed to evaluate the performance of three optimization algorithms. To illustrate the results, visual representations, including result graphs and an occupancy map, were developed. For generating these results, Matlab R2022a, complemented by the Global Optimization 4.7 and Navigation 2.2 toolboxes was used. The Global Optimization toolbox is required to implement the different bio-inspired algorithms implemented. The navigation toolbox is used to generate clothoid paths. The trajectory generated by the optimal solution of each algorithm is displayed and the progression of key performance indicators is monitored throughout the optimisation process.

Conditions specific to each scenario were defined based on their complexity to ensure a consistent approach across all optimization methods. A consistent fitness function was applied throughout the iterative process, with results documented at each step. A log file was created for each metaheuristic approach, recording details for each iteration: timestamp, generation number, individual index, fitness function value, solution angle, MDO and ADO. This allowed comparisons to be made between individual solutions of the different metaheuristic techniques and the evolution of the metrics to be tracked throughout the optimization.

Data was systematically organised and stored in CSV format, capturing the evolution of the metrics over time and facilitating detailed analysis of the trajectories generated. This method allows continuous evaluation of both intermediate and final solutions, providing insight into the performance of the algorithm throughout the optimisation process. The metrics collected were analysed, including timestamps, generations, individual counts, fitness function values, minimum average distances to obstacles and the time taken to achieve collision-free and optimal solutions.

5 | Results

This section provides a detailed visual and quantitative analysis of the results obtained by using three different metaheuristic bio-inspired algorithms to solve the AGV trajectory planning problem. To ensure a reliable and fair comparison between the algorithms, 10 simulations were performed for each optimization strategy. For each algorithm, the result with the best performance in the experiments was selected, thus obtaining its performance in a consistent manner. This approach allows a clear assessment of the strengths of each algorithm under equivalent conditions, providing significant differences in the metrics evaluated.

The optimal trajectory is graphically shown in a plan view of an industrial environment. The route is represented by a red line and red dots correspond to the start, end and intermediate points, for clarity. The AGV profile is highlighted in blue, while yellow lines represent the proximity of the vehicle to surrounding obstacles. These lines show the closest distance from the vehicle edges to obstacles in the occupancy map.

In addition, the graphs show the evolution of each metric value over the iterations during optimization, using colours to distinguish the algorithms: blue for WOA, yellow for BA and green for GOA.

5.1 | Basic Challenge Scenario

To establish the trajectories, an initial occupancy map featuring 9 obstacles within a confined area was utilised as a basic challenge. Along with the start and end points, 5 intermediate points were incorporated into the map. The output angles for the start and end points were predetermined, whereas the output angles for the intermediate points served as the optimization variables. The optimization process was carried out over 60 iterations to achieve the results.

Figure 6 illustrates the optimal trajectories produced by the three optimization strategies. In this instance, all methods successfully found a collision-free solution. It is observed that in this case, both the BA and GOA find a solution that looks very similar, while the WOA shows a greater difference compared to the other two methods.

Figure 7 illustrates the progression of the fitness function for each approach throughout the optimization process. In this scenario, all methods have managed to find feasible solutions, although with slight variations. This is reflected in the fitness function's trend, which consistently ends below 1, showing success in achieving collision-free solutions. It is observed that the BA is capable of finding a feasible solution with a good fitness function value almost at the beginning of the optimization process, while the WOA takes several generations to identify solutions with a lower cost value. This demonstrates that, in this case, BA can find a feasible and high-quality solution earlier than WOA, but it is the latter that ultimately discovers a better solution. In the case of GOA, the method takes more generations to find a feasible, collision-free solution and then improves the fitness function values, eventually finding a solution that is similar but inferior to those found by the other two methods. For this case, the axes limits of Figure 7 have been adjusted to focus on the evolution of the fitness function values within the feasible solution space. Without this adjustment, the values in the infeasible (collision) region would dominate the plot, obscuring important details in the feasible region.

Figure 8 shows the MDO evolution for the basic scenario. It is observed that both BA and WOA achieve maximum values of average distance as they find satisfactory solutions to the problem. In this case, GOA does not manage to find collision-free

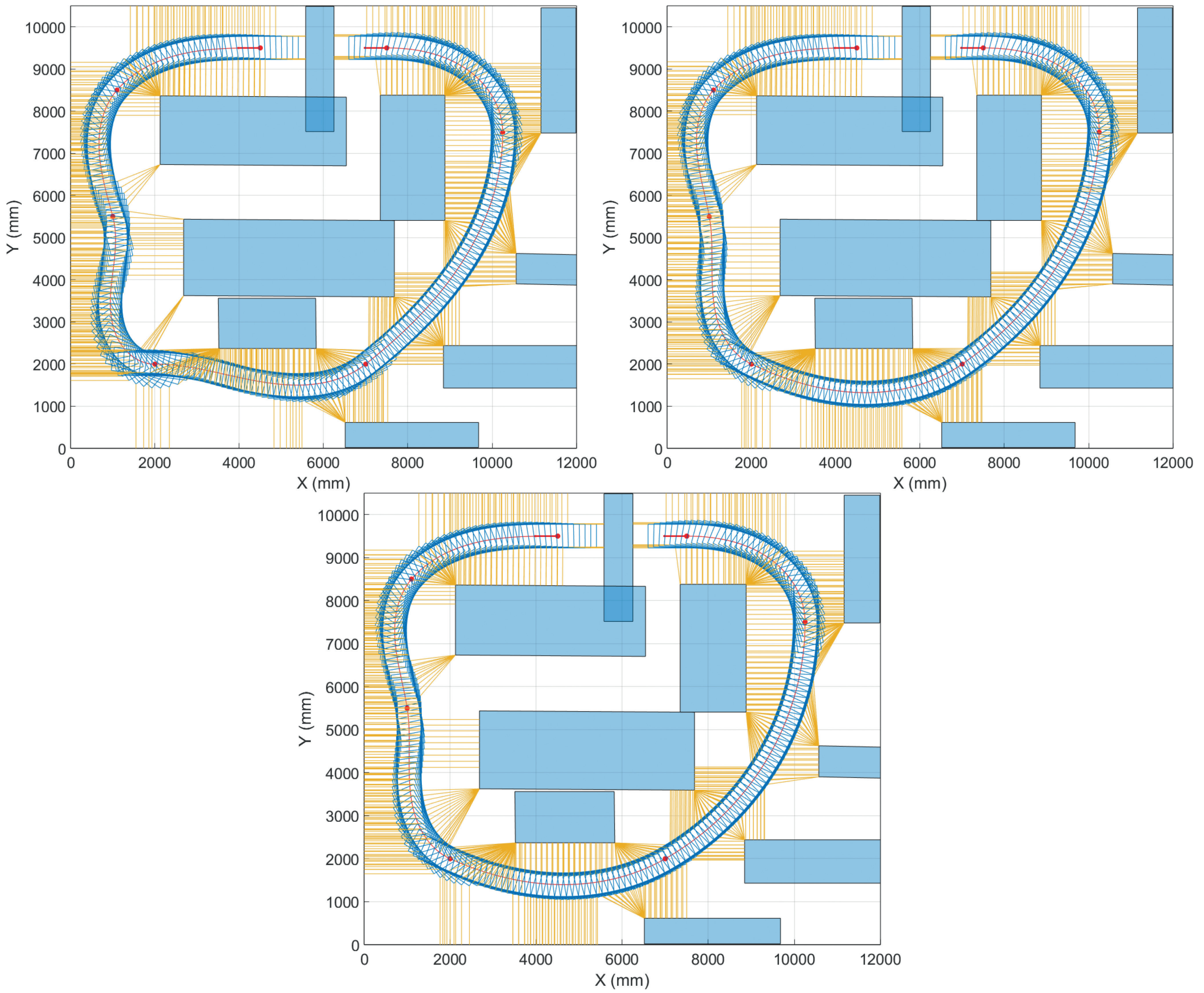

FIGURE 6 | Basic challenge scenario, optimised trajectories. (a) BA (b) WOA (c) GOA.

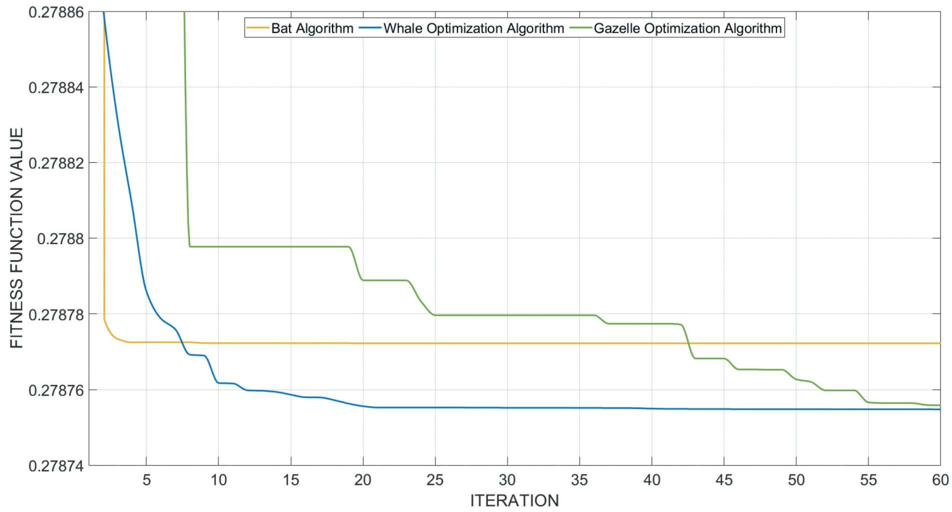

FIGURE 7 | Fitness function evolution in the basic challenge scenario.

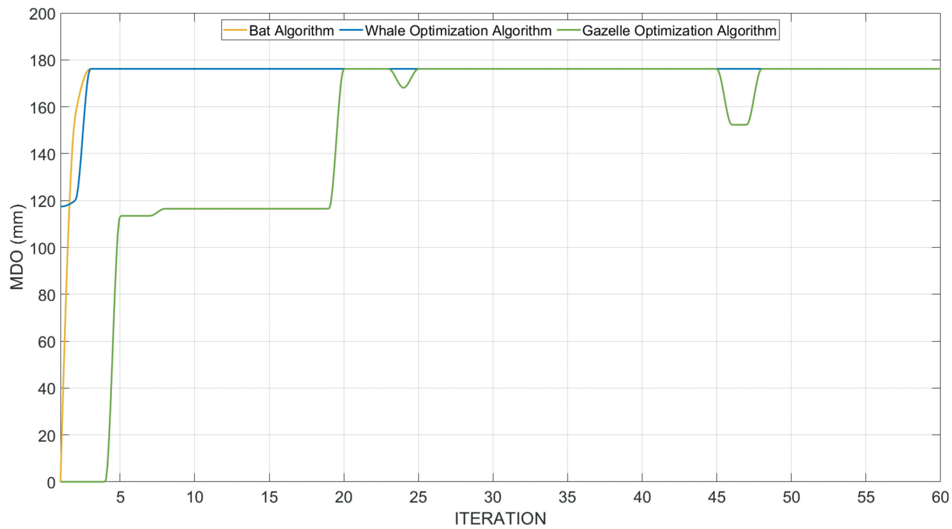

FIGURE 8 | MDO evolution in the basic challenge scenario.

solutions until the fourth iteration, showing zero values for minimum distance. From the fifth iteration onwards, the method progressively finds more suitable solutions, eventually reaching average distance values very similar to those of the other two methods.

Similarly, in Figure 9, it can be observed how the average distances to obstacles are adjusted as the method develops more optimal solutions. In this case, it is shown that WOA ultimately achieves the best ADO.

On the other hand, considering that the final part of the fitness function prioritises minimising the trajectory travel time, Figure 10 shows which method ultimately achieves the fastest solution while adhering to the safety standards established in the fitness function. In this case, we see that WOA is the method that achieves the best TTT value for its optimal solution. However, it is again observed that BA obtains the best value earlier, consistent with what was observed in the case of the fitness function.

Tables 6 and 7 present the analytical values obtained, indicating the first collision-free solution and the initial solution that meets the safety parameters.

Finally, Figure 11 presents the evolution of the solutions found by the different methods during the first five iterations, allowing for visualisation of the progress made by the methods during the most significant moments of progress in the search for the optimal solution. In this instance, it is apparent that BA is the method that converges most quickly towards near-optimal solutions, followed by WOA. For GOA, as indicated by the other metrics, the initial iterations yield lower-quality solutions, making it the slowest method in this particular scenario. On the other hand, as the methods continue to operate, BA maintains the same values achieved during the initial iterations, whereas WOA and GOA continue to improve upon their solutions. In this case, WOA converges to more suitable solutions faster than GOA; however, GOA is able to sustain improvements over time, ultimately equating its results with those of WOA.

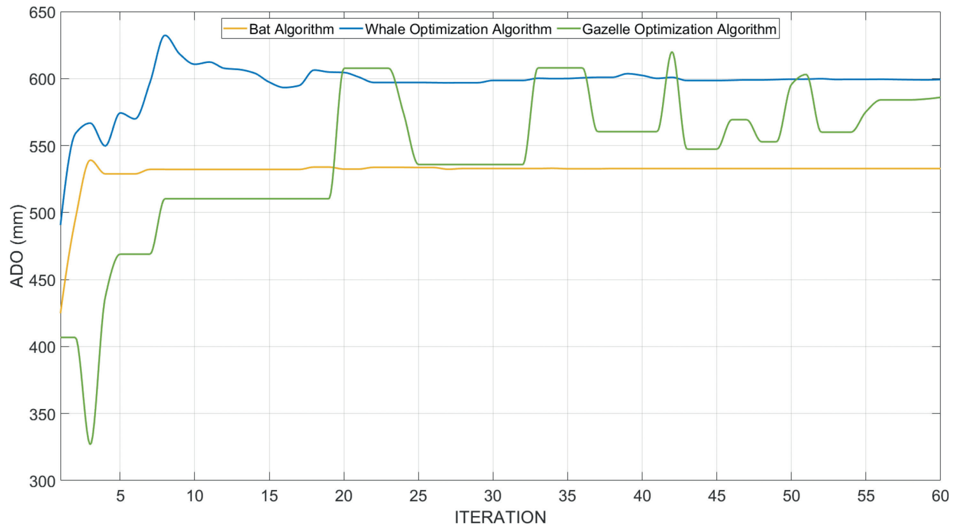

FIGURE 9 | ADO evolution in the basic challenge scenario.

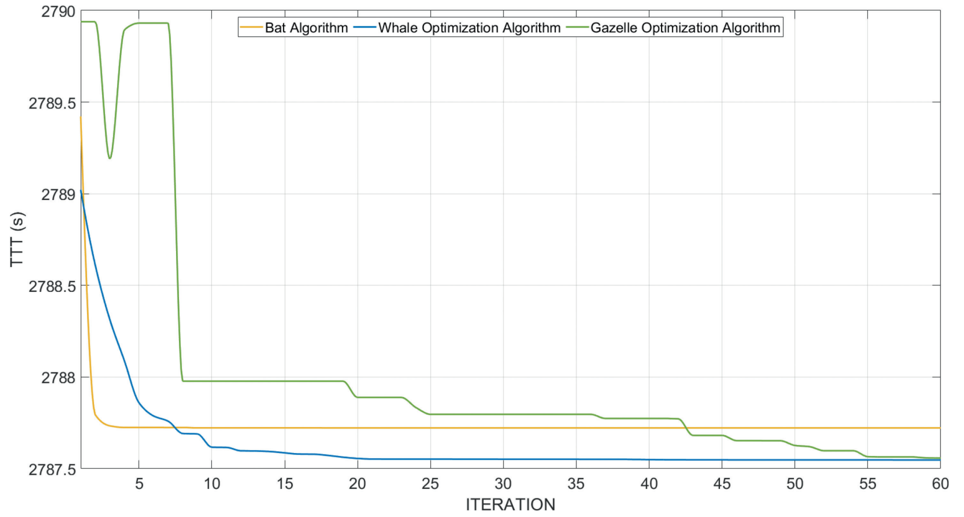

FIGURE 10 | TTT evolution in the basic challenge scenario.

5.2 | Advance Challenge Scenario

For this scenario, trajectories were determined using an advance challenge occupancy map containing 7 obstacles within a restricted area. Besides the start and end points, 5 intermediate points were introduced. The angles for the start and end points were set in advance, while the angles for the intermediate points were treated as variables in the optimization process. The optimization was conducted over 60 iterations to reach the final results.

Figure 12 illustrates the final iteration trajectories produced by the three optimization strategies. Both WOA and GOA successfully found similar and effective collision-free solutions. However, BA was unable to find a collision-free solution during the 60 iterations planned.

This situation is confirmed through the analysis of the metrics recorded during the operation of the algorithms. Figure 13 shows how both WOA and GOA follow different paths in finding better fitness functions and therefore more optimal solutions, but over the course of iterations, they manage to find fitness function values that are collision-free and meet the established safety standards. However, this same figure demonstrates that BA improves its fitness function values during its execution, but in a very slow and inefficient manner, ultimately resulting in suboptimal values far from acceptable results, as shown in Figure 12.

In Figure 14, MDO metric provides further validation of the conclusions drawn from the previous figures. In the graph corresponding to this metric, it is apparent that WOA achieves minimum distance values of 100 mm or more earlier than GOA, thus meeting the requirements set by the fitness function. This finding is consistent with the observations in Figure 13, where WOA proved to be more efficient in finding optimal solutions. As expected, BA shows a minimum distance of zero, reflecting its inability to find collision-free solutions, thereby confirming its lower performance in this scenario.

On the other hand, ADO metric reveals in Figure 15 that GOA is able to achieve higher values more rapidly. However, once both

TABLE 6 | Basic challenge scenario, first non-collision iteration analytic results.

Technique	NCT (s)	NCT iterations	ADO (mm)	MDO (mm)	Fitness function
BA	80	51	452.87	14.04	0.92978
WOA	69	42	490–72	117.43	0.27890
GOA	561	202	468.97	113.49	0.27899

TABLE 7 | Basic challenge scenario, initial safe path (ISP) analytic results.

Technique	ISP (s)	ISP iterations	ADO (mm)	MDO (mm)	Fitness function
BA	89	57	484.72	107.37	0.27889
WOA	69	42	490.72	117.43	0.27890
GOA	561	202	468.97	113.49	0.27899

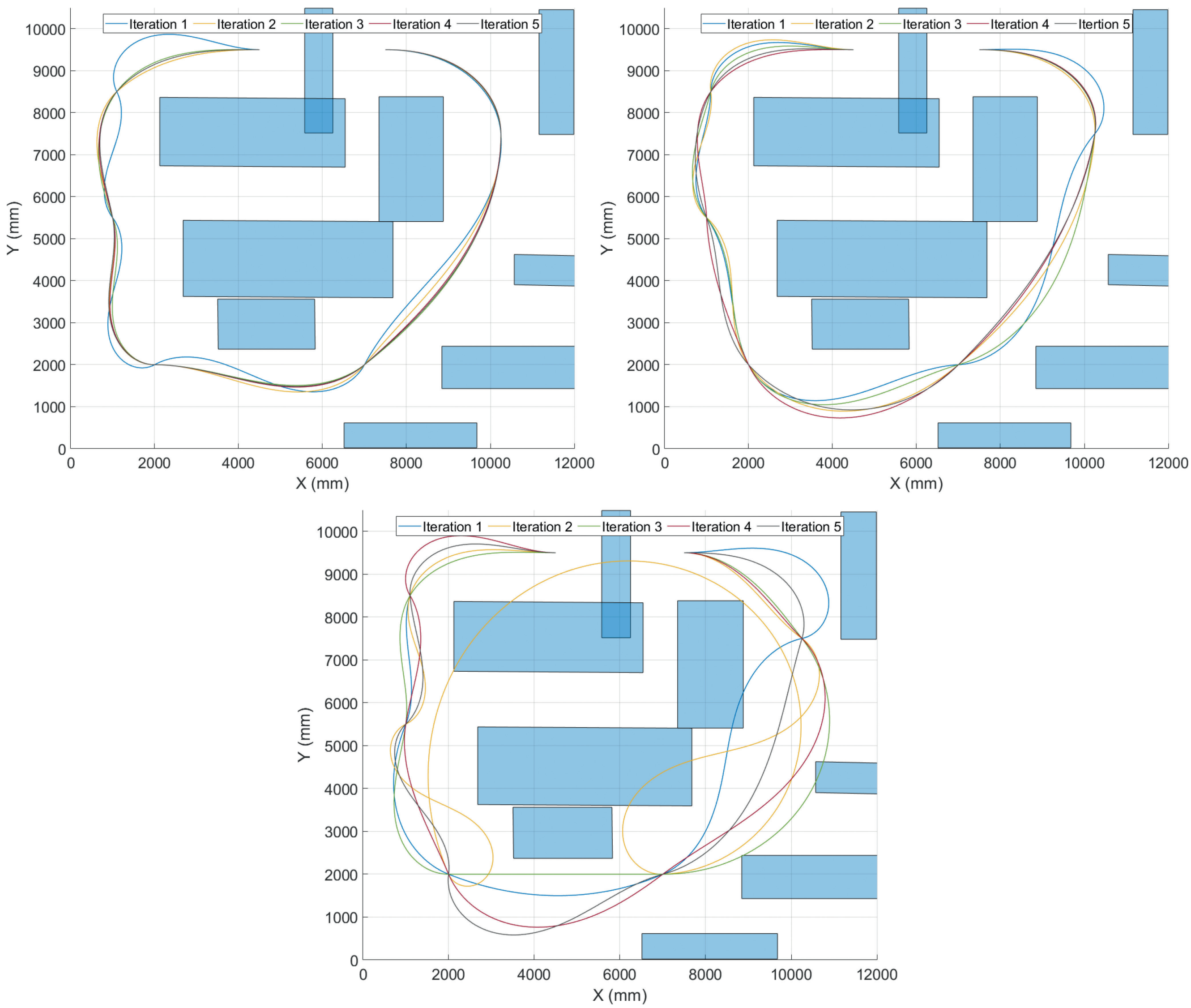

FIGURE 11 | Solution trajectories during first five iterations for basic challenge scenario. (a) BA, (b) WOA, (c) GOA.

WOA and GOA reach solutions that meet the established safety standards, the average distance values converge to similar levels. In this particular case, GOA ultimately achieves the highest ADO values. As seen previously, BA remains stalled at very low and inadequate values due to the collisions present in the trajectories it identifies.

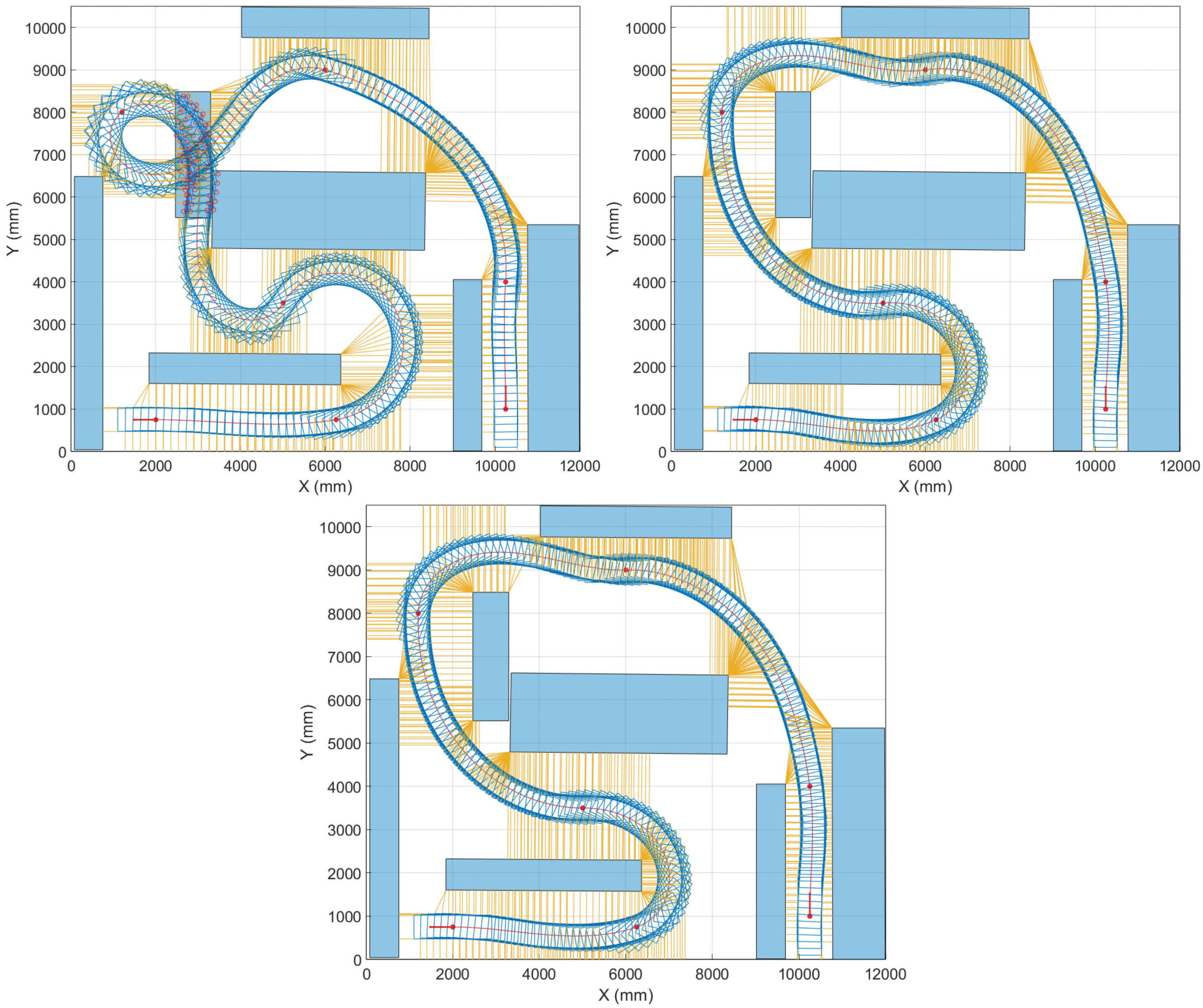

FIGURE 12 | Advance challenge scenario, optimised trajectories. (a) BA, (b) WOA, (c) GOA.

Figure 16 shows metric TTT where both WOA and GOA progressively improve their times as they identify better solutions and, consequently, better fitness functions. For GOA, it is apparent that once it finds fitness functions within the established safety limits, the travel times of the trajectories start to improve accordingly. During the middle iterations, values close to the final results begin to emerge, aligning with the findings from the other metrics analysed. In the case of BA, we observe that the TTT metric improves only marginally and slowly, mirroring its performance in the fitness function analysis.

Tables 8 and 9 present the analytical values obtained, indicating the first collision-free solution and the best value of the whole series.

Ultimately, Figure 17 displays the progression of solutions identified by the studied methods over the first five iterations, providing a visual representation of the advancements made by each method during the critical phases of the optimization process. In this scenario, it becomes clear that BA exhibits minimal improvement during the initial iterations, stagnating in non-acceptable solutions. Meanwhile, GOA also produces solutions that are far from optimal in the early iterations. In contrast, WOA stands out as the method that more consistently achieves near-acceptable trajectories during these initial stages of iteration showing the quickest optimization progress for this scenario.

5.3 | Discussion on Algorithmic Performance

Upon examining the performance of the three bio-inspired algorithms applied to the AGV trajectory planning problem, several distinct patterns emerge. While each algorithm operates under the shared principles of nature-inspired optimization, their unique mechanisms lead to varying degrees of effectiveness depending on the specific scenario.

The computational scalability of BA, WOA and GOA, as well as optimization algorithms in general, increases as the complexity of the problem increases, in this case with larger

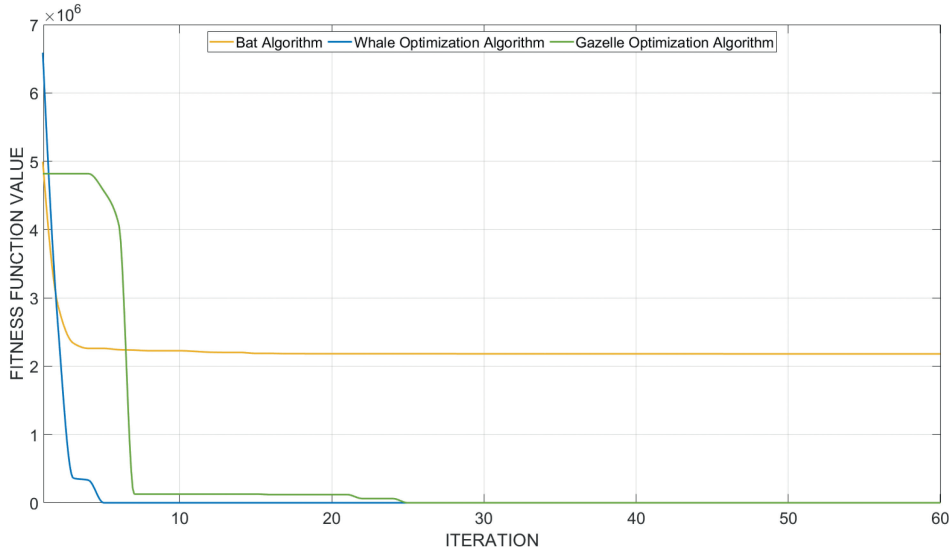

FIGURE 13 | Fitness function evolution in the advance challenge scenario.

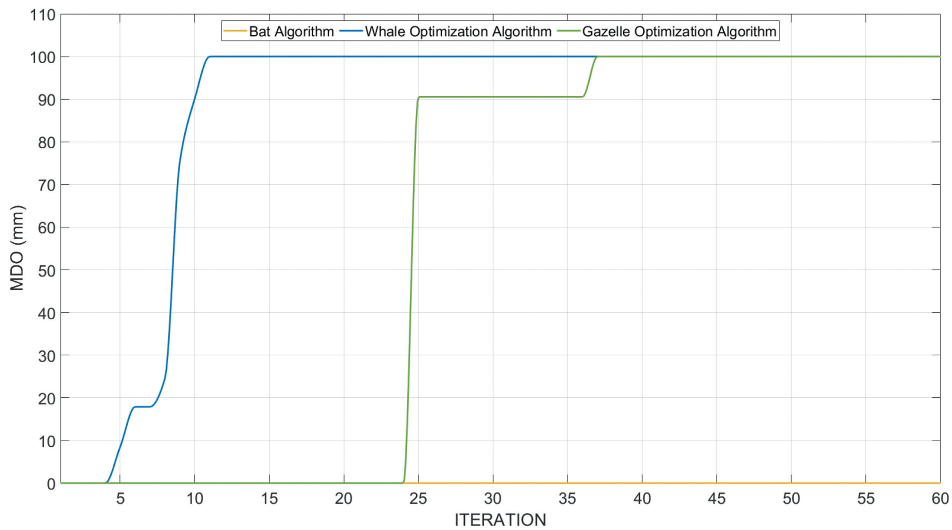

FIGURE 14 | MDO evolution in the advance challenge scenario.

occupancy maps and higher obstacle density. When evaluating the computational demand, it is useful to distinguish between two factors: time per iteration and time to convergence. The time per iteration for all algorithms is expected to increase with the size of the map and the number of obstacles since all algorithms require additional computations to evaluate larger and more complex environments.

However, the time to convergence, or the total time required to reach a satisfactory solution, varies between the algorithms due to differences in their search strategies. To provide insight into each algorithm's trade-off between safety and computational efficiency, the following cost–benefit analysis highlights the features associated with BA, WOA and GOA in AGV route planning.

- *BA*: maintains relatively low computational demands due to its focus on local search and exploitation rather than extensive exploration. This strategy allows it to operate efficiently, but with one drawback: BA can occasionally compromise safety in complex environments with many obstacles as it tends to establish locally optimal routes without exploring other safer alternatives. As a population-based algorithm, BA scales linearly with the number of obstacles and the map size and each bat (individual) evaluates its position relative to the obstacles at each iteration. This linear scaling makes BA computationally feasible for a wide range of problem sizes, although its computational burden gradually increases with more complex scenarios.
- *WOA*: leverages spiral and encircling movements to achieve a good level of safety by balancing exploration and exploitation. Its search strategy allows for reliable obstacle avoidance at the cost of increased computational demand, as each update involves complex interactions based on the best-known solution at each iteration. WOA is sensitive to increasing problem complexity; as the size of the occupancy map and obstacle density increase, spiral and encircling mechanisms tend to slow down its performance, particularly in highly complex environments.

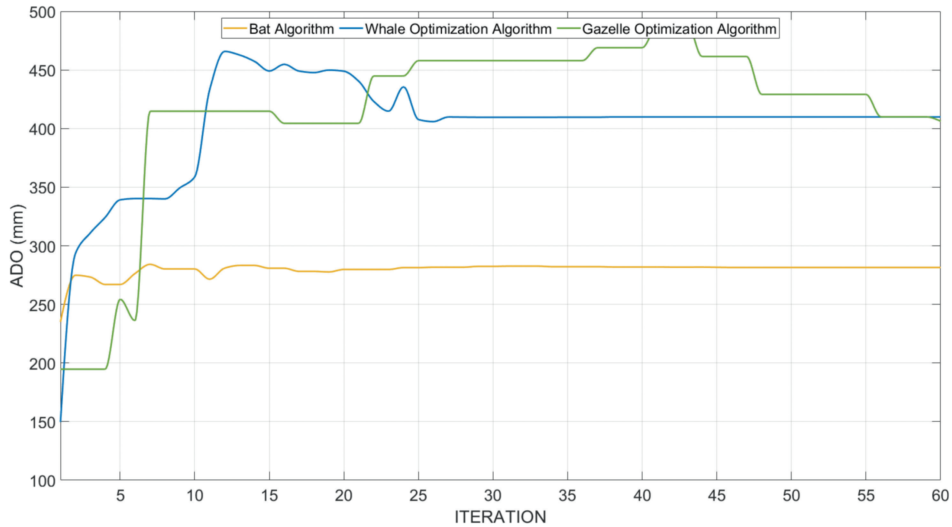

FIGURE 15 | ADO evolution in the advance challenge scenario.

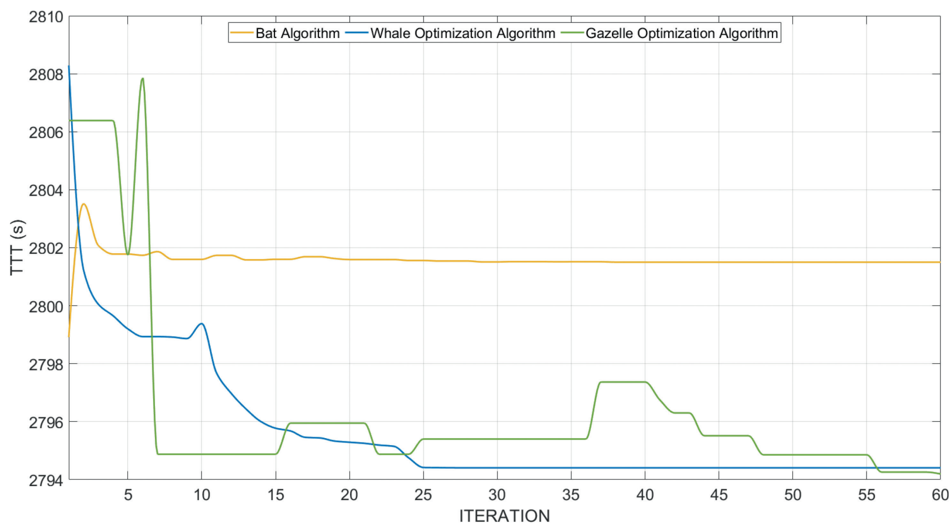

FIGURE 16 | TTT evolution in the advance challenge scenario.

TABLE 8 | Advance challenge scenario, first non-collision iteration analytic results.

Technique	Non-collision time (s)	Non-collision iterations	ADO (mm)	MDO (mm)	Fitness function
BA	—	—	—	—	—
WOA	275	204	339.19	8.46	0.95765
GOA	3355	1247	457.96	90.53	0.54733

Consequently, WOA is well-suited for moderate problem sizes where safety is prioritised over computational speed and where computational resources can support its intensive update requirements

- *GOA*: Its adaptability due to its perception range and PSR mechanism make it highly effective at maintaining safe distances in environments with many obstacles, allowing navigation in complex spaces although also increasing computational demand due to frequent recalculations over multiple iterations. GOA exhibits high sensitivity to increasing problem complexity; as the size of the occupancy map and obstacle density increase, the computational load of the algorithm intensifies, requiring it to evaluate a larger search space while maintaining adaptive updates. Consequently, GOA's runtime can grow significantly in very complex scenarios, making it more suitable for applications where safety is paramount but computational resources can support its demanding requirements

Commonalities among the three algorithms include their nature-inspired mechanisms, which allow them to efficiently navigate

TABLE 9 | Advance challenge scenario and initial safe path (ISP) analytic results.

Technique	ISP (s)	ISP iterations	ADO (mm)	MDO (mm)	Fitness function
BA	—	—	—	—	—
WOA	1155	502	390.74	100.05	0.27992
GOA	5124	1840	468.96	100.03	0.27973

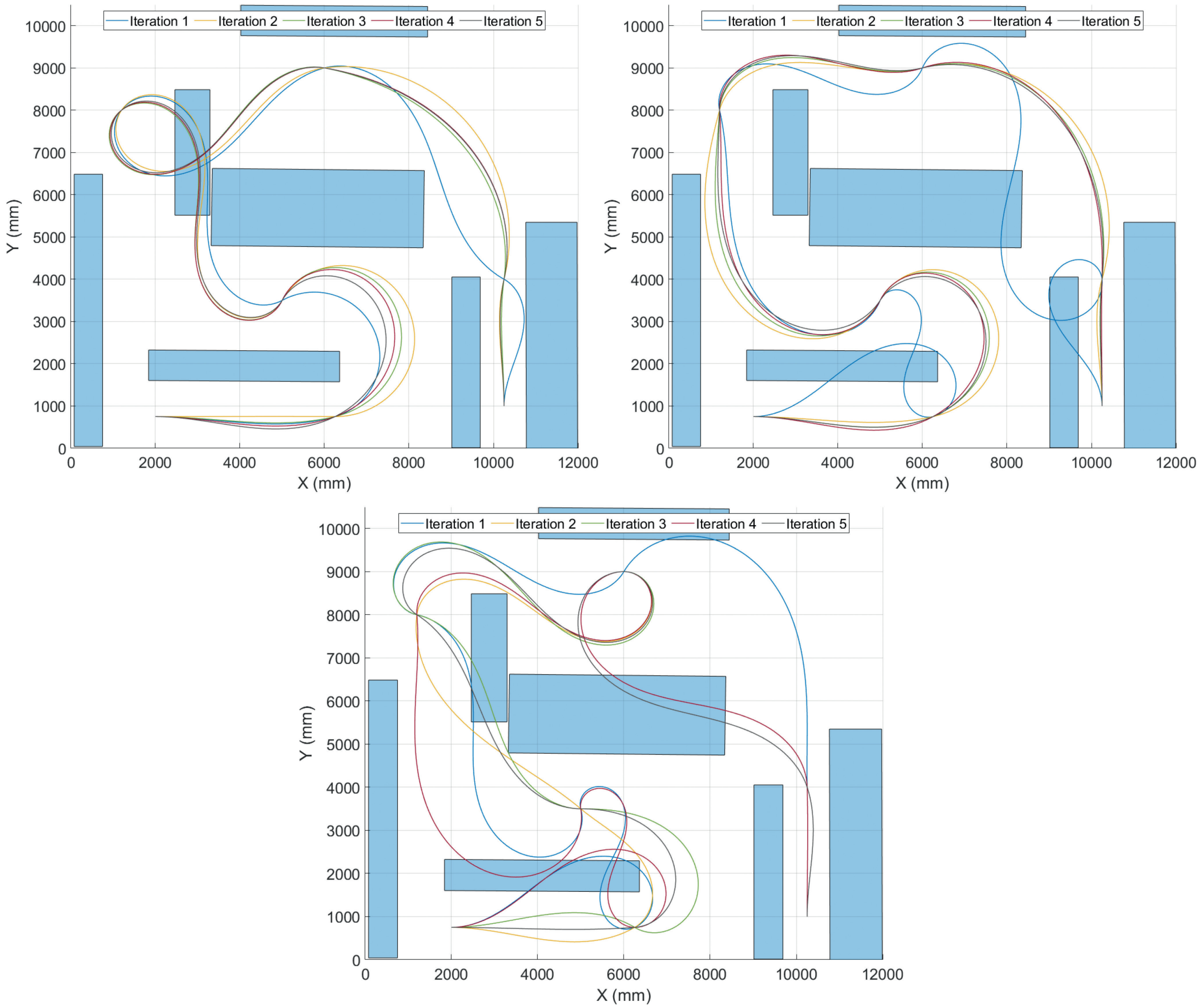

FIGURE 17 | Solution trajectories during first five iterations for advance challenge scenario. (a) BA, (b) WOA, (c) GOA.

large and complex search spaces. All three methods implement strategies to balance the trade-off between exploration (global search) and exploitation (local search), which is significant for avoiding local minima and ensuring convergence towards an optimal solution. Additionally, each algorithm demonstrates a capacity for adaptability, allowing it to perform effectively across varying levels of problem complexity.

However, the results indicate several differences in how these algorithms perform under varying conditions:

1. Exploration and Convergence Speed: BA showed a remarkable ability to quickly converge to feasible solutions in less challenging scenarios, using its echolocation-based approach to optimise its search process. WOA, however, outperformed in more complex environments, where its bubble-net hunting strategy allowed it to maintain a balanced exploration-exploitation dynamic. GOA, while effective in difficult landscapes, exhibited slower initial convergence due to its emphasis on exploration through evasive manoeuvres.

2. Performance consistency: BA underperforms WOA and GOA due to its search strategy, which focuses on local exploitation once it finds a promising region. This may limit BA's adaptability to complex, obstacle-rich environments, as it may struggle to dynamically adjust to new route constraints. In contrast, WOA, through spiral and enveloping motions and GOA, through perceptual range and PSR adjustments, use adaptive mechanisms that allow for more flexible route exploration and an effective balance between exploration and exploitation. Across the different scenarios, WOA demonstrated consistent performance, reliably finding optimal or near-optimal solutions even in scenarios with high obstacle density. GOA also proved highly effective in scenarios requiring precision and careful navigation, although it sometimes required many iterations to achieve optimal results in less complex environments.
3. Handling of Complex Scenarios: In environments characterised by dense obstacles or narrow passages, GOA's agility and focus on immediate collision avoidance provided an advantage, particularly in achieving high safety margins. WOA's adaptive spiral and encircling strategies also allowed it to perform well in such challenging environments. Conversely, BA encountered difficulties in these scenarios, frequently needing additional iterations to successfully navigate around obstacles.

The findings suggest that while each algorithm has its strengths, its effectiveness is context-dependent. WOA stands out as a versatile and consistently high-performing algorithm, particularly in more complex environments. GOA excels in scenarios demanding high precision and safety but may require more iterations in simpler cases. BA, with its rapid convergence in straightforward scenarios, remains a valuable tool but may need additional support in more challenging tasks. These insights indicate that combining the strengths of these algorithms could lead to a more robust approach for optimising AGV trajectories across various scenarios.

Applying these three algorithms, BA, WOA and GOA, to dynamic and potentially rapidly changing environments presents interesting novel challenges, as these algorithms are generally designed for static scenarios. In changing environments, where obstacle positions frequently change in real-time, adapting these algorithms would require substantial modification.

A practical approach might involve periodic route re-evaluation rather than continuous optimization, which would allow algorithms to update the AGV's route at intervals rather than in real-time. Another option would be a hybrid method, where BA, WOA or GOA handled global route optimization, while a reactive collision avoidance layer addressed immediate and unforeseen obstacles. This layered approach would allow AGVs to navigate safely while the optimization algorithm periodically re-evaluated the best route.

6 | Conclusions and Future Works

The comparative study of bio-inspired algorithms—BA, WOA and GOA—highlights their effectiveness in optimising AGV trajectory planning within complex environments. While all three algorithms are inspired by natural phenomena and share a common goal of balancing exploration and exploitation during the search process, they exhibit unique characteristics that differentiate their performance and application suitability. Each algorithm demonstrated specific strengths, which varied depending on the complexity of the scenario.

In the basic challenge scenario, all three algorithms were successful in generating efficient and collision-free trajectories. WOA emerged as the most effective, achieving faster convergence and slightly more efficient paths. This suggests that WOA's balance between exploration and exploitation is particularly advantageous in simpler environments, where rapid convergence to a solution is critical.

However, in the advanced challenge scenario, BA failed to find a collision-free solution, indicating its limitations in more complex environments. On the other hand, both WOA and GOA managed to find suitable solutions, with GOA excelling in navigating tight spaces and avoiding densely packed obstacles. GOA's performance in this scenario underscores its suitability for highly constrained environments where other algorithms, including BA, may struggle.

WOA demonstrated consistent performance across both scenarios, confirming its robustness and adaptability. While WOA may not have always been the fastest, its reliability in finding feasible solutions in both simple and complex scenarios highlights its potential as a versatile tool in AGV trajectory planning.

The findings of this study emphasise the importance of selecting the appropriate bio-inspired algorithm based on the specific challenges of the environment. The varying performance across different scenarios suggests that hybrid algorithms, which integrate the strengths of WOA and GOA, could offer improved performance in AGV trajectory optimization.

While BA, WOA and GOA have proven effective in AGV path planning in static or semi-dynamic environments, applying these algorithms to highly dynamic environments, with rapidly changing obstacles, presents unique challenges. Future research could explore adaptations that allow these algorithms to operate in environments where real-time changes are an important factor to consider.

A potential approach might involve periodic route re-evaluation, which would allow algorithms to update the AGV's route at intervals rather than in continuous real-time optimization. This method would allow the AGV to adjust its route as the environment changes, providing a balance between computational efficiency and responsiveness to dynamic conditions.

Another promising direction for future research would be a hybrid approach, where BA, WOA or GOA handled global route optimization, while a reactive collision avoidance layer addressed immediate and unforeseen obstacles. This layered approach would allow AGVs to navigate safely while the optimization algorithm periodically re-evaluated the best route.

Exploring these possibilities would improve the applicability of BA, WOA and GOA in complex real-world scenarios, supporting the development of AGV systems that can navigate dynamic environments autonomously. This research would address the growing demands of Industry 4.0, where flexible and adaptable AGV operations are increasingly essential for efficient, safe and automated logistics.

Finally, the integration of these bio-inspired algorithms with advanced machine learning techniques, such as reinforcement learning, may offer new opportunities to improve AGV decision-making in highly dynamic and unpredictable environments. This integration could lead to more intelligent and autonomous navigation systems capable of adapting to real-time challenges in industrial environments.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Aadithya, G., S. Krishnan, V. Arvindh, and K. Sivanathan. 2018. "Online Decentralized Receding Horizon Trajectory Optimization for Multi-Robot Systems." https://doi.org/10.48550/arXiv.1812.11135.

Abdollahzadeh, B., F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili. 2022. "Mountain Gazelle Optimizer: A New Nature-Inspired Metaheuristic Algorithm for Global Optimization Problems." *Advances in Engineering Software* 174: 103282. https://doi.org/10.1016/j.advengsoft.2022.103282.

Abualigah, L., A. Diabat, and R. A. Zitar. 2022. "Orthogonal Learning Rosenbrock's Direct Rotation With the Gazelle Optimization Algorithm for Global Optimization." *Mathematics* 10, no. 23: 4509. https://doi.org/10.3390/math10234509.

Agushaka, J. O., A. E. Ezugwu, and L. Abualigah. 2022. "Gazelle Optimization Algorithm: A Novel Nature-Inspired Metaheuristic Optimizer." *Neural Computing and Applications* 35, no. 5: 4099–4131. https://doi.org/10.1007/s00521-022-07854-6.

Alqudsi, Y. S., A. H. Kassem, and G. M. El-Bayoumi. 2021. "Trajectory Generation and Optimization Algorithm for Autonomous Aerial Robots." In *2021 1st International Conference on Emerging Smart Technologies and Applications (Esmarta)*, 1–8. Sana'a, Yemen: IEEE. https://doi.org/10.1109/esmarta52612.2021.9515735.

Bayona, E., J. E. Sierra-García, and M. Santos. 2024. "Comparative Analysis of Metaheuristic Optimization Methods for Trajectory Generation of Automated Guided Vehicles." *Electronics (Switzerland)* 13, no. 4: 728. https://doi.org/10.3390/ELECTRONICS13040728.

Bayona, E., J. E. Sierra-García, M. Santos, and I. Mariolis. 2024. "In Search of the Best Fitness Function for Optimum Generation of Trajectories for Automated Guided Vehicles." *Engineering Applications of Artificial Intelligence* 133: 108440. https://doi.org/10.1016/J.ENGAPPAI.2024.108440.

Bertolazzi, E., and M. Frego. 2015. "G1 Fitting With Clothoids." *Mathematical Methods in the Applied Sciences* 38, no. 5: 881–897. https://doi.org/10.1002/mma.3114.

Chandrasekaran, G., N. S. Kumar, P. R. Karthikeyan, K. Vanchinathan, N. Priyadarshi, and B. Twala. 2022. "Test Scheduling and Test Time Minimization of System-On-Chip Using Modified Bat Algorithm." *IEEE Access* 10: 126199–126216. https://doi.org/10.1109/access.2022.3224924.

Chaudhry, I. A., A. F. Rafique, I. A.-Q. Elbadawi, et al. 2022. "Integrated Scheduling of Machines and Automated Guided Vehicles (Agvs) in Flexible Job Shop Environment Using Genetic Algorithms." *International Journal of Industrial Engineering Computations* 13, no. 3: 343–362. https://doi.org/10.5267/j.ijiec.2022.2.002.

Chen, Y., S. Shi, Z. Chen, T. Wang, L. Miao, and H. Song. 2023. "Optimizing Port Multi-Agv Trajectory Planning Through Priority Coordination: Enhancing Efficiency and Safety." *Axioms* 12, no. 9: 900. https://doi.org/10.3390/axioms12090900.

Chevallier, M., and C. Clairmont. 2024. "Machine Learning-Based Surrogate Model for Genetic Algorithm With Aggressive Mutation for Feature Selection." *International Journal of Hybrid Intelligent Systems* 20, no. 3: 259–274. https://doi.org/10.3233/his-240019.

Dalla Vedova, M. D. L., P. C. Berri, and S. Re. 2019. "A Comparison of Bio-Inspired Meta-Heuristic Algorithms for Aircraft Actuator Prognostics." In *Proceedings of the 29th European Safety and Reliability Conference (Esrel)*, 1064–1071. Politecnico di Torino, Italy: DIMEAS. https://doi.org/10.3850/978-981-11-2724-3_0476-cd.

Dao, T.-K., T.-G. Ngo, J.-S. Pan, T.-T.-T. Nguyen, and T.-T. Nguyen. 2024. "Enhancing Path Planning Capabilities of Automated Guided Vehicles in Dynamic Environments: Multi-Objective Pso and Dynamic-Window Approach." *Biomimetics* 9, no. 1: 35. https://doi.org/10.3390/biomimetics9010035.

Du, J., J. Hou, H. Wang, and Z. Chen. 2023. "Application of an Improved Whale Optimization Algorithm in Time-Optimal Trajectory Planning for Manipulators." *Mathematical Biosciences and Engineering* 20, no. 9: 16304–16329. https://doi.org/10.3934/mbe.2023728.

Ekinci, S., D. Izci, and A. G. Hussien. 2024. "Comparative Analysis of the Hybrid Gazelle-Nelder–Mead Algorithm for Parameter Extraction and Optimization of Solar Photovoltaic Systems." *IET Renewable Power Generation* 18, no. 6: 959–978. https://doi.org/10.1049/rpg2.12974.

Gonçalves, C. B., J. R. Souza, and H. Fernandes. 2022. "Cnn Architecture Optimization Using Bio-Inspired Algorithms for Breast Cancer Detection in Infrared Images." *Computers in Biology and Medicine* 142: 105205. https://doi.org/10.1016/j.compbiomed.2021.105205.

Hasanien, H. M., I. Alsaleh, M. Tostado-Véliz, A. Alassaf, A. Alateeq, and F. Jurado. 2023. "Optimal Parameters Estimation of Lithium-Ion Battery in Smart Grid Applications Based on Gazelle Optimization Algorithm." *Energy* 285: 129509. https://doi.org/10.1016/j.energy.2023.129509.

Hason, F., and S. Al-Darraji. 2023. "A Review: On Bio-Inspired Optimization Methods for Path Planning of Mobile Robot." *Iraqi Journal of Intelligent Computing and Informatics (IJICI)* 2, no. 1: 1–10. https://doi.org/10.52940/ijici.v2i1.24.

Jakšić, Z., S. Devi, O. Jakšić, and K. Guha. 2023. "A Comprehensive Review of Bio-Inspired Optimization Algorithms Including Applications in Microelectronics and Nanophotonics." *Biomimetics* 8, no. 3: 278. https://doi.org/10.3390/biomimetics8030278.

Johnvictor, A. C., V. Durgamahanthi, R. M. Pariti Venkata, and N. Jethi. 2022. "Critical Review of Bio-Inspired Optimization Techniques." *WIREs Computational Statistics* 14, no. 1: e1528. https://doi.org/10.1002/wics.1528.

Khalife, E., M. Kaveh, A. Younesi, D. Balasubramanian, S. Khanmohammadi, and B. Najafi. 2022. "Comparative of Various Bio-Inspired Meta-Heuristic Optimization Algorithms in Performance and Emissions of Diesel Engine Fuelled With b5 Containing Water and Cerium Oxide Additive Blends." *International Journal of Energy Research* 46, no. 15: 21266–21280. https://doi.org/10.1002/er.8315.

Khamis, A., A. Hussein, and A. Elmogy. 2015. "Multi-Robot Task Allocation: A Review of the State-Of-The-Art." In *Cooperative Robots and Sensor Networks 2015*, 31–51. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-18299-5_2.

Koyama, S., S. M. Chase, A. S. Whitford, M. Velliste, A. B. Schwartz, and R. E. Kass. 2009. "Comparison of Brain–Computer Interface Decoding Algorithms in Open-Loop and Closed-Loop Control." *Journal of Computational Neuroscience* 29, no. 1–2: 73–87. https://doi.org/10.1007/s10827-009-0196-9.

Li, Y., B. Ren, L. Liu, and L. Tang. 2023. "Agv Scheduling Problem With Flexible Charging Operations and Limited Number of Charging Stations." In *2023 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 1–6. Marseille, France: IEEE. https://doi.org/10.1109/icnsc58704.2023.10319028.

Liang, S., S. Fa, Z. Chen, and Y. Li. 2023. "Fast Safe Rectangular Corridor-Based Online Agv Trajectory Optimization With Obstacle Avoidance." https://api.semanticscholar.org/CorpusID:262012877.

Loucif, F., S. Kechida, and A. Sebbagh. 2020. "Whale Optimizer Algorithm to Tune Pid Controller for the Trajectory Tracking Control of Robot Manipulator." *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42, no. 1: 1–11. https://doi.org/10.1007/s40430-019-2074-3.

Lu, Y., K. Chen, F. Yao, R. Chai, L. Cui, and W. Liang. 2023. "Agv Trajectory Planning Based on Multi-Objective Quantum Particle Swarm Optimization Algorithm in the Sight of Essential Supply Distribution During an Epidemic." In *2023 35th Chinese Control and Decision Conference (Ccdc)*, 1837–1842. Yichang, China: IEEE. https://doi.org/10.1109/CCDC58219.2023.10327463.

Miljković, Z., and M. Petrović. 2020. "Single Mobile Robot Scheduling Problem: A Survey of Current Biologically Inspired Algorithms, Research Challenges and Real-World Applications." In *New Technologies, Development and Application Ii*, 33–41. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-18072-0_4.

Mirjalili, S., and A. Lewis. 2016. "The Whale Optimization Algorithm." *Advances in Engineering Software* 95: 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

Mohajeri, S., F. Harsej, M. Sadeghpour, and J. Khaleghinia 2022. "Industry 4.0 Technologies Assessment: An Integrated Reverse Supply Chain Model With the Whale Optimization Algorithm." *International Journal of Nonlinear Analysis and Applications* 13, no. 1: 1803–1825. https://doi.org/10.22075/ijnaa.2021.25194.2943.

Mohammed, H. M., S. U. Umar, and T. A. Rashid. 2019. "A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm." *Computational Intelligence and Neuroscience* 2019: 1–25. https://doi.org/10.1155/2019/8718571.

Molina, D., J. Poyatos, J. D. Ser, S. García, A. Hussain, and F. Herrera. 2020. "Comprehensive Taxonomies of Nature- and Bio-Inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations." *Cognitive Computation* 12, no. 5: 897–939. https://doi.org/10.1007/s12559-020-09730-8.

Morrell, B., R. Thakker, G. Merewether, et al. 2018. "Comparison of Trajectory Optimization Algorithms for High-Speed Quadrotor Flight Near Obstacles." *IEEE Robotics and Automation Letters* 3, no. 4: 4399–4406. https://doi.org/10.1109/lra.2018.2868866.

Okwu, M. O., and L. K. Tartibu. 2021. "Bat Algorithm." In *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*, Studies in Computational Intelligence, vol. 927. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-61111-8_8.

Oyekanlu, E. A., A. C. Smith, W. P. Thomas, et al. 2020. "A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-Based Smart Manufacturing Applications." *IEEE Access* 8: 202312–202353. https://doi.org/10.1109/ACCESS.2020.3035729.

Peñacoba, M., J. E. Sierra-García, M. Santos, and I. Mariolis. 2023. "Path Optimization Using Metaheuristic Techniques for a Surveillance Robot." *Applied Sciences* 13, no. 20: 11182. https://doi.org/10.3390/app132011182.

Perez, J., P. Melin, O. Castillo, F. Valdez, C. Gonzalez, and G. Martinez. 2018. "Trajectory Optimization for an Autonomous Mobile Robot Using the Bat Algorithm." In *Fuzzy Logic in Intelligent System Design*, edited by P. Melin, O. Castillo, J. Kacprzyk, M. Reformat, and W. Melek, 232–241. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-67137-6_25.

Qin, S., H. Zeng, W. Sun, J. Wu, and J. Yang. 2024. "Multi-Strategy Improved Particle Swarm Optimization Algorithm and Gazelle Optimization Algorithm and Application." *Electronics* 13, no. 8: 1580. https://doi.org/10.3390/electronics13081580.

Rani, S., H. Babbar, P. Kaur, M. D. Alshehri, and S. H. A. Shah. 2023. "An Optimized Approach of Dynamic Target Nodes in Wireless Sensor Network Using Bio Inspired Algorithms for Maritime Rescue." *IEEE Transactions on Intelligent Transportation Systems* 24, no. 2: 2548–2555. https://doi.org/10.1109/tits.2021.3129914.

Rehberg, W., J. Ortiz-Haro, M. Toussaint, and W. Honig. 2023. "Comparison of Optimization-Based Methods for Energy-Optimal Quadrotor Motion Planning." https://doi.org/10.48550/arXiv.2304.14062.

Roberge, V., M. Tarbouchi, and G. Labonte. 2013. "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time Uav Path Planning." *IEEE Transactions on Industrial Informatics* 9, no. 1: 132–141. https://doi.org/10.1109/tii.2012.2198665.

Salamat, B., and A. Tonello. 2017. "Stochastic Trajectory Generation Using Particle Swarm Optimization for Quadrotor Unmanned Aerial Vehicles (UAVs)." *Aerospace* 4, no. 2: 27. https://doi.org/10.3390/aerospace4020027.

Savsani, P., R. L. Jhala, and V. J. Savsani. 2016. "Comparative Study of Different Metaheuristics for the Trajectory Planning of a Robotic Arm." *IEEE Systems Journal* 10, no. 2: 697–708. https://doi.org/10.1109/jsyst.2014.2342292.

Sierra-Garcia, J. E., and M. Santos. 2024. "Combining Reinforcement Learning and Conventional Control to Improve Automatic Guided Vehicles Tracking of Complex Trajectories." *Expert Systems* 41, no. 2: e13076. https://doi.org/10.1111/exsy.13076.

Sodiya, E. O., U. J. Umoga, O. O. Amoo, and A. Atadoga. 2024. "Ai-Driven Warehouse Automation: A Comprehensive Review of Systems." *GSC Advanced Research and Reviews* 18, no. 2: 272–282. https://doi.org/10.30574/gscarr.2024.18.2.0063.

Soria, E., F. Schiano, and D. Floreano. 2022. "Distributed Predictive Drone Swarms in Cluttered Environments." *IEEE Robotics and Automation Letters* 7, no. 1: 73–80. https://doi.org/10.1109/lra.2021.3118091.

Soto, R., B. Crawford, C. Zec, A. Alarcón, and B. Almonacid. 2016. "A Bat Algorithm to Solve the Manufacturing Cell Design Problem." In *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. Canaria, Spain: IEEE. https://doi.org/10.1109/cisti.2016.7521577.

Srivastava, S., and S. K. Sahana. 2019. "Bat Algorithm-Based Traffic Signal Optimization Problem." In *Soft Computing for Problem Solving*, edited by J. C. Bansal, K. N. Das, A. Nagar, K. Deep, and A. K. Ojha, 927–936. Singapore: Springer. https://doi.org/10.1007/978-981-13-1592-3_74.

Suárez, P., A. Iglesias, and A. Gálvez. 2019. "Make Robots Be Bats: Specializing Robotic Swarms to the Bat Algorithm." *Swarm and Evolutionary Computation* 44: 113–129. https://doi.org/10.1016/j.swevo.2018.01.005.

Vinaykumar, V. N., J. A. Babu, and J. Frnda. 2023. "Optimal Guidance Whale Optimization Algorithm and Hybrid Deep Learning Networks for Land Use Land Cover Classification." *EURASIP Journal on*

Advances in Signal Processing 2023, no. 1: 1–21. https://doi.org/10.1186/s13634-023-00980-w.

Wang, J., Y. Luo, and X. Tan. 2021. "Path Planning for Automatic Guided Vehicles (Agvs) Fusing Mh-Rrt With Improved Teb." *Actuators* 10, no. 12: 314. https://doi.org/10.3390/act10120314.

Wu, D., L. Wu, T. Wen, and L. Li. 2024. "Microgrid Operation Optimization Method Considering Power-To-Gas Equipment: An Improved Gazelle Optimization Algorithm." *Symmetry* 16, no. 1: 83. https://doi.org/10.3390/sym16010083.

Yan, Z., J. Zhang, J. Zeng, and J. Tang. 2022. "Three-Dimensional Path Planning for Autonomous Underwater Vehicles Based on a Whale Optimization Algorithm." *Ocean Engineering* 250: 111070. https://doi.org/10.1016/j.oceaneng.2022.111070.

Yang, L., P. Li, S. Qian, et al. 2023. "Path Planning Technique for Mobile Robots: A Review." *Machines* 11, no. 10: 980. https://doi.org/10.3390/machines11100980.

Yang, X., and A. Hossein Gandomi. 2012. "Bat Algorithm: A Novel Approach for Global Engineering Optimization." *Engineering Computations* 29, no. 5: 464–483. https://doi.org/10.1108/02644401211235834.

Yang, X.-S. 2010. "A New Metaheuristic Bat-Inspired Algorithm." In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, edited by J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, 65–74. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6.

Yao, J., and H. Deng. 2023. "Fiber Raman Amplifier Based on Improved Whale Algorithm." *Journal of Physics: Conference Series* 2525, no. 1: 012016. https://doi.org/10.1088/1742-6596/2525/1/012016.

Yu, S., J. Zhu, and C. Lv. 2023. "A Quantum Annealing Bat Algorithm for Node Localization in Wireless Sensor Networks." *Sensors* 23, no. 2: 782. https://doi.org/10.3390/s23020782.

Zellagui, M., N. Belbachir, and R. A. El-Sehiemy. 2023. "Solving the Optimal Power Flow Problem in Power Systems Using the Mountain Gazelle Algorithm." *Engineering Proceedings* 56, no. 1: 176. https://doi.org/10.3390/ASEC2023-16269.

Zhang, R., R. Chai, S. Chai, Y. Xia, and A. Tsourdos. 2024. "Design and Practical Implementation of a High Efficiency Two-Layer Trajectory Planning Method for Agv." *IEEE Transactions on Industrial Electronics* 71, no. 2: 1811–1822. https://doi.org/10.1109/TIE.2023.3250847.

Zhang, Y., M. C. Fontaine, V. Bhatt, S. Nikolaidis, and J. Li. 2023. "Multi-Robot Coordination and Layout Design for Automated Warehousing." In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence IJCAI-23*, edited by E. Elkind, 5503–5511. Macao, China: International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2023/611.

Ziadi, S., M. Njah, and S. Charfi. 2020. "Which Is Better for Mobile Robot Trajectory Optimization: PSO or Ga?" In *New Trends in Robot Control*, edited by J. Ghommam, N. Derbel, and Q. Zhu, vol. 270, 287–300. Singapore: Springer. https://doi.org/10.1007/978-981-15-1819-5_13.