Journal of Computers Vol. 37 No. 5, October 2026, pp. 13-25
https://doi.org/10.63367/19911599202610370500 2

Research on Warehouse Path Planning for Logistics Robots Based on Improved Whale Optimization Algorithm

Shizheng Liu* and Xinnan Ji

Zhejiang Industry Polytechnic College, Shaoxing 312000, Zhejiang, China
zjliushizheng@yeah.net, zjsxjxn2025@126.net

Received 1 May 2025; Revised 25 July 2025; Accepted 31 July 2025

Abstract. How to efficiently and reasonably carry out path planning has always been a hot spot of research in the field of warehouse logistics robots. We propose a logistics robot warehouse path planning strategy based on the Improved Whale Optimisation Algorithm (IWOA). The whale optimisation algorithm's convergence speed is slow, making it easy to fall into the limitations of the local optimum. First, we use Logistic Chaos-based population initialisation to ensure the diversity of algorithmic individuals. Second, we use adaptive weights in the whale bubble behaviour, which can effectively avoid the algorithm from falling into the local optimum. Finally, we use the Bresenham straight line method in the warehouse path movement. Simulation experiments, the algorithm of this paper and Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), Whale Optimisation Algorithm (WOA) in different benchmark functions for comparison, the results show that the algorithm has a better effect, in the 20 * 20 unit area in the simulation of four different warehouse scenarios, the results show that the algorithm of this paper has a clear advantage over the ACO, PSO advantage, and compared to the WOA algorithm path length reduced by 1.94%, which shows that the The algorithm can make the robot adapt to the more congested warehouse scenarios.

Keywords: whale optimization algorithm, population initialization, adaptive weighting, path planning

1 Introduction

With the development of information and industrial automation technology, many types of robots have achieved good applications in different fields such as industry, agriculture, and medicine [1]. Logistics robots have become an important part of the current logistics and warehouse automation technology, which can improve the efficiency of warehouse operations, optimize resources and reduce costs. In large-scale logistics warehouses, logistics robots carry out operations such as handling, storage, and pickup of goods by accepting commands, in which reasonable path planning is one of the important judgment standards of logistics robot performance indicators. Therefore, how to plan a reasonable path planning is particularly important [2]. In recent years, some scholars have used meta-heuristic algorithms for path planning to achieve better results, but the meta-heuristic algorithm's own performance has a certain impact on path planning [3]. Based on this, we combine the whale optimization algorithm with the logistics robot moving path to reduce the length of the robot moving path by using the whale optimization algorithm. Whale optimization algorithm [4] is a meta-heuristic algorithm that has been widely used in recent years, which has the advantages of simple structure and easy implementation of the algorithm, but at the same time, it has the shortcomings of slow convergence and easy to fall into the local optimum. In order to use this algorithm to get better results in the path planning of mobile robots, we propose an improved whale optimization algorithm-IWOA algorithm, whose main idea is to use Logistic chaos-based ideas in the initialization of the population, adaptive weights in the bubble behavior of whales, and Bresenham straight-line method in the unit path movement of warehouses, and through simulation Experiments verify that the algorithm has a better moving effect in different congested scenarios.

This paper is structured as follows: Section 2 summarizes and categorises recent research findings in path planning. Section 3 describes the basic whale optimisation algorithm and the Bresenham line algorithm. Section 4 discusses the updated whale optimisation algorithm. Section 5 evaluates the efficiency of the enhanced whale optimisation method and runs simulations on pathways in various challenging circumstances to illustrate the algorithm's usefulness. Section 6 summarizes the entire paper and suggests potential future research directions.

* Corresponding Author

2 Path Planning Related Research

With the acceleration of global economic integration and the rapid development of e-commerce, the logistics industry is undergoing unprecedented transformation. As the core driving force behind the automation and intelligent transformation of logistics, logistics robots—particularly automated guided vehicles (AGVs) and autonomous mobile robots (AMRs)—are playing an increasingly important role in warehousing, sorting, handling, and distribution operations. The efficient operation of these robots directly impacts logistics efficiency, cost control, and customer satisfaction. In the complex and dynamic logistics environment, planning the optimal, safest, and most economical paths for logistics robots is a key challenge in achieving their autonomous and intelligent operation. This not only involves obstacle avoidance and path optimization for individual robots in static or dynamic obstacle environments but also encompasses complex issues such as task allocation, conflict avoidance, and global path optimization during multi-robot collaborative operations. Therefore, conducting in-depth research and a systematic review of logistics robot path planning technology holds significant theoretical significance and practical application value for advancing the development of intelligent logistics systems.

2.1 Traditional Path Planning Algorithms

Traditional path planning algorithms primarily rely on precise modeling of environmental information and effective exploration of the search space. Algorithm A* is an improved version of Dijkstra's algorithm, which introduces a heuristic function to guide the search direction, thereby significantly enhancing search efficiency. The author [5] proposes an A* algorithm based on an adaptive cost function to improve robot path planning efficiency. The author [6] investigates path planning based on the A* algorithm from the perspective of time efficiency, and simulation results demonstrate that this algorithm indeed enhances path planning effectiveness. The Dynamic algorithm is a path planning algorithm suitable for dynamic environments, capable of real-time path updates during robot movement to address unknown obstacles or changes in target positions within the environment. The author [7] proposes a robot motion path planning algorithm based on dynamic path planning. Simulation experiments demonstrate that this method can effectively avoid suddenly appearing obstacles and reduce the obstacles encountered by the robot. The author [8] proposes a dynamic path planning method for mobile automata with limited environmental information, which obtains local information about its surrounding environment through "perception" feedback. The RRT algorithm is a sampling-based path planning algorithm that explores traversable spaces by randomly sampling points in the environment and gradually constructing a search tree. The author [9] proposes an RRT algorithm based on an exploration strategy, and simulation experiments demonstrate that this strategy significantly reduces the path generation time for robots. The author [10] proposes an improved RRT algorithm for robot path planning, and simulation experiments validate the algorithm's advantages in terms of time and path length during path planning.

Traditional path planning algorithms rely on accurate modeling of the environment and effective exploration of the search space. For example, the A* algorithm improves search efficiency through heuristic functions. while variants of A* that incorporate adaptive cost functions and time efficiency optimizations can improve path planning performance, they still face limitations such as high computational complexity and insufficient real-time capability in complex dynamic environments, especially when sudden obstacles appear in the environment. Although dynamic path planning algorithms can update paths in real time, their response speed to unknown obstacles and path optimization capabilities are still constrained by the completeness of environmental information acquisition; The RRT algorithm constructs a search tree through random sampling. Although the improved strategy reduces path generation time, randomness makes it difficult to guarantee path optimality, and search efficiency significantly decreases in high-dimensional spaces, exposing the inherent shortcomings of traditional algorithms in terms of weak adaptability to dynamic environments and stringent modeling requirements.

2.2 Based on Meta-heuristic Algorithms

Meta-heuristic algorithms are an effective method for solving NP problems, and are used to optimize factors such as time and cost in path planning. The author [11] proposed a path planning strategy based on genetic algorithms, which uses genetic algorithms to solve time and cost problems in path planning. Simulation experiments showed that this algorithm has good planning results. The author [12] introduces the particle swarm algorithm into path

planning for mobile robots, enhancing algorithm performance through a two-population strategy. Simulation experiments demonstrate that the improved algorithm effectively reduces path length. The author [13] proposes an evolutionary method for robot route planning based on linear sorting likelihood. Simulation results show that compared with traditional genetic algorithms, this algorithm significantly reduces the movement time of robots. The author [14] proposes a strategy of incorporating ant colony algorithms and frog leap algorithms into path planning, leveraging the advantages of both algorithms to enhance the effectiveness of robot path planning. The author [15] and [16] respectively introduce dynamic windows into particle swarm algorithms and ant colony algorithms for path planning, thereby improving the path planning effectiveness of mobile robots and reducing the total path length. The author [17] proposes an enhanced algorithm for global path planning based on differential wheeled mobile robots, which integrates the whale optimization algorithm with the particle swarm algorithm. Simulation experiments demonstrate that this algorithm can reduce path length by up to 19.3%. The author [18] proposes the application of the ant colony algorithm for 3D path planning in logistics drones. Simulation results demonstrate that the algorithm's positioning performance exhibits better practicality. The author [19] proposes the integration of the artificial bee colony algorithm and the genetic algorithm for robot path planning. Simulation results indicate that the algorithm achieves better path planning performance. The application of meta-heuristic algorithms proposed in the reference [20-21] also yields better results in dynamic path planning for UAVs.

Heuristic algorithms, as an effective means of solving NP problems, demonstrate significant advantages in path planning. They do not require precise mathematical modeling and instead design search strategies by simulating natural phenomena, thereby balancing global exploration and local optimization. For example, genetic algorithms optimize path time and cost through selection, crossover, and mutation operations. Improved algorithms based on linear sorting probabilities effectively reduce robot movement time compared to traditional genetic algorithms. Particle swarm algorithms, when combined with a dual-population strategy, significantly shorten path length. Hybrid strategies such as the integration of ant colony algorithms with frog leap algorithms, or the combination of whale optimization algorithms with particle swarm algorithms, further enhance planning efficiency, with notable reductions in path length. These algorithms not only handle multi-objective optimization problems (such as simultaneously considering path length, energy consumption, and safety) but also feature flexible parameter adjustment and ease of integration with domain knowledge. For instance, introducing dynamic windows in particle swarm and ant colony algorithms effectively reduces total path length, making them more adaptable to complex scenarios in warehouse logistics, especially when handling high-dimensional, nonlinear optimization problems. Through iterative search and maintenance of population diversity, the risk of getting stuck in local optima is reduced.

2.3 Artificial Intelligence Technology

With the fast advancement of AI technologies, especially the rise of machine learning and deep learning, new ideas and solutions have emerged for logistics robot path planning. In terms of SVM, the author [22] proposes a new algorithm for path planning in unstructured environments based on multi-class support vector machines (MSVM). The author [23] proposes a novel path planning method based on Support Vector Machines (SVM) and the Optimal Accessible Path with Course Correction (LAP-CC). Simulation results demonstrate its advantages in terms of simplicity, feasibility, low computational cost, and good repeatability. In the context of Convolutional Neural Networks (CNN), The author [24] proposes a path generated by optimizing the Theta* algorithm using a Convolutional Neural Network. Experimental results show that this algorithm can generate safe and reliable paths with high success rates in extremely short computation times. The author [25] employs an improved CNN model for path planning, with simulation results indicating that this model achieves a significant improvement in path planning efficiency compared to the CNN model. In the RNN domain, the author [26] proposes the use of RNN models for robot path planning, with simulation results demonstrating the effectiveness of this method. The author [27] proposes an RNN-based system for tracking the behavior of mobile robots, which significantly improves the path planning performance of robots. In terms of LSTM, the author [28] proposes a real-time UAV path planning algorithm based on the Long Short-Term Memory (LSTM) network, which combines the memory characteristics of recurrent neural networks (RNN) with deep reinforcement learning algorithms. Simulation results demonstrate that this algorithm exhibits significantly enhanced robustness and accuracy when dealing with more complex environments. The author [29] describes a path planning model (SAC-LSTM) that combines a short-term and long-term memory network (LSTM) with a soft actor-critic (SAC) algorithm to improve the algorithm's ability to capture historical information using the LSTM's memory property. To address the issue

of memory degradation caused by resetting the hidden state of an LSTM to zero during training, the model implements a warm-up training technique to optimise performance. Simulation experiments demonstrate that the model can not only reduce path planning time, but also produce more concise and efficient planning paths. The author [30] proposes a hybrid algorithm combining Q-Learning with FPA to address the slow convergence issue of traditional Q-Learning algorithms. The author [31] integrates heuristic search strategies and simulated annealing mechanisms into the Q-Learning algorithm to improve the path planning performance of mobile robots.

The advancement of artificial intelligence technology opens up new possibilities for path planning, and various machine learning and deep learning approaches, such as SVM, CNN, and RNN/LSTM, are commonly employed. Algorithms based on MSVM and SVM demonstrate simple and feasible advantages in unstructured environments, but they rely on the quality of sample data; CNNs can improve planning efficiency by optimizing paths or enhancing models, but they require significant computational resources and lack interpretability; RNNs and LSTMs combined with reinforcement learning enhance robustness in complex environments, but their training processes are complex and real-time performance struggles to meet dynamic scenario demands. Fusion algorithms integrating deep learning with Q-Learning improve convergence speed, but they feature high model complexity, substantial deployment costs, and discrepancies between simulation and real-world scenarios.

Through the above research, we have found that traditional algorithms are no longer able to meet the requirements of current mobile robot path planning. Although the emergence of artificial intelligence technology can optimize path planning, such technologies have high requirements for sample data or model performance, leading to high training requirements and increased hardware costs. Therefore, using meta-heuristic algorithms for robot path planning is currently a more reasonable approach.

3 Related Algorithms

3.1 WOA

Mirjalili and Lewis proposed the WOA algorithm, which optimises swarm intelligence. The algorithm takes whales in the ocean as its research object, combining the living habits of whale populations, and divides the algorithm into three stages: encirclement and predation, group attack, and food search. The specific process of the algorithm has been detailed in Reference [4], and we provide a brief description here. In the algorithm, $X_i = (X_i^1, X_i^2, ..., X_i^D)$, $i = 1, 2, \ldots, N$ represents the position of the ith whale in a D-dimensional space.

(1) Surrounding Predation

The primary goal of this phase is to conduct predation by defining the presence of prey at the optimal location of the current individual, which can guide other surrounding whales to the current individual's location, and the mathematical model is expressed as follows:

$$X(t+1) = X_p(t) - A \bullet \left| C \bullet X_p(t) - X(t) \right| \tag{1}$$

where $X_p = (X_p^1, X_p^2, \ldots X_p^D)$ is the prey location, $A \cdot |C.X_p(t) - X(t)|$ is the bracketing step, t is the current number of iterations, and the mathematical formulae for C and A are given below.

$$C = 2 \bullet rand_2 \tag{2}$$

$$A = 2a \bullet rand_1 - a \tag{3}$$

In the above formulations, $rand_1$ and $rand_2$ represent random integers between [0,1], respectively, and a represents the convergence factor, which varies between [2,0] with a decreasing trend, and its expression is as follows:

$$a = 2 - 2t / t_{\max} \tag{4}$$

where t and $t_{\max}$ are represent the current number of running iterations and the maximum number of iterations in the algorithm

(2) Bubble attack

Whales in the water utilise their own spray to generate massive bubbles around their prey, and each individual employs a spiral ascent to update its position. The mathematical model is described as follows:

$$X\left(t+1\right)=\vec{D}\times e^{lb}\times\cos\left(2\pi l\right)+X_p\left(t\right) \tag{5}$$

In the formula, where $\vec{D}=\left|X_p\left(t\right)-X\left(t\right)\right|$ is the distance between the individual whale and the current target prey, b is the integer number of the spiral shape exhibited when the whale attacks, and l is a random value ranging from -1 to 1. Thus, the previously mentioned shrink-wrap and the present bubble assault become the whale algorithm's two major methods, with the chance of each set to 0.5 for fair selection.

(3) Food-seeking stage

During the phase of seeking for new food, the algorithm instructs each individual whale to hunt for nearby food in a randomised manner, and the mathematical model is described as follows:

$$X\left(t+1\right)=X_{rand}\left(t\right)-A\bullet\left|C\bullet X_p\left(t\right)-X\left(t\right)\right| \tag{6}$$

X_{rand} represents the position of each particular whale.

3.2 Bresenham Straight Line Algorithm

The Bresenham Straight Line Algorithm is a classic algorithm for drawing straight lines in computer graphics, centred on the idea of determining the pixel points on a line through a series of decisions in order to approximate the ideal line as accurately as possible.

The core of the algorithm is the use of integer arithmetic to avoid floating-point arithmetic and thus increase computational efficiency. It is based on the principle that when drawing a straight line, for each x-coordinate, the corresponding y-coordinate needs to be determined. By comparing the position of the line with the current pixel point, it is decided whether to choose the current pixel point or its neighbouring pixel points as the points on the line. Main Idea: Suppose that a straight line is to be drawn from point (x_0, y_0) to point (x_1, y_1). The slope of the line is $k=\dfrac{y_1-y_0}{x_1-x_0}$. In Bresenham's straight line algorithm, instead of calculating the slope directly, a decision variable is used to determine the location of the next pixel point. Let the decision variable be d, with initial value $d = 2 \times \Delta y - \Delta x$, where $\Delta x = x_1 - x_0$, $\Delta y = y_1 - y_0$. During the drawing process, starting from the starting point, advance one unit at a time in the x-direction, and then decide the movement in the y-direction according to the value of the decision variable d: If $d \geq 0$, then select the pixel point to the upper right of the current pixel point, i.e., add 1 to the x-coordinate and 1 to the y-coordinate, and also update the decision variable $d = d + 2 \times \Delta y - \Delta x$. Otherwise, the pixel point to the right of the current pixel point is selected, i.e., the x coordinate is increased by 1, the y coordinate is unchanged, and the decision variable $d = d + 2 \times \Delta y$ is updated.

4 Improved Whale Optimization Algorithm

Most meta-heuristic algorithms have the disadvantage of easily falling into the local optimum, facing the high-dimensional problem algorithm's global search ability is weak; the whale optimisation algorithm has the same problem; in order to solve the slow convergence speed of the algorithm and further improve the accuracy of the algorithm, we intend to improve the algorithm from the initialisation of the population and the adaptive weights to improve the algorithm's performance.

(1) Population initialization

For meta-heuristic algorithms, good or bad population initialisation has a direct impact on the algorithm's

solution correctness and convergence rate [32]. In the current whale algorithm, the lack of initialisation of the individual population affects the algorithm's performance; to avoid this situation, we take advantage of the randomness and traversal of the chaotic idea, and adopt the Logistic-based chaotic mapping for population initialisation; the expression is as follows:

$$x_{n+1} = \mu \times x_n \times \left(1 - x_n\right) \tag{7}$$

$$x' = \begin{cases} x_n f\left(x_n\right) > f\left(x'_{n+1}\right) \\ x'_{n+1} f\left(x_n\right) < f\left(x'_{n+1}\right) \end{cases} \tag{8}$$

In Eq. (7), new population individuals are generated by logistic chaos. μ is 4.

(2) Adaptive weighting

Literature [33] shows that adaptive weighting in metaheuristic algorithms can effectively improve the performance of the algorithm. In the WOA algorithm initial stage, the weights are large, and the algorithm tends to explore globally, helping the whale's candidate solutions to approach the potential optimal region quickly and avoid premature convergence. In the later stages, the weights are gradually reduced, and the algorithm focuses on local exploitation, finely adjusting the position of the current solution so that it gradually converges to the vicinity of the global optimal solution. The values of the weight expressions are as follows:

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right) \times \left(1 - \frac{iter}{Max_{iter}}\right) \tag{9}$$

where $w_{\max}$ is 0.9, $w_{\min}$ is 0.2, *iter* is the current number of iterations, and Max_{iter} is the maximum number of iterations.

The flowchart of the modified method is shown in Fig. 1:

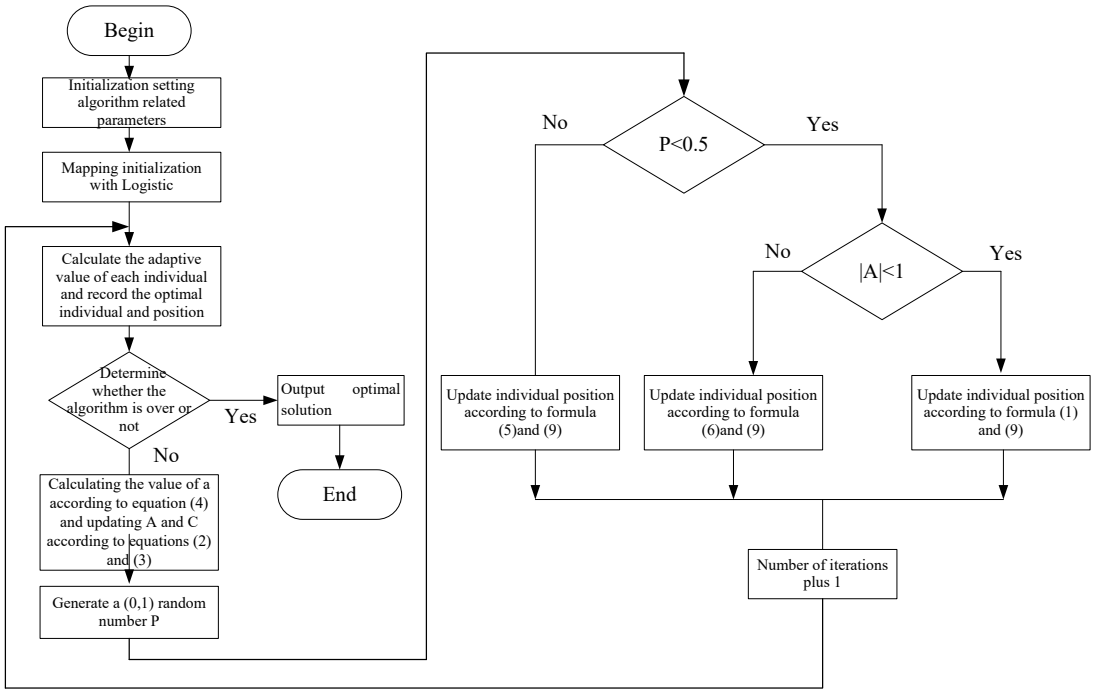


Fig. 1. Flow of the improved algorithm

(3) Algorithmic complexity

The time complexity and space complexity of the improved whale optimization algorithm (which uses a logistic chaotic map to initialize the population and introduces an adaptive weighting strategy) are essentially the same as those of the traditional whale optimization algorithm (WOA), but there are differences in the computational details of some steps. From the perspective of time complexity, the core computations of the traditional WOA are concentrated in the population iteration process, where each iteration involves operations such as surrounding the prey, spiral bubble attacks, and random searches, resulting in an overall complexity of O(N×T) (where N is the population size and T is the maximum number of iterations). When introducing the Logistic chaotic map for population initialization, each individual must generate a chaotic sequence and map it to the solution space, with a computational complexity of O(N), which is linear complexity and has a limited impact on the overall time cost; the adaptive weighting strategy typically dynamically adjusts parameters based on the number of iterations or the quality of the current solution, with a computational complexity of O(1) per iteration, with cumulative complexity of O(T), which is compatible with the original algorithm's iterative structure. Therefore, the time complexity of the improved algorithm remains dominated by O(N×T), without introducing higher-order complexity due to the optimization strategy. From the perspective of space complexity, the algorithm requires storing population positions, fitness values, and related parameters, with space occupancy being O(N). Chaotic sequence generation and weight parameters are scalar computations, which do not add additional spatial dimensions, so the space complexity remains O(N). Overall, the whale optimization algorithm based on chaotic initialization and adaptive weights has the same computational complexity as traditional algorithms, with only minor differences in constant factors due to operational details. However, this does not alter the algorithm's complexity order of magnitude. This allows the improved algorithm to maintain high efficiency while enhancing search performance through strategy optimization.

The time complexity and space complexity of the improved whale optimization algorithm are essentially the same as those of the traditional whale optimization algorithm (WOA), but there are differences in the computational details of some parts. From the perspective of time complexity, the core computations of the traditional WOA are concentrated in the population iteration process, where each iteration involves operations such as surrounding the prey, spiral bubble attacks, and random searches, resulting in an overall complexity of O(N×T) (where N is the population size and T is the maximum number of iterations). When introducing the Logistic chaotic map for population initialization, each individual must generate a chaotic sequence and map it to the solution space, with a computational complexity of O(N), which is linear complexity and has a limited impact on the overall time cost; the adaptive weighting strategy typically dynamically adjusts parameters based on the number of iterations or the quality of the current solution, with a computational complexity of O(1) per iteration, with cumulative complexity of O(T), which is compatible with the original algorithm's iterative structure. Therefore, the time complexity of the improved algorithm remains dominated by O(N×T), without introducing higher-order complexity due to the optimization strategy. From the perspective of space complexity, the algorithm requires storing population positions, fitness values, and related parameters, with space occupancy being O(N). Chaotic sequence generation and weight parameters are scalar computations, which do not add additional spatial dimensions, so the space complexity remains O(N). Overall, the whale optimization algorithm based on chaotic initialization and adaptive weights has the same computational complexity as traditional algorithms, with only minor differences in constant factors due to operational details. However, this does not alter the algorithm's complexity order, enabling the improved algorithm to maintain high efficiency while enhancing search performance through strategy optimization.

5 Simulation Experiment

To demonstrate the effectiveness of the algorithm in this paper, we conducted a two-part study: one to verify the algorithm's performance in this paper, and the second to verify the algorithm's effectiveness in warehouse path planning.

5.1 Verify the Performance of This Paper's Algorithm

The algorithm of this paper is compared with ACO, PSO and WOA algorithms by selecting 5 benchmark test functions in 3 dimensions (5 dimensions, 10 dimensions and 30 dimensions) as shown in Table 1, and selecting

the maximum, minimum, average and standard values as the metrics. The comparison results are shown in Table 2. When testing the performance of the improved whale optimization algorithm, Step, sphere, Ackley, schwefel2.22, Rosenbrock and other benchmark functions are used because they can comprehensively evaluate the search ability and robustness of the algorithm. Among them, the sphere function is a single-peak convex function, which can test the local optimization accuracy of the algorithm; the step function is a step function, which can test the algorithm's ability to deal with the discrete space; the Ackley function contains multiple local optimal solutions, which can verify the algorithm's ability to search for the global optimization by jumping out of the local extremes; the schwefel2.22 function combines the characteristics of the single-peak and the multiple-peak to assess the algorithm's balanced search The schwefel2.22 function combines single-peak and multi-peak characteristics, which can evaluate the balance search ability of the algorithm; the Rosenbrock function is a pathological single-peak function, which requires high convergence speed and stability of the algorithm. The Rosenbrock function is a pathological single-peak function, which requires high convergence speed and stability of the algorithm. These functions comprehensively evaluate the performance of the algorithm in different dimensions (single-peak/multi-peak, continuous/discrete, convex/non-convex), which ensures that the improved algorithm is universally applicable in complex optimization problems.

Table 1. Table of 5 benchmark functions

Function name	Benchmark function
F1	Step
F2	Sphere
F3	Ackley
F4	Schwefel2.22
F5	Rosenbrock

Table 2. Algorithm performance results

Function name	Dim	Algorithm	Max	Min	Average	Standard
F1	5	ACO	3.2500e+02	1.2500e+02	2.4500e+02	6.2767e+01
		PSO	0	0	0	0
		WOA	0	0	0	0
		IWOA	0	0	0	0
	10	ACO	8.3700e+02	7.0200e+02	7.8777e+02	3.7220e+01
		PSO	0	0	0	0
		WOA	0	0	0	0
		IWOA	0	0	0	0
	30	ACO	2.8710e+03	2.7970e+03	2.8350e+03	1.8862e+01
		PSO	2.1000e+01	0	4.8667e+00	4.6218e+00
		WOA	0	0	0	0
		IWOA	0	0	0	0
F2	5	ACO	3.2500e+02	7.5000e+01	2.4083e+02	7.3544e+01
		PSO	1.9966e-39	4.2241e-43	2.8196e-40	4.3160e-40
		WOA	2.9175e-105	5.2461e-126	9.9740e-107	5.3232e-106
		IWOA	3.2393e-202	4.6492e-214	1.2099e-203	0
	10	ACO	8.4198e+02	6.9383e+02	7.8568e+02	4.2234e+01
		PSO	8.5886e-27	1.4963e-29	8.6722e-28	1.7723e-27
		WOA	3.6048e-97	6.7194e-119	1.3216e-98	6.5725e-98
		IWOA	1.0665e-199	7.2122e-209	5.4081e-201	0
	30	ACO	2.8763e+03	2.8107e+03	2.8525e+03	1.3228e+01
		PSO	1.3653e-03	2.1506e-09	4.6371e-05	2.4912e-04
		WOA	5.7214e-93	5.5414e-108	3.6181e-94	1.2643e-93
		IWOA	1.3478e-196	3.1950e-208	4.9902e-198	0
F3	5	ACO	1.6012e+01	9.3743e+00	1.4352e+01	1.5507e+00
		PSO	3.9968e-15	4.4409e-16	2.8126e-15	1.7034e-15
		WOA	3.9968e-15	4.4409e-16	1.5099e-15	1.6559e-15
		IWOA	3.9968e-15	4.4409e-16	1.7468e-15	1.7413e-15
	10	ACO	1.7067e+01	1.6790e+01	1.7040e+01	8.4538e-02
		PSO	1.1551e+00	7.5495e-15	3.8505e-02	2.1090e-01
		WOA	2.8147e+00	4.4409e-16	9.3823e-02	5.1389e-01
		IWOA	3.9968e-15	4.4409e-16	1.9836e-15	1.7906e-15
	30	ACO	1.7265e+01	1.7201e+01	1.7224e+01	1.9150e-02
		PSO	8.0687e+00	1.1551e+00	3.1753e+00	2.0712e+00
		WOA	7.5495e-15	4.4409e-16	2.3389e-15	2.0298e-15
		IWOA	3.9968e-15	4.4409e-16	2.4573e-15	1.7906e-15

F4	5	ACO	3.5000e+01	1.5000e+01	2.7000e+01	5.5086e+00
		PSO	1.9554e-20	7.6739e-22	6.5509e-21	5.7431e-21
		WOA	7.1351e-69	6.1874e-78	3.2260e-70	1.3255e-69
		IWOA	1.2822e-102	2.1804e-107	1.2705e-103	2.7252e-103
	10	ACO	8.6420e+08	4.1152e+07	4.2830e+08	2.3588e+08
		PSO	2.7309e-13	2.2850e-15	4.6974e-14	6.8975e-14
		WOA	3.0916e-64	1.6138e-71	1.2502e-65	5.6225e-65
		IWOA	6.1377e-98	1.8482e-105	2.0620e-99	1.1203e-98
	30	ACO	1.6052e+29	2.5462e+28	5.0580e+28	4.1495e+28
		PSO	7.0016e+01	9.1903e-03	1.7289e+01	1.5291e+01
		WOA	5.4208e-62	4.5101e-73	5.7694e-63	1.4368e-62
		IWOA	1.8294e-98	8.3834e-105	1.3293e-99	3.8907e-99
F5	5	ACO	2.3578e+06	7.5039e+04	1.3673e+06	7.3240e+05
		PSO	1.0022e+01	3.8840e-03	1.0620e+00	2.2784e+00
		WOA	5.3819e+00	9.3311e-03	1.4707e+00	1.5935e+00
		IWOA	5.6011e+00	7.8339e-08	1.0367e+00	1.3693e+00
	10	ACO	8.1130e+06	6.4993e+06	7.5269e+06	3.9487e+05
		PSO	1.0002e+04	3.8278e-02	3.8959e+02	1.8252e+03
		WOA	8.7249e+00	5.8352e+00	7.4299e+00	6.5660e-01
		IWOA	8.0002e+00	6.1081e+00	7.2905e+00	4.5470e-01
	30	ACO	3.2906e+07	3.1634e+07	3.2398e+07	2.8250e+05
		PSO	1.0074e+04	1.3411e+00	7.6000e+02	2.5327e+03
		WOA	2.8749e+01	2.7384e+01	2.8063e+01	4.0271e-01
		IWOA	2.8717e+01	2.6951e+01	2.8142e+01	4.1130e-01

From the above experimental results, it can be seen that IWOA shows significant superiority in the optimisation tasks of several benchmark functions, especially in high-dimensional complex problems, where its convergence accuracy, stability and robustness are better than those of ACO, PSO and WOA.

In the Step function test, IWOA shows extremely good optimisation performance. In the 5- and 10-dimensional cases, IWOA and PSO and WOA can perfectly converge to the theoretical optimal value (0.0000e+00), showing excellent adaptability to discrete problems. However, under the 30-dimensional high-dimensional condition, IWOA still maintains the perfect optimisation (0.0000e+00), while PSO starts to show slight fluctuations (maximum value of 2.1000e+01, mean value of 4.8667e+00), which indicates that PSO starts to show limitations in the optimisation of the Step function in higher dimensions. In contrast, IWOA is able to maintain absolute optimisation accuracy when the dimension grows through the improved search mechanism, fully demonstrating the superiority of its algorithm design.

In the Sphere function test, IWOA shows extremely low optimisation errors in all dimensions, with the maximum, minimum and average values close to the theoretical optimal value of 0 and a standard deviation of 0, indicating that the algorithm has extremely high convergence accuracy and stability. In contrast, the optimisation error of ACO increases significantly in high dimensions (e.g., the mean value reaches 2.8525e+03 at 30D), while PSO and WOA perform better, but the mean values of PSO (4.6371e-05) and WOA (3.6181e-94) are still much higher than that of IWOA (4.9902e-198) at 30D. This indicates that IWOA has an overwhelming advantage in global search ability, and can effectively avoid premature convergence and approach the theoretical optimal solution.

In the Ackley function test, IWOA also shows excellent performance. In the 5- and 10-dimensional cases, the optimisation results of IWOA are comparable to those of PSO and WOA, which are both close to the theoretical minimum (4.4409e-16). However, in the 30-dimensional high-dimensional condition, the mean value of IWOA (2.4573e-15) is significantly better than that of PSO (3.1753e+00) and WOA (2.3389e-15), and its maximum and minimum values are stable at a very low level, which indicates that the IWOA still maintains a strong robustness in high-dimensional complex environments. In contrast, PSO shows large fluctuations in 30 dimensions (the maximum value reaches 8.0687e+00), indicating that it is prone to fall into local optimality, which is effectively circumvented by IWOA through the improved search strategy.

For the Schwefel2.22 function, the advantage of IWOA is even more obvious. In the 10- and 30-dimensional tests, the optimisation result of ACO fails completely (the mean value reaches 4.2830e+08 and 5.0580e+28, respectively), while PSO and WOA can achieve lower errors, but the optimisation accuracy of IWOA is higher (e.g., the mean value of 1.3293e-99 at 30D is far superior to that of PSO's 1.7289e+01 and WOA's 5.7694e-63). It is worth noting that the standard deviation of IWOA remains close to 0 in the high-dimensional case, indicating that its optimisation process is highly stable and unaffected by dimensional growth.

IWOA also performs well on complex problems such as the Rosenbrock function with a narrow global optimal region. Despite the complete failure of ACO (30-dimensional mean up to 3.2398e+07) and the large fluctuations in the optimisation results of PSO and WOA (e.g., the mean value of PSO at 30 dimensions is 7.6000e+02,

and that of WOA is 2.8063e+01), the optimisation results of IWOA are closer to the theoretical optimum (30-dimensional mean of 2.8142e+01), and its standard deviation (4.1130e-01) is significantly lower than that of PSO and WOA, indicating that its search strategy can balance global exploration and local exploitation more effectively.

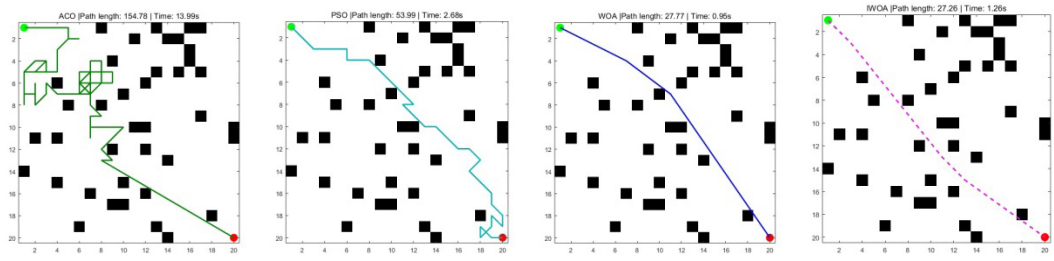
In summary, IWOA significantly outperforms the comparison algorithms in terms of convergence accuracy, stability and high-dimensional adaptability. Its core advantages are (1) Improved search mechanism: by introducing adaptive weights, it enhances the ability to explore the complex solution space and avoids premature convergence. (2) Robustness: it can still maintain a very low standard deviation in high-dimensional problems, indicating that the algorithm is insensitive to initial conditions and parameter settings. (3) Global optimisation ability: it performs well in complex functions such as Schwefel2.22 and Rosenbrock, proving that it can effectively handle multi-peak and non-linear problems. These features make IWOA particularly suitable for high-dimensional optimisation tasks, such as engineering design, machine learning and neural network tuning parameterisation, etc. Its performance far exceeds that of traditional optimisation algorithms and shows a broad practical prospect.

5.2 Path Planning Test

In order to better illustrate the path planning performance of our proposed algorithm, we simulated four different warehouse scenarios as shown in Fig. 2 to Fig. 5, in which an area of 20*20 unit length is used, and simulated from the starting position (upper left) to the end position (lower right) to these four scenarios set the environment from simple to complex, and different path lengths are calculated by the four algorithms (shown in Table 3). From the results in the table, four different scenes, the path length of this paper's algorithm is the shortest, which shows the effect of this paper's algorithm has, especially in the scene in Fig. 3, this paper's algorithm's time are the shortest, although the time of this paper's algorithm is not the optimal one in Scene 1, Scene 2 and Scene 4, but it is almost with the WOA algorithm is not much difference, from a practical point of view, the logistic robots handle the goods is relatively critical in terms of path length.

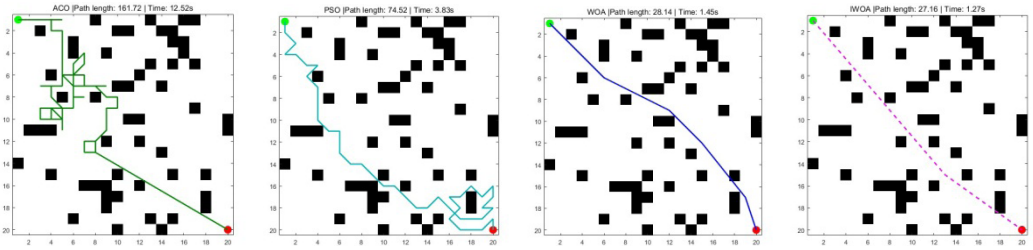
Table 3. Comparison results of different scenarios

Warehouse setting	ACO		PSO		WOA		IWOA	
	Length	Times (S)	Length	Times (S)	Length	Times (S)	Length	Times (S)
Warehouse setting 1	154.78	13.99	53.99	2.68	27.22	0.96	27.26	1.26
Warehouse setting 2	161.72	12.52	74.52	3.83	28.14	1.45	27.16	1.27
Warehouse setting 3	171.58	13.88	67.65	3.27	30.91	1.51	29.76	1.79
Warehouse setting 4	5160.05	11.96	49.39	4.31	31.74	1.88	31.58	2.21



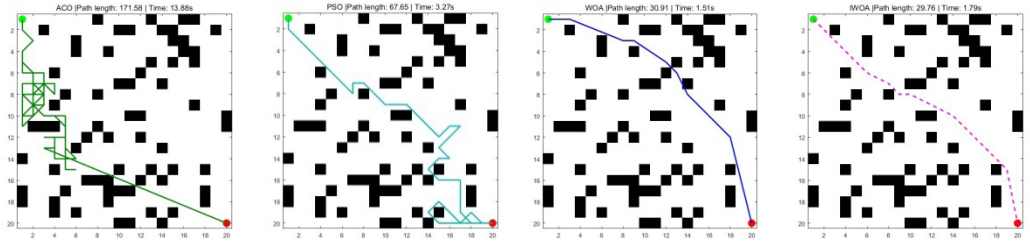
(a) ACO (b) PSO (c) WOA (d) IWOA

Fig. 2. Warehouse setting 1



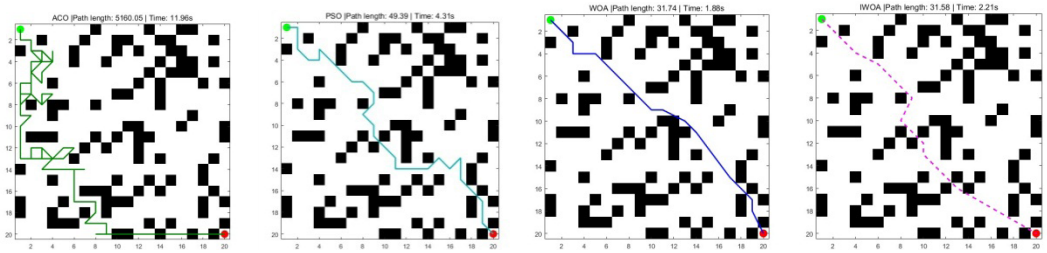
(a) ACO (b) PSO (c) WOA (d) IWOA

Fig. 3. Warehouse setting 2



(a) ACO (b) PSO (c) WOA (d) IWOA

Fig. 4. Warehouse setting 3



(a) ACO (b) PSO (c) WOA (d) IWOA

Fig. 5. Warehouse setting 4

6 Conclusion

In this paper, we propose a path planning algorithm for warehouse mobile robots based on an improved whale optimization algorithm. Simulation results demonstrate that the algorithm performs well across various path configurations, particularly excelling in planning the shortest path distance in more complex warehouse environments. Moving forward, we aim to incorporate the LSTM model into path planning. While the WOA possesses global search capabilities, enabling it to explore optimal paths in the solution space by simulating whale hunting behavior, it is particularly adept at handling nonlinear constraints and multi-objective optimization problems in path planning (such as minimizing path length and obstacle avoidance costs). However, in dynamic environments, real-time changes in environmental information may cause delays in search strategy adjustments, and a single algorithm may easily become trapped in local optima. By introducing an LSTM model, which leverages long-short term memory mechanisms, we can effectively capture the temporal characteristics of environmental states. For example, by learning from sequential data such as historical obstacle locations and robot movement trajectories, we can predict the movement trends of dynamic obstacles or environmental change patterns, thereby providing forward-looking environmental state predictions for path planning.

Acknowledgement

This work was supported by the General Scientific Research Project of Zhejiang Provincial Department of Education in 2024 (Project No.: Y202455503).

References

[1] G.N. DeSouza, A.C. Kak, Vision for mobile robot navigation: A survey, IEEE transactions on pattern analysis and machine intelligence 24(2)(2002) 237-267.
https://doi.org/10.1109/34.982903

[2] M.K. Ng, Y.W. Chong, K. Ko, Y. Park, Y. Leau, Adaptive path finding algorithm in dynamic environment for warehouse robot, Neural Computing and Applications 32(17)(2020) 13155-13171.
https://doi.org/10.1007/s00521-020-04764-3

[3] M.N.A. Wahab, S. Nefti-Meziani, A. Atyabi, A comparative review on mobile robot path planning: Classical or meta-heuristic methods?, Annual Reviews in Control 50(2020) 233-252.
https://doi.org/10.1016/j.arcontrol.2020.10.001

[4] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in engineering software 95(2016) 51-67.
https://doi.org/10.1016/j.advengsoft.2016.01.008

[5] R. Kabir, Y. Watanobe, M.R. Islam, K. Naruse, Enhanced Robot Motion Block of A-Star Algorithm for Robotic Path Planning, Sensors 24(5)(2024) 1422.
https://doi.org/10.3390/s24051422

[6] A.K. Guruji, H. Agarwal, D.K. Parsediya, Time-efficient A* algorithm for robot path planning, Procedia Technology 23(2016) 144-149.
https://doi.org/10.1016/j.protcy.2016.03.010

[7] X. Hu, L. Chen, B. Tang, D. Cao, H. He, Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles, Mechanical systems and signal processing 100(2018) 482-500.
https://doi.org/10.1016/j.ymssp.2017.07.019.

[8] V. Lumelsky, A. Stepanov, Dynamic path planning for a mobile automaton with limited information on the environment, IEEE transactions on Automatic control 31(11)(2003) 1058-1063.
https://doi.org/10.1109/TAC.1986.1104175

[9] B.H. Meng, I.S. Godage, I. Kanj, RRT*-based path planning for continuum arms, IEEE Robotics and Automation Letters 7(3)(2022) 6830-6837.
https://doi.org/10.1109/LRA.2022.3174257

[10] B. Liao, F. Wan, Y. Hua, R. Ma, S. Zhu, X. Qing, F-RRT*: An improved path planning algorithm with improved initial solution and convergence rate, Expert Systems with Applications 184(2021) 115457.
https://doi.org/10.1016/j.eswa.2021.115457

[11] P.G. Luan, N.T. Thinh, Hybrid genetic algorithm based smooth global-path planning for a mobile robot, Mechanics Based Design of Structures and Machines 51(3)(2023) 1758-1774.
https://doi.org/10.1080/15397734.2021.1876569

[12] B. Tao, J.H. Kim, Mobile robot path planning based on bi-population particle swarm optimization with random perturbation strategy, Journal of King Saud University-Computer and Information Sciences 36(2)(2024) 101974.
https://doi.org/10.1016/j.jksuci.2024.101974

[13] M.N. Wahab, A. Nazir, A. Khalil, W.J. Ho, M.F Akbar, M.H.M. Noor, A.S.A. Mohamed, Improved genetic algorithm for mobile robot path planning in static environments, Expert Systems with Applications 249(2024) 123762.
https://doi.org/10.1016/j.eswa.2024.123762

[14] X. Pu, X. Song, L. Tan, Y. Zhang, Improved ant colony algorithm in path planning of a single robot and multi-robots with multi-objective, Evolutionary Intelligence 17(3)(2024) 1313-1326.
https://doi.org/10.1007/s12065-023-00821-7

[15] S. Fu, H. Huang, S. Wang, Research on Robot Path Planning Technology by Integrating Particle Swarm Optimization Algorithm and DWA Algorithm, in: Proc. 2024 10th International Conference on Control, Automation and Robotics, 2024.
https://doi.org/10.1109/ICCAR61844.2024.10569525.

[16] B. Song, S. Tang, Y. Li, A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments, Mathematical Biosciences and Engineering 21(2)(2024) 2189-2211.
https://doi.org/10.3934/mbe.2024096

[17] H.T. Najm, N.S. Ahmad, A.S. Al-Araji, Enhanced path planning algorithm via hybrid WOA-PSO for differential wheeled mobile robots, Systems Science & Control Engineering 12(1)(2024) 2334301.
https://doi.org/10.1080/21642583.2024.2334301

[18] G. Hu, F. Huang, B. Shu, G. Wei, MAHACO: Multi-algorithm hybrid ant colony optimizer for 3D path planning of a group of UAVs, Information Sciences 694(2025) 121714.
https://doi.org/10.1016/j.ins.2024.121714
[19] F. Ye, P. Duan, L. Meng, L. Xue, A hybrid artificial bee colony algorithm with genetic augmented exploration mechanism toward safe and smooth path planning for mobile robot, Biomimetic Intelligence and Robotics 5(2)(2025) 100206.
https://doi.org/10.1016/j.birob.2024.100206
[20] C. Fei, Z. Lu, W. Jiang, Heuristic optimization-based trajectory planning for UAV swarms in urban target strike operations, Drones 8(12)(2024) 777.
https://doi.org/10.3390/drones8120777.
[21] W. Jiang, H. Han, Y. Zhang, J. Wu, Multi-controller placement in software defined satellite networks: A meta-heuristic approach, in: Proc. 2024 IEEE 99th Vehicular Technology Conference, 2024.
https://doi.org/10.1109/VTC2024-Spring62846.2024.10683485
[22] N. Morales, J. Toledo, L. Acosta, Path planning using a multiclass support vector machine, Applied Soft Computing 43(2016) 498-509.
https://doi.org/10.1016/j.asoc.2016.02.037
[23] T. Xu, S.W. Chen, D. Wang, T. Wu, Y. Xu, W.G. Zhang, A novel path planning method for articulated road roller using support vector machine and longest accessible path with course correction, IEEE Access 7(2019) 182784-182795.
https://doi.org/10.1109/ACCESS.2019.2959346
[24] D. Sartori, D. Zou, L. Pei, W. Yu, CNN-based path planning on a map, in: Proc. 2021 IEEE International Conference on Robotics and Biomimetics, 2021.
https://doi.org/10.1109/ROBIO54168.2021.9739331
[25] Y.J. Choi, T. Rahim, I.N.A. Ramatryana, S.Y. Shin, Improved CNN-based path planning for stairs climbing in autonomous UAV with LiDAR sensor, in: Proc. 2021 international conference on electronics, Information, and Communication, 2021.
https://doi.org/10.1109/ICEIC51217.2021.9369805
[26] R.S. Nair, P. Supriya, Robotic path planning using recurrent neural networks, in: Proc. 2020 11th International Conference on Computing, Communication and Networking Technologies, 2020.
https://doi.org/10.1109/ICCCNT49239.2020.9225479
[27] E.A. Banu, S. Chidambaranathan, N.N. Jose, P. Kadiri, R.E. Abed, A. Al-Hilali, A System to Track the Behaviour or Pattern of Mobile Robot Through RNN Technique, in: Proc. 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering, 2024.
https://doi.org/10.1109/ICACITE60783.2024.10617430
[28] J. Zhang, Y. Guo, L. Zheng, Q. Yang, G. Shi, Y. Wu, Real-time UAV path planning based on LSTM network, Journal of Systems Engineering and Electronics 35(2)(2024) 374-385.
https://doi.org/10.23919/JSEE.2023.000157
[29] Y. Zhang, P. Chen, Path planning of a mobile robot for a dynamic indoor environment based on an SAC-LSTM algorithm, Sensors 23(24)(2023) 9802.
https://doi.org/10.3390/s23249802
[30] E.S. Low, P. Ong, K.C. Cheah, Solving the optimal path planning of a mobile robot using improved Q-learning, Robotics and Autonomous Systems 115(2019) 143-161.
https://doi.org/10.1016/j.robot.2019.02.013
[31] M. Pei, H. An, B. Liu, C. Wang, An improved dyna-q algorithm for mobile robot path planning in unknown dynamic environment, IEEE Transactions on Systems, Man and Cybernetics: Systems 52(7)(2021) 4415-4425.
https://doi.org/10.1109/TSMC.2021.3096935
[32] J.O. Agushaka, A.E. Ezugwu, Initialisation approaches for population-based metaheuristic algorithms: a comprehensive review, Applied Sciences 12(2)(2022) 896.
https://doi.org/10.3390/app12020896
[33] P.M. Diaz, M.J.E. Jiju, A comparative analysis of meta-heuristic optimization algorithms for feature selection and feature weighting in neural networks, Evolutionary Intelligence 15(4)(2022) 2631-2650.
https://doi.org/10.1007/s12065-021-00634-6