International Journal of Industrial Engineering, 32(4), 876-891, 2025

RESEARCH ON TRAJECTORY PLANNING CONTROL OF INDUSTRIAL MANIPULATOR BASED ON ALO ALGORITHM

Xianxia Liang[1,2*], Longfei Yue[1], Yupeng Li[1], Longxue Cheng[1], and Litao Sun[1]

[1]Department of Electrical Engineering
Hebei Institute of Mechanical and Electrical Technology
Xingtai, China
[*]Corresponding author's e-mail: liangxianxia@hbjd.edu.cn

[2]Technology Innovation Center for Digital Design and Integration of Intelligent Equipment
Hebei Institute of Mechanical and Electrical Technology
Xingtai, China

Aiming at the shortcomings of the ant lion optimization algorithm (ALO) in industrial manipulator trajectory planning, such as long path length, time-consuming rotation time, and uneven path, an improved ALO (IALO) is proposed. Firstly, the population is initialized by cubic chaotic mapping to improve the quality of ant lion population. Secondly, the trust region mutation is used to improve the location update mode of ant lion population and balance the global search ability and local mining ability. Finally, the Gaussian mutation disturbance strategy is used to improve the location update mode of ant lion population and enhance the ability of the algorithm to jump out of local optimization. Taking trajectory length, rotation time, and redundancy rate as indicators, compared with the ABC algorithm and classic ALO, this algorithm has a shorter path length and less rotation time.

Keywords: Cubic Chaotic Mapping; Trust Region Mutation; Gaussian Mutation Disturbance Strategy; Industrial Manipulator; Trajectory Planning.

(Received on February 7, 2025; Accepted on July 28, 2025)

1. INTRODUCTION

With the rapid development of modern manufacturing and automation technology, industrial manipulator have played an important role in improving production efficiency, accuracy, and flexibility (Sandakalum *et al*., 2022). In the industrial robot system, trajectory planning control, as one of the core issues, determines how the manipulator can move smoothly and accurately from the initial position to the target position to complete all kinds of complex tasks (Hentout *et al*., 2023). Trajectory planning not only needs to generate a feasible and optimized path, but also needs to meet the kinematics, dynamics, and environmental constraints of the manipulator (Xuhai *et al*., 2023). Therefore, how to design and control the trajectory of manipulator efficiently becomes a key factor to improve the performance of the industrial robot system. Traditional industrial robot trajectory planning methods usually model the problem as an optimization task, which involves many factors such as path length, motion smoothness, speed profile, joint limitation, and obstacle avoidance (J. Liu *et al*., 2024).

Traditional industrial manipulator trajectory planning methods mainly include Dijkstra (Alshammrei *et al*., 2022), A* (Kabir *et al*., 2024), interpolation method (Luo *et al*., 2015), optimal control method (Massaro *et al*., 2023), and so on. In the trajectory planning of industrial manipulator, the Dijkstra algorithm gradually expands the search range by constantly selecting the shortest path node until the shortest path from the starting point to the end point is found. However, it is only suitable for obstacle avoidance and path planning in a static environment. Its computational complexity is high, especially in high-dimensional space or with many obstacles, and the search process may be very time-consuming (L. Liu *et al*., 2023). For the A* algorithm, the heuristic function is introduced to evaluate the path, so that the algorithm can find the shortest path and has higher efficiency.

Then, the design of the heuristic function of A* algorithm is more critical, and the wrong heuristic function may lead to inefficient search or failure to find the optimal path (Tang *et al*., 2021). The interpolation method constructs a smooth trajectory between a group of discrete path points through interpolation technology. The commonly used interpolation methods include linear interpolation, spline interpolation, and Bessel interpolation, which have good performance in smoothness and continuity of robot path and are suitable for details processing in trajectory planning. However,

DOI: 10.23055/ijietap.2025.32.4.10921 ISSN 1072-4761

interpolation usually assumes that the discrete points of the path are given, and it does not involve obstacle avoidance or path optimization, so its ability is limited for complex obstacle environments or paths that need to be optimized (González *et al*., 2015). For the optimal control method, by introducing mathematical optimization technology and considering the constraints of path, speed, acceleration, and other aspects, the trajectory of the manipulator is optimized to ensure the minimization of energy consumption and time consumption during the movement. However, the calculation process of the optimal control method is usually complicated, and the calculation amount in high-dimensional space is large, so it needs strong calculation ability (Ma *et al*., 2022). Most of these trajectory planning methods rely on analytical solutions or numerical calculation methods, and they perform well under simple constraints. Then, with the increasing complexity of robot tasks and the increasing uncertainty of the environment, it is difficult for traditional methods to deal with high-dimensional, nonlinear, and multi-objective optimization problems, which makes trajectory planning more and more difficult.

In order to overcome these limitations, in recent years, researchers have begun to use natural heuristic optimization algorithms to solve complex, non-convex, and high-dimensional trajectory planning problems. By simulating the optimization process in nature, these heuristic algorithms can efficiently search the complex solution space and avoid falling into the local optimal solution, and have achieved remarkable results. The Ant Lion Optimizer (ALO) (Mirjalili, 2015) is a swarm intelligence optimization algorithm developed by Mirjalili by imitating the foraging behavior strategy of the ant lion. Because of its excellent global search ability and efficient convergence performance, it stands out among many heuristic algorithms and has attracted wide attention. Although ant lion optimization algorithm has achieved good application results in many optimization fields, when solving complex engineering optimization problems, the classic ALO algorithm still has some problems, such as slow convergence speed and easy to fall into local optimization, and its application in industrial manipulator trajectory planning is still limited, especially in real-time path generation and dynamic environment adaptation (Heidari *et al*., 2020). Therefore, in order to solve the above problems, this paper proposes an improved ALO algorithm, IALO, to solve the trajectory planning problem of industrial manipulator. Firstly, the ant lion population is initialized by the cubic chaotic mapping method, which makes its population distribution more uniform and increases the diversity of the population. Then, a trust region mutation is proposed to improve the position of the ant lion population in the ant lion trap stage, so that the algorithm keeps a balance between global search and local mining. Finally, the Gaussian mutation disturbance strategy is used to update the position of ant lion population in the trap reconstruction stage, which enhances the ability of the algorithm to jump out of local optimization.

The main contributions of this paper are as follows:

1. An improved ALO algorithm (IALO) is innovatively proposed to solve the path planning problem of the industrial manipulator for operation and maintenance.
2. The cubic chaotic mapping method is used to initialize the ant lions foraging population and improve the quality of the initial solution of the algorithm.
3. A trust region mutation is proposed to improve the location of ant lions in the foraging stage, and balance the global search ability and local excavation ability of the algorithm.
4. The Gaussian mutation disturbance strategy is used to update the position of ant lions in the chain foraging stage, and enhance the ability of the algorithm to jump out of local optimization.

2. RELATED WORK

The traditional global path planning algorithm is widely used to solve the trajectory planning problem of industrial manipulator. In order to make the robot manipulator work efficiently and safely in a cooperative dynamic unstructured environment, Wei *et al*. (2018) proposed an autonomous obstacle avoidance dynamic path planning method for robot manipulator (S-RRT) based on the improved RRT algorithm. The sampling speed and efficiency of RRT can be significantly improved by extending the method with directed nodes as the target, which has the advantages of probability completeness, perfect expansibility, and fast exploration speed, and has been widely used in the dynamic path planning of high-dimensional robot manipulators. In addition, a path optimization strategy based on maximum curvature constraint is proposed to generate smooth and curved continuous executable paths for the manipulator, and the correctness, effectiveness, and practicability of this method are verified by autonomous obstacle avoidance experiments. Aiming at the problems of slow convergence and low search efficiency of RRT* algorithm, Yi *et al*. (2022) proposed a path planning method of manipulator based on improved RRT* algorithm, which was used to solve the path planning problem of manipulator in three-dimensional space. Firstly, random sampling method based on potential function was adopted, and the nearest neighbor node was selected by using the nearest Euclidean distance of random sampling points and the minimum cost function based on probability value, and in the expansion of new nodes, the search efficiency of the algorithm was accelerated by twice expansion. Then, re-select the parent node of the new node and re-route the path to get a clear path from the initial point to the target point. Redundant node deletion and maximum curvature constraint are used to delete redundant nodes and minimize the curvature on the generated path to reduce the tortuosity of the path. Moreover, a Bezier curve is used to fit the machining path and obtain the trajectory planning curve of the manipulator. The experimental results verify the effectiveness and superiority of the improved algorithm. So as to solve the problem of low efficiency in planning the trajectory of intelligent manipulator in free space with narrow channels, Chen *et al*. (2021) proposed a new PRM sampling strategy based on virtual force field to generate a configuration more suitable for practical application in free space, and designed a three-stage connection strategy, which can gradually improve the connectivity of road map, increase the sampling density of narrow channels in free space and reduce the sample redundancy in open areas of free space, and is more suitable for path planning of manipulator in complex environment, and can quickly plan a path in the presence of obstacles. For the purpose of planning a successful and practical path for the manipulator in the complex obstacle environment, Jiang *et al*. (2022) proposed an improved RRT method, established a collision detection model between the manipulator and the obstacle based on the cylindrical and spherical bounding boxes, calculated the shortest distance between the connecting rod and the obstacle at each time, and replaced the original random sampling with mixed constraint sampling, and the node distribution obtained was close to the channel between the obstacles. In addition, combined with the artificial potential field method, the information of the environment and nodes is used to expand to the target area in a short time, which reduces the over-exploration and collision area expansion. Using local minimum processing and dynamic path planning methods, dual trees are generated in Cartesian space and configuration space, respectively, for local path re-planning. The simulation results show that the improved algorithm is more robust to the singularity problem of the manipulator, and the optimized path is smoother after pruning, which can effectively plan a safe and optimal obstacle avoidance path with the best comprehensive performance. In an effort to realize the obstacle avoidance of litchi picking robot in dynamic and unstructured environment, Cao *et al*. (2019) proposed an improved RRT algorithm. By establishing the collision detection model between the manipulator and obstacles, the path planning was carried out. The idea of target gravity was adopted to speed up the path search, and the genetic algorithm was combined to smooth the path, which effectively reduced the path length and successfully drove the manipulator from the initial position to the target position without collision. Aiming at the trajectory planning problem of industrial robots, Shen *et al*. (2023) proposed a new optimal RRT* path planning strategy based on operability. When sampling in the search space, two constraints of path length and operability were imposed to find the minimum cost path connecting the starting point and the target point. By tracking the generated path, the end effector can traverse the workspace in a shorter length, while avoiding the singularity of configuration, improving the operability, and helping the space-filled tree to grow efficiently to the unexplored area. With the aim of solving the problem that the candidate paths generated by traditional local path planning methods are not suitable for dynamic environments and narrow space, Masato *et al*. (Kobayashi *et al*., 2022) proposed a new local path planning method for virtual manipulator based on dynamic window method. By assuming a constant speed to generate straight and circular candidate paths, and using virtual manipulators and environmental information to generate the speed of reflection motion, the candidate paths can be generated according to the variable speed and the predicted positions of static and dynamic obstacles, and obstacle avoidance paths including non-straight and non-arc paths can be generated in the environment with dynamic obstacles. The experimental results verify the effectiveness of this method.

In recent years, the swarm intelligence optimization algorithm has been gradually applied to the trajectory planning of industrial robots. With a view to avoid the collision between paths or trajectories and enable industrial manipulators to plan paths according to assigned tasks, Ranjan *et al*. (2023) proposed an improved PSO algorithm to solve the path planning problem of industrial manipulators, which helped to find the best possible shortest path between two points, controlled the geometry and topological shape of paths, and showed the cooperative behavior of self-organizing and decentralized systems. Moreover, due to the local interaction between population particles, Disordered particles exchange their best experience and produce global coordination among particles in an orderly way, which is helpful to find a possible path when the robot system encounters some spherical obstacles in the configuration space and avoid the obstacles in the space. In order to improve the operation accuracy and productivity of the flexible manipulator and suppress the vibration of the flexible manipulator, Cui *et al*. (2020) proposed a trajectory planning method to suppress the vibration of the flexible manipulator in space, established the dynamic model of each link of the flexible manipulator, and established the dynamic model of the flexible manipulator as differential-algebraic equations according to the constraint equation. Then, the trajectory function is designed as a quintic polynomial, and the conditions that satisfy the acceleration limit of each joint are deduced. Then the trajectory planning problem is transformed into an optimization problem, and the PSO algorithm is used to solve the optimization problem. The numerical simulation shows that the method has good performance. Wang *et al*. (2019) proposed a new trajectory planning TPBSO algorithm to solve the trajectory planning problem of manipulator, and divided the trajectory planning problem of manipulator into point-to-point planning and fixed geometric path planning. Then, the control model is established, and the numerical experiments on two planning tasks verify the effectiveness of the algorithm, which has relatively fast calculation speed and good control performance without increasing computational complexity. Kim *et al*. (Kim *et al*., 2015) proposed a method based on improved PSO to solve the motion planning problem of industrial manipulators with complex constraints. Taking advantage of PSO's simplicity and fast convergence, the trajectory modification was coded into the particles optimized by particle swarm optimization, and the particles were initialized by using the concept of normalized step cost (NSC), and a parameter reuse method was developed to store the optimized parameters together with the corresponding NSC vectors. When constraint violations occurred, Selecting the parameters associated with the NSC vector improves the convergence of particle swarm optimization in motion planning. The experimental results show that the proposed algorithm successfully optimizes the trajectory while satisfying the constraints, and it is unlikely to converge to the local minimum. In the interest of solving the trajectory planning problem of industrial manipulator, Josenalde *et al.* (2017) designed an optimal trajectory generation method based on the GWO algorithm, which can perform simple, error-free, fast convergent continuous motion, find the trajectory path with minimum tracking error, combination speed, joint acceleration wrinkle, and joint tilt motion, and can be a smooth continuous path without error. The effectiveness of the algorithm is verified by simulation experiments. Garg *et al*. (2002) combined the GA algorithm and the SA algorithm, proposed a new strategy for optimal trajectory planning of multiple industrial manipulators. By using genetic algorithms and simulated annealing as optimization tools, the initial and final positions of the end effector were designated as single manipulator and double manipulator carrying the same payload, and the optimal trajectory was identified based on the minimum joint torque requirement. The simulation results show that the proposed method can converge to the global minimum, and the convergence speed is faster.

3. CLASSICAL ALO METHOD

The ant lion optimization algorithm is inspired by the process of catching ants by the ant lion. Ants crawl irregularly on the ground, and the ant lion makes traps in advance to wait for the ants. When the ants are caught by the ant lion, the ant lion will continue to dig traps to wait for the next ant. The ALO algorithm is to solve practical problems by imitating the process of catching ants by the ant lion, to learn from excellent ant lions to ensure the global optimal value, and to find the optimal solution by constantly preying on ants.

3.1 Random Walks

In order to ensure that ants walk randomly in the feasible region to complete the search, this paper normalizes the random position of ants to make their movement range within the feasible region, and expresses it as follows, with the formula:

$$X_i^t = \frac{(X_i^t - a_i)\times(d_i^t - c_i^t)}{(b_i - a_i)} + c_i \quad , \tag{1}$$

where X_i^t represents the position where ants walk randomly, and t represents the number of iterations. i represents the number of variables and c_i stands for the number of random walks. a_i represents the minimum value of ant random walk. b_i stands for the maximum value of ant random walk. c_i^t represents the number of random walks of the *i*-th variable in the *t*-th iteration, and d_i^t represents the maximum value of the *i*-th variable in the *t*-th iteration.

3.2 Antlion Trapping

Because the movement process of ants caught in the edge of the trap is influenced by the ant lion trap, in order to ensure that the ants caught in the trap can be captured by the ant lion, the behavior of this kind of ants is modeled and expressed as follows:

$$\begin{cases} c_i^t = Antlion_j^t + c^t \\ d_i^t = Antlion_j^t + d^t \end{cases} , \tag{2}$$

where c_i^t is the minimum value of all variables in the t-th iteration. d_i^t represents a vector including the maximum value of all variables in the *t*-th iteration. c^t is the minimum value of all variables of the *i*-th ant. d^t is the maximum value of all variables of the *i*-th ant, and $Antlion_j^t$ represents the position of the selected j-th ant in the *t*-th iteration.

Assuming that the ant lion fitness is represented by $f(Antlion_j^t)$, and the ant lion is selected by roulette strategy according to its fitness, and the ant lion with lower fitness has a higher chance of catching ants. In order to improve the convergence speed of the algorithm, the range of ant random walk is reduced with the increase of iteration times, and its mathematical model is expressed as follows:

$$\begin{cases} c' = \dfrac{c^t}{I} \\ d' = \dfrac{d^t}{I} \end{cases}, \tag{3}$$

where c' is the minimum value of all variables in the t iteration, and d' represents a vector including the maximum value of all variables in the t iteration. I is the ratio, which reduces the radius of ant location update, simulates the sliding process of ants in the pit, and ensures the utilization of search space, which is expressed by the formula as follows:

$$I = \begin{cases} 1 & , t \leqslant 0.1T \\ 10^w \cdot \dfrac{t}{T} & , t > 0.1T \end{cases} \tag{4}$$

where w represents a random number that obeys normal distribution, T represents the maximum number of iterations, and t represents the iteration.

3.3 Pit Rebuilding

When the ant lion catches the ant, it needs to build a trap to catch the ant. If the fitness of an ant is higher than that of an ant lion, it is considered to be captured by the ant lion, and the position of the ant lion is updated, which is expressed by the formula as follows:

$$Antlion_j^t = Ant_j^t \text{ if } f(Ant_j^t) > f(Antlion_j^t) \tag{5}$$

where $Antlion_j^t$ stands for ant lion, Ant_j^t stands for ant, and $f(x)$ stands for fitness function.

3.4 Elite Strategy

Elite strategy, as an important feature of the evolutionary algorithm, can ensure that the optimal solution obtained at each stage can be preserved in the optimization process of the algorithm and continue to affect the evolution of the solution in subsequent iterations. For the ALO algorithm, the best ants obtained in each iteration are saved and regarded as elites, which can affect the movement of all ants in the iteration process. By using the double optimization measures of roulette wheel selection's ant lion and elite ant lion, the current random movement of ants is influenced. The ant lion with the best fitness is determined as the current optimal elite individual, and its information is stored, which is expressed by the formula as follows:

$$Ant_i^{t+1} = \frac{R_A^t(l) + R_E^t(l)}{2} \tag{6}$$

where, Ant_i^{t+1} represents the value generated when the ant lion randomly walks in roulette strategy selection, and $R_A^t(l)$ represents the number of steps of random walking. $R_E^t(l)$ represents the position of ant lion randomly walking around the elite.

4. IMPROVED ALO METHOD

4.1 Cubic Chaotic Mapping

According to the ant lion algorithm, in the process of updating the initial individual position of the ant lion, the solution set is constantly approaching the current optimal solution. If the current optimal solution is the global optimal solution, the algorithm can quickly complete the optimization search. If the current optimal solution is not the global optimal solution, the optimization process of the algorithm may fall into the local optimal solution (Ali *et al*., 2016). In addition, the optimization process of the ant-lion algorithm depends on the interaction between individuals, and the random search formula is used to generate the same number of new solutions as the discarded solutions. After evaluation, the best ant-lion in the population will be recorded and spread to the next generation to complete an iteration. With the increase of iterations, the individuals in the population will approach the optimal solution step by step. At this time, the diversity of the population drops sharply. After the diversity of the population drops, the probability of the algorithm falling into a local optimum increases, which will greatly affect the accuracy and convergence speed of the algorithm.

Chaotic mapping has the characteristics of randomness, regularity, and ergodicity, which makes it a tool widely used in optimization algorithms (Yang *et al*., 2015). In the process of solving the algorithm, the initialization mode of the population has an important influence on the results. By using chaotic mapping to initialize, the diversity of the initial population can be increased, thus further improving the global search ability of the algorithm. Based on the above reasons, the Cubic chaotic mapping perturbation strategy is introduced into the ant lion algorithm, which increases the diversity of the population in the optimization process of the algorithm and avoids the algorithm from falling into a local optimum, thus improving the solution accuracy and convergence speed of the algorithm. Compared with the Tent chaotic map and Sine chaotic map, the search ability of Cubic chaotic map perturbation strategy is better in convergence speed and accuracy (Lian *et al*., 2020). Therefore, this paper uses Cubic chaotic mapping to search in the allowed solution space to initialize the ant lion population and effectively improve the quality of the initial solution, which is expressed as follows:

$$r_{k+1} = ur_k^3 + (1-u)r_k \ , \tag{7}$$

where, r_k is the *k*-th chaotic number, and k is the number of iterations. When $3.2 < u < 4$, the output is a chaotic sequence $-1 \leq r_{k+1} < 1$.

4.2 Trust Region Mutation

In the antlion trapping stage of the original ant lion algorithm, although it includes different attack behaviors of global exploration and search, it ignores the situation that the prey will escape in the attack of the ant lion, which makes the algorithm fail to find the global optimal solution and easily leads to the local optimal solution (Cheng *et al*., 2015). In addition, the search intensity of the algorithm is limited due to the lack of information exchange in the attack stage. Therefore, by introducing the quantitative mechanism of prey escape energy, the prey escape energy in the ant lion algorithm is defined as follows:

$$E = (2 \times rand(1) - 1) \times exp\left(-2 \times \left(\frac{t}{T}\right)^2\right) \tag{8}$$

where E_0 stands for prey escape threshold, which is set to 0.1. $rand(1)$ represents a random number from 0 to 1. When $|E| < E_0$, it means that the prey has no extra energy to escape from the attack circle of the ant lion. At this time, the original attack strategy is adopted to update the position, and the formula is as follows:

$$X_{t+1}^{i,j} = X_t^{i,j} + E\big(1 + sin(r_d)\big) \times X_t^{i,j} \ , \tag{9}$$

where, $X_{t+1}^{i,j}$ represents the position of the updated ant lion. r_d represents a random number from 0 to 1. According to the formula, the updated ant lion position changes slowly with the increase of iteration times, which leads to the limited search range of the algorithm, while E oscillates between (-1,1), which makes the update step of the algorithm more random so that the algorithm can search more promising areas and find more excellent solutions.

When $|E| \geq E_0$, it means that the prey has enough escape energy to escape the attack of the ant lion. At this time, the ant lion should intensify its search to prevent the prey from escaping. Therefore, this paper combines the idea of trust region to make the current individual communicate with all individuals in its trust region, thus improving the convergence speed of the algorithm and preventing the algorithm from falling into local optimum.

First of all, the Euclidean distance between individual i and individual j of ant lion is defined and expressed as follows:

$$d\big(X_i, X_j\big) = \sqrt{\big(x_{i1} - x_{j1}\big)^2 + \big(x_{i2} - x_{j2}\big)^2 + \cdots + \big(x_{iD} - x_{jD}\big)^2} \tag{10}$$

where, $d\big(X_i, X_j\big)$ represents the Euclidean distance between the i-th ant lion and the j-th ant lion. Then, calculate the average position of all individuals of the ant lion, which is expressed by the formula as follows:

$$X_{me} = \frac{1}{N} \sum_{i=1}^{N} d\left(X_i, X_j\right) \ , \tag{11}$$

where, X_{me} is the average distance of all ant lion individuals, the initial trust region radius of the current individual is calculated according to the distance between the current individual and the average individual, which is expressed by the formula as follows:

$$R_i = \frac{f(X_i)}{\sum_{i=1}^{N} f(X_i)} \times d(X_i, X_{me}) \ , \tag{12}$$

where, $f(X_i)$ is the fitness function, $d(X_i, X_{me})$ represents the individual dimension, X_i is the position of the ith individual in the d dimension, and R_i is the initial trust region radius of the i-th individual.
Finally, an individual whose distance from the current individual is less than the radius of the trust region is regarded as a trustworthy individual, and the formula is as follows:

$$B(X_{bi}) = \{X_{bi} \in X_i | d(X_i, X_{bi}) \leq R_i\} \ , \tag{13}$$

where, $B(X_{bi})$ stands for the individual ant lion in the trusted domain. X_{bi} stands for the adjacent ant lion individual which enters the trust domain. If there is no trustworthy individual, consider the neighboring individuals as trustworthy individuals, and then calculate the mean vector position X_m^b of trustworthy individuals, which is expressed as follows:

$$X_m^b = \frac{1}{N_0} \sum_{i=1}^{N_0} X_{bi} \ , \tag{14}$$

where, X_m^b stands for the average vector position of individual ants and lions in the trusted domain. N_0 represents the number of individual ants and lions.

4.3 Gaussian Mutation Disturbance

Gaussian mutation disturbance is a strong disturbance that conforms to a Gaussian distribution (Yu *et al*., 2020). By applying the Gaussian mutation disturbance mechanism to the pit rebuilding stage of the ALO algorithm, the algorithm is increased to jump out of local optimization. Then, this paper uses a Gaussian mutation mechanism to disturb the position of the ant lion to obtain the optimal position. The position of the ant lion with Gaussian mutation is expressed as follows:

$$Y_i = X_i + \lambda_k Gaussian(\mu, \sigma^2) \ , \tag{15}$$

where, X_i represents the position of the ant lion. Y_i represents the updated value of the ant lion position after Gaussian mutation. λ_k represents the weight vector that conforms to the disturbance characteristics of the whole ant lion population, and $Gaussian(\mu, \sigma^2)$ is a Gaussian distribution random number with average value μ and standard deviation σ.

According to the characteristics of the Gaussian distribution, Gaussian mutation can realize the key search of the local area near the original individual. Therefore, the mechanism of introducing Gaussian mutation has two important functions that can not be ignored. It can not only effectively expand and strengthen the local search scope and intensity of the ant-lion optimization algorithm, thus helping to improve its local development ability, but also help the ant-lion optimization algorithm to escape local convergence when there is a risk of local convergence in a certain local area. To sum up, the key search area of Gaussian mutation is the local area near the original individual, which not only improves the robustness of the algorithm, but also helps to enhance the local search ability of the algorithm, so that the algorithm can efficiently find the global minimum point, thus promoting the balance between global exploration and local development and improving its global optimization performance.

5. SIMULATION EXPERIMENTS

5.1 Scene Construction

In order to verify the validity and feasibility of the proposed algorithm, the industrial robot arm IRB 120 produced by ABB Company is selected as the simulation model, as shown in Figure 1.

Figure 1. IRB 120 robot

IRB 120 is a high-precision, high-load six-degree-of-freedom robot arm, which is widely used in various automation fields, such as welding, handling, assembly, etc. It has high flexibility and reliability, and has been widely used in industrial production, so it has high representativeness and practical value as a simulation model. In the simulation process, by modeling and solving the kinematics and dynamics model of the IRB 120 robot arm, the robot's path planning, trajectory control, and cooperative operation can be realized. In addition, when the ALO algorithm is used to solve the trajectory planning problem of an industrial manipulator, the ant lion population is set to 40, and the maximum iteration number is 300.

5.2 Comparative Scheme

In workspace 1, the ABC algorithm (Karaboga *et al*., 2011), ALO algorithm, and IALO algorithm are respectively used for trajectory planning of industrial manipulator. Among them, the magenta line is the convergence curve of the ABC algorithm trajectory planning, the blue line represents the convergence curve of the ALO algorithm trajectory planning, and the red line represents the convergence curve of the IALO algorithm trajectory planning. The convergence curve effects of the trajectory planning of the three algorithms are shown in Figure 2.

When trajectory planning is carried out in workspace 1 of IRB 120 manipulator, based on the kinematics model of IRB 120 manipulator, the moving target and constraint conditions of the end effector are defined, and the path optimization is carried out by using the IALO algorithm. Through multi-generation iterative optimization, complex trajectory planning problems can be effectively solved. In order to analyze the effect of the IALO algorithm in trajectory planning, the curve of the position, velocity, and acceleration of the end effector of the manipulator in the X-axis direction is drawn as shown in Figure 3.

As can be seen from the figure, under the control of the IALO algorithm, the X-axis position of the end effector of the industrial manipulator approaches the target position steadily with the progress of time, and the speed and acceleration during the movement show a smooth change trend. Especially in the control of speed and acceleration, the IALO algorithm can effectively avoid violent fluctuations and ensure that the robot is efficient and safe when completing the task.

In order to more intuitively see the trajectory planning effect of the IALO algorithm in workspace 1 of IRB 120 manipulator, the trajectory curve of the IALO algorithm in this space is drawn as shown in Figure 4.

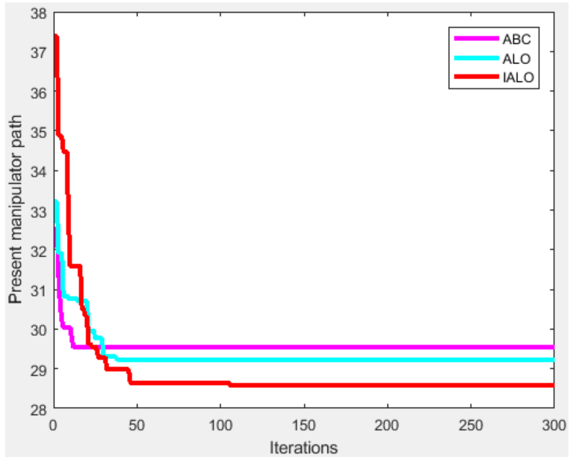


Figure 2. Trajectory convergence results of the proposed algorithm in workspace 1.

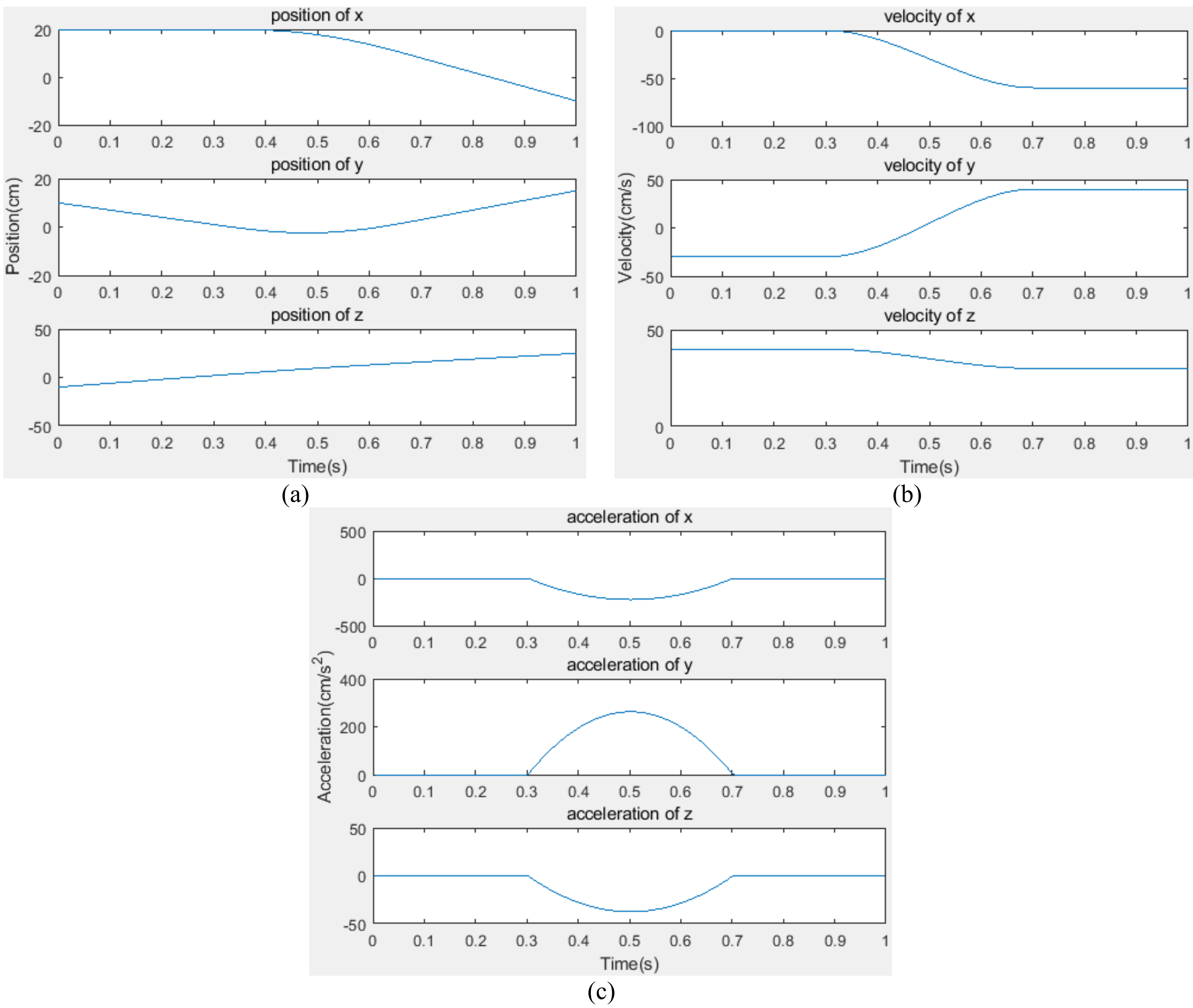


Figure 3. Motion characteristics of IALO in workspace 1: (a) the x-axis position of the end effector; (b) the x-axis velocity of the end effector; (c) the x-axis acceleration of the end effector.

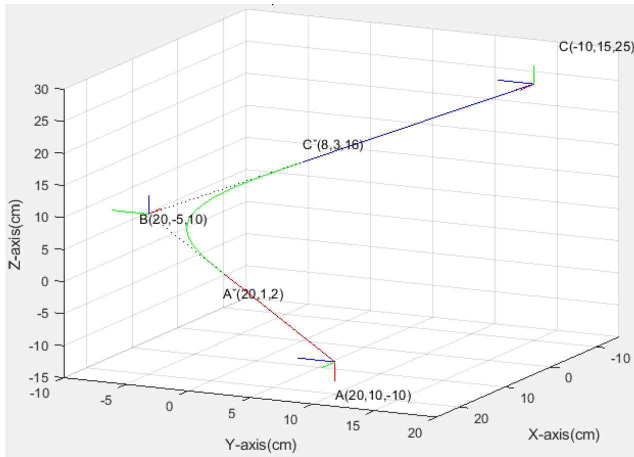


Figure 4. Trajectory of IRB 120 Manipulator in Workspace 1 under the control of IALO.

The figure shows the whole movement process of the end effector of the manipulator from the initial position to the target position. As can be seen from the figure, under the optimization of IALO algorithm, the trajectory changes show a high-precision and smooth transition, especially between the starting point and the target point, and the movement of the end effector of the manipulator shows a smooth curve without obvious abrupt changes, which shows that IALO algorithm has advantages in avoiding shocks and violent fluctuations.

In order to observe the movement change of industrial manipulator IRB 120 under the control of the IALO algorithm more clearly, the movement direction curve of industrial manipulator IRB 120 in workspace 1 is drawn, as shown in Figure 5.

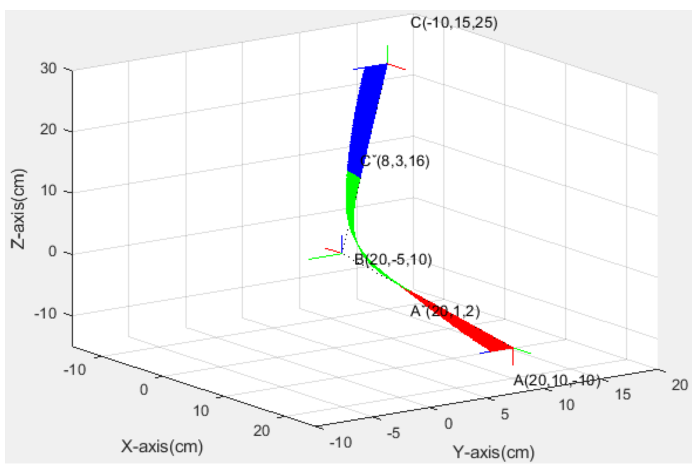


Figure 5. Motion direction trajectory of IRB 120 manipulator in workspace 1 under the control of IALO.

The figure shows the trajectory of the end effector and the changing law of its direction during the task execution of the manipulator. Through this graph, we can intuitively see the movement path and direction adjustment of the manipulator from the initial point to the target point in space, and the smoothness of the curve shows that the IALO algorithm finely adjusts the movement direction in the process of trajectory planning, avoiding unnecessary deviation caused by the instability of the algorithm.

In order to verify the universality of the IALO algorithm, the ABC algorithm, ALO algorithm, and IALO algorithm are applied to the trajectory planning of industrial manipulator in workspace 2, and the convergence effects of the three algorithms are shown in Figure 6.

When trajectory planning is carried out in workspace 2 of IRB 120 manipulator, based on its kinematics model, the moving target and constraint conditions of the end effector are set, and the path optimization is carried out by using the IALO algorithm. Through multi-generation iterative optimization, this method can effectively deal with complex trajectory planning problems. In order to evaluate the performance of the IALO algorithm in trajectory planning, the displacement, velocity, and acceleration curves of manipulator end effector in the X-axis direction are drawn, as shown in Figure 7.

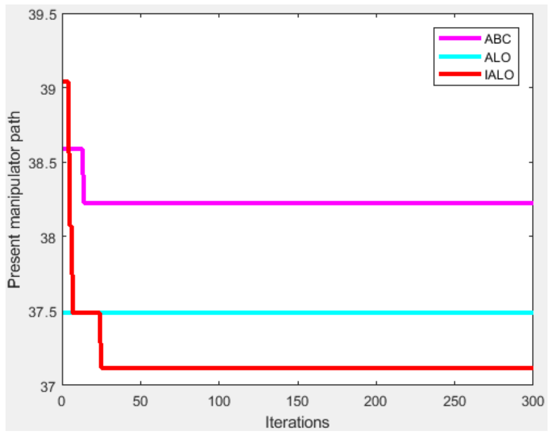
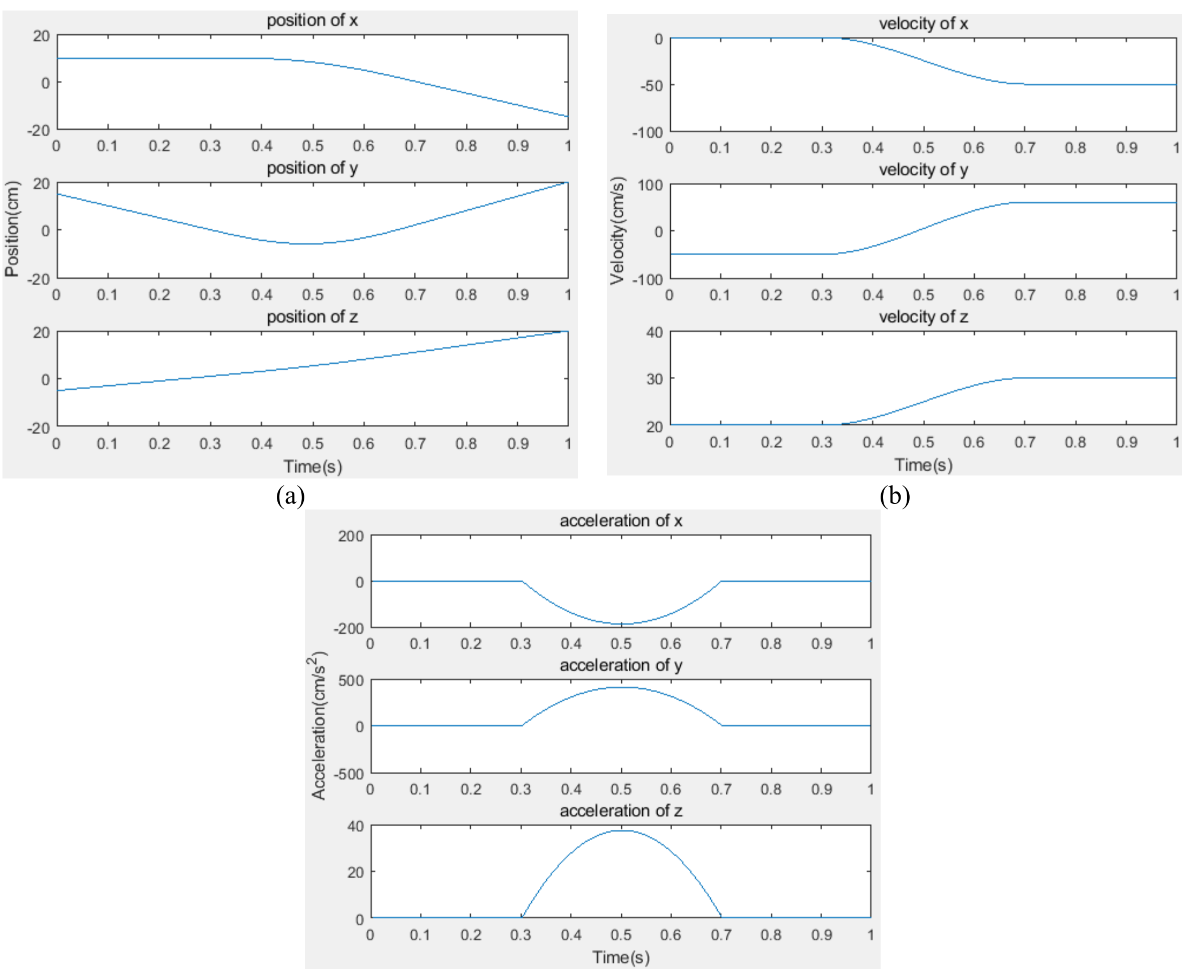


Figure 6. Trajectory convergence results of the proposed algorithm in workspace 2: (a) trajectory planned by the ABC; (b) trajectory planned by the ALO; (c) trajectory planned by the IALO.



(a) (b)

(c)

Figure 7. Motion characteristics of IALO in workspace 2: (a) the x-axis position of the end effector; (b) the x-axis velocity of the end effector; (c) the x-axis acceleration of the end effector.

It can be observed from the figure that the X-axis position of the end effector of the industrial manipulator controlled by the IALO algorithm approaches the target position stably with time, and the speed and acceleration change smoothly during the movement. Especially in the control of speed and acceleration, the IALO algorithm effectively avoids violent fluctuations and ensures the efficiency and safety of the robot when performing tasks. In order to show the trajectory planning effect of the IALO algorithm in workspace 2 of IRB 120 manipulator more intuitively, the trajectory curve in this space is drawn, as shown in Figure 8.

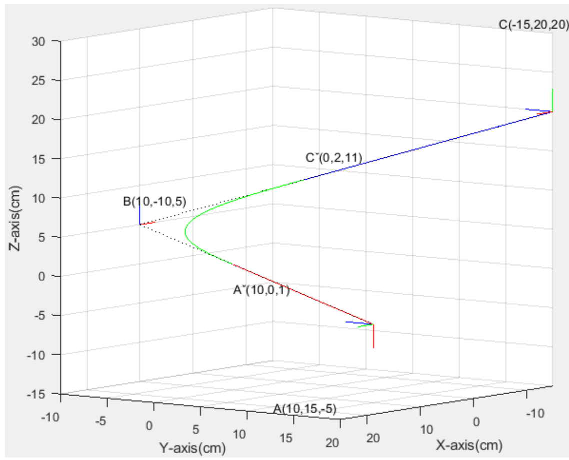


Figure 8. Trajectory of IRB 120 Manipulator in Workspace 2 under the control of IALO.

The figure shows the complete movement process of the manipulator end effector from the initial position to the target position. It can be seen that the trajectory changes show a high-precision and smooth transition under the optimization of the IALO algorithm, especially between the starting point and the target point, and the trajectory of the end effector is smooth and continuous, with no obvious mutation, which shows that the IALO algorithm has obvious advantages in restraining impact and fluctuation. In order to more intuitively show the motion changes of the industrial manipulator IRB 120 under the control of the IALO algorithm, the motion direction curve of the industrial manipulator IRB 120 in workspace 2 is drawn, as shown in Figure 9.

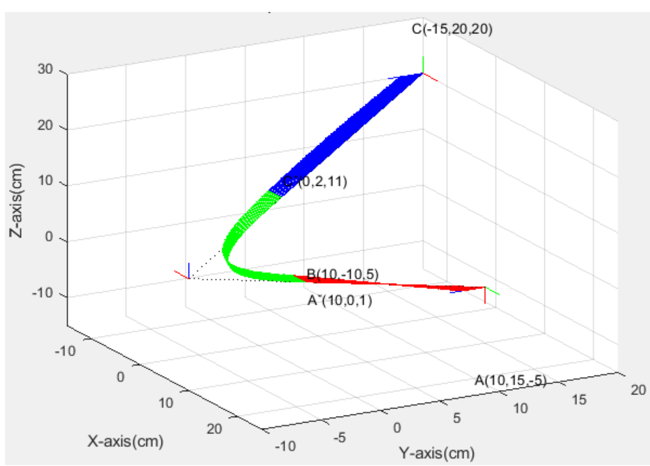


Figure 9. Motion direction trajectory of IRB 120 manipulator in workspace 2 under the control of IALO.

The figure shows the trajectory of the end effector and its direction change law during the task execution of the manipulator. Through this diagram, we can intuitively observe the movement path and direction adjustment of the manipulator from the initial position to the target position. The smoothness of the trajectory shows that the IALO algorithm accurately optimizes the motion direction in the trajectory planning process and effectively avoids the deviation caused by the instability of the algorithm.

After completing the comparative experiment of work space 1 and work space 2, the flow chart of the proposed IALO algorithm is shown in Figure 10, and the pseudo code of the proposed IALO algorithm is listed in Table 1.

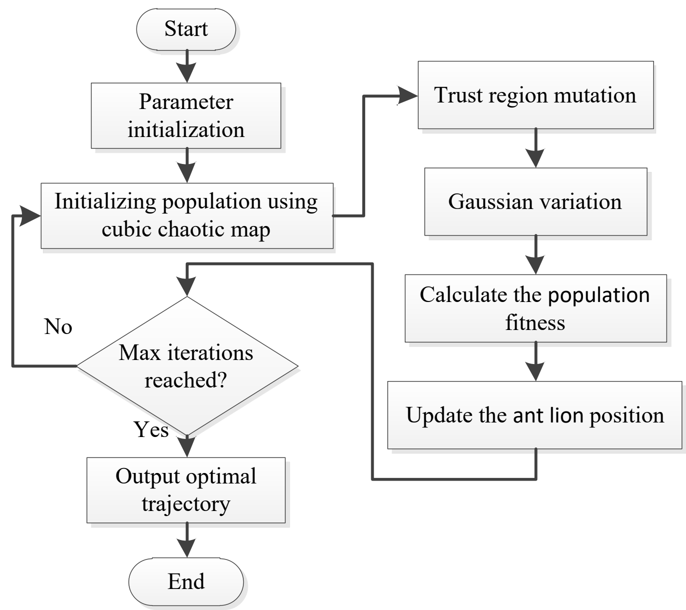


Figure 10. Flow chart of the proposed IALO algorithm.

Table 1. The pseudo-code of the proposed IALO algorithm

Algorithm: IALO.
Input: population size *N*, max iterations *T*.
Output: optimal trajectory.
1. Parameter initialization.
2. Using Cubic chaotic map to generate initial population.
3. Calculate the ant lion fitness.
4. For $i = 1$ to T do:
5. Update the ant lion position.
6. For $i = 1$ to N do:
7. Randomly choose an ant lion according to roulette.
8. if rand() < 0.5:
Perform trust region mutation on the ant lion.
9. else:
Perform Gaussian mutation on the ant lion.
10. End if
11. Update the ant lion position
12. End For
13. End For
14. Update the elite ant lion fitness.
15. Output the best elite ant lion trajectory.
16. End

Then, three algorithms are summarized from three indicators, such as trajectory length, rotation time, and redundancy rate, as shown in Table 2.

Table 2 Comparison of performance indexes of related algorithms

Metrics / Algorithms	Work Space 1			Work Space 2		
	Trajectory Length(m)	Rotation Time(s)	Redundancy Rate(%)	Trajectory Length(m)	Rotation Time(s)	Redundancy Rate(%)
ABC	29.94	14.83	65.41	38.74	11.29	53.58
ALO	29.75	9.17	43.87	37.59	7.28	41.82
IALO	28.36	6.53	38.36	37.02	4.61	32.43

As can be seen from the table, whether in workspace 1 or workspace 2, compared with the ABC algorithm and ALO algorithm, the IALO algorithm has the best planned path length, rotation time, and redundancy rate.

6. CONCLUSION

In this paper, an improved ALO algorithm is innovatively proposed for the trajectory planning of IRB 120 industrial manipulator. Firstly, the ant lion population is initialized by cubic chaotic mapping to improve the diversity of ant lion population. Then, a trust region mutation is proposed to improve the position update mode of ant lion population in the trap stage and balance the global search ability and local exploration ability of the algorithm. Finally, the Gaussian mutation disturbance strategy is used to improve the position update mode of ant lion population in the trap reconstruction stage and increase the probability of the algorithm jumping out of the local optimum. The proposed IALO algorithm is simulated in Workspace 1 and Workspace 2. The results show that compared with the ABC algorithm and ALO algorithm, the trajectory length, rotation time, and redundancy rate of IRB 120 industrial manipulator planned by the IALO algorithm are reduced by 6.58%, 14.68% and 46.58% respectively. The IALO algorithm proposed in this paper is only evaluated in the simulation of IRB 120 industrial manipulator, and future research should focus on applying the IALO algorithm to more types of industrial machinery to verify its effectiveness.

ACKNOWLEDGMENTS

This paper was supported by Key Research and Development Plan Self Funded Project of Xingtai, Hebei Province in 2024 "Research on Dynamic Target Tracking of Industrial Robots Based on Machine Vision" (2024ZC045).

REFERENCES

Ali, E., Abd Elazim, S., & Abdelaziz, A. (2016). Ant lion optimization algorithm for renewable distributed generations. *Energy, 116*, 445-458.

Alshammrei, S., Boubaker, S., & Kolsi, L. (2022). Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance. *Comput. Mater. Contin, 72*, 5939-5954.

Cao, X., Zou, X., Jia, C., Chen, M., & Zeng, Z. (2019). RRT-based path planning for an intelligent litchi-picking manipulator. *Computers and electronics in agriculture, 156*, 105-118.

Chen, G., Luo, N., Liu, D., Zhao, Z., & Liang, C. (2021). Path planning for manipulators based on an improved probabilistic roadmap method. *Robotics and Computer-Integrated Manufacturing, 72*, 102196.

Cheng, G. H., Younis, A., Haji Hajikolaei, K., & Gary Wang, G. (2015). Trust region based mode pursuing sampling method for global optimization of high dimensional design problems. *Journal of Mechanical Design, 137*(2), 021407.

Cui, L., Wang, H., & Chen, W. (2020). Trajectory planning of a spatial flexible manipulator for vibration suppression. *Robotics and Autonomous Systems, 123*, 103316.

Garg, D. P., & Kumar, M. (2002). Optimization techniques applied to multiple manipulators for path planning and torque minimization. *Engineering applications of artificial intelligence, 15*(3-4), 241-252.

González, D., Pérez, J., Milanés, V., & Nashashibi, F. (2015). A review of motion planning techniques for automated vehicles. *IEEE Transactions on intelligent transportation systems, 17*(4), 1135-1145.

Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., & Mafarja, M. (2020). Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. *Nature-inspired optimizers: theories, literature reviews and applications*, 23-46.

Hentout, A., Maoudj, A., & Aouache, M. (2023). A review of the literature on fuzzy-logic approaches for collision-free path planning of manipulator robots. *Artificial Intelligence Review, 56*(4), 3369-3444.

Jiang, L., Liu, S., Cui, Y., & Jiang, H. (2022). Path planning for robotic manipulator in complex multi-obstacle environment based on improved_RRT. *IEEE/ASME transactions on mechatronics, 27*(6), 4774-4785.

Kabir, R., Watanobe, Y., Islam, M. R., & Naruse, K. (2024). Enhanced Robot Motion Block of A-Star Algorithm for Robotic Path Planning. *Sensors, 24*(5), 1422.

Karaboga, D., & Ozturk, C. (2011). A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied soft computing, 11*(1), 652-657.

Kim, J.-J., & Lee, J.-J. (2015). Trajectory optimization with particle swarm optimization for manipulator motion planning. *IEEE transactions on industrial informatics, 11*(3), 620-631.

Kobayashi, M., & Motoi, N. (2022). Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles. *IEEE Access, 10*, 17018-17029.

Kumar, R., & Kumar, K. (2023). *An optimized approach of intelligent path planning of a robot manipulator using PSO algorithm.* Paper presented at the AIP Conference Proceedings.

Lian, J., Yu, W., Xiao, K., & Liu, W. (2020). Cubic spline interpolation - based robot path planning using a chaotic adaptive particle swarm optimization algorithm. *Mathematical problems in engineering, 2020*(1), 1849240.

Liu, J., Yap, H. J., & Khairuddin, A. S. M. (2024). Review on Motion Planning of Robotic Manipulator in Dynamic Environments. *Journal of Sensors, 2024*(1), 5969512.

Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications, 227*, 120254.

Luo, L.-P., Yuan, C., Yan, R.-J., Yuan, Q., Wu, J., Shin, K.-S., & Han, C.-S. (2015). Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method. *International Journal of Precision Engineering and Manufacturing, 16*, 911-917.

Ma, D., Hao, S., Ma, W., Zheng, H., & Xu, X. (2022). An optimal control-based path planning method for unmanned surface vehicles in complex environments. *Ocean Engineering, 245*, 110532.

Massaro, M., Lovato, S., Bottin, M., & Rosati, G. (2023). An optimal control approach to the minimum-time trajectory planning of robotic manipulators. *Robotics, 12*(3), 64.

Mirjalili, S. (2015). The ant lion optimizer. *Advances in engineering software, 83*, 80-98.

Oliveira, J., Oliveira, P. M., Boaventura-Cunha, J., & Pinho, T. (2017). Chaos-based grey wolf optimizer for higher order sliding mode position control of a robotic manipulator. *Nonlinear Dynamics, 90*, 1353-1362.

Sandakalum, T., & Ang Jr, M. H. (2022). Motion planning for mobile manipulators—a systematic review. *Machines, 10*(2), 97.

Shen, H., Xie, W.-F., Tang, J., & Zhou, T. (2023). Adaptive manipulability-based path planning strategy for industrial robot manipulators. *IEEE/ASME transactions on mechatronics, 28*(3), 1742-1753.

Tang, G., Tang, C., Claramunt, C., Hu, X., & Zhou, P. (2021). Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment. *IEEE Access, 9*, 59196-59210.

Wang, L., Wu, Q., Lin, F., Li, S., & Chen, D. (2019). A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators. *IEEE Access, 7*, 154331-154345.

Wei, K., & Ren, B. (2018). A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. *Sensors, 18*(2), 571.

Xuhai, Y., Wenhao, Z., Yufeng, L., Xiaochen, Q., & Qian, Z. (2023). Review of path planning algorithms for picking manipulator. *Journal of Chinese Agricultural Mechanization, 44*(5), 161.

Yang, X., Min, L., & Wang, X. (2015). A cubic map chaos criterion theorem with applications in generalized synchronization based pseudorandom number generator and image encryption. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 25*(5).

Yi, J., Yuan, Q., Sun, R., & Bai, H. (2022). Path planning of a manipulator based on an improved P_RRT* algorithm. *Complex & intelligent systems, 8*(3), 2227-2245.

Yu, H., Gao, Y., Wang, L., & Meng, J. (2020). A hybrid particle swarm optimization algorithm enhanced with nonlinear inertial weight and Gaussian mutation for job shop scheduling problems. *Mathematics, 8*(8), 1355.