Path Planning Optimization for Industrial Robots using an Adaptive Improved Differential Evolution Algorithm

Yang Zhang, Gaofeng Xu*
School of Artificial Intelligence Application, Shanghai Urban Construction Vocational College, Shanghai 201415, China
E-mail: hd_2323@126.com
*Corresponding author

Keywords: path planning, robotics, DE, IASCV, velocity steps

Received: March 6, 2025

Industrial robot path planning faces mechanical stress accumulation and motion instability caused by velocity step in complex scenarios. In this study, an adaptive optimization model is proposed, integrating the improved differential evolutionary algorithm and asymmetric S-type velocity planning. This model achieves global-local cooperative optimization through dynamic parameter coupling and velocity field feedback mechanism. Moreover, this study constructed a 10m × 6m dynamic simulation environment (including 12 static/dynamic obstacles) based on the Gazebo platform, and set parameters such as the maximum linear velocity of the robot as 2m/s and the safety distance as 0.3m. The performance of the model was also compared with traditional benchmark methods (ant colony optimization algorithm, etc.) and most advanced methods (improved dynamic window algorithm, etc.). Simulation experiments showed that the average path error of the model was reduced by 76.3%, the number of convergence iterations was reduced by 33.3%, and the AUC value was improved by 9.6%. The model's dynamic obstacle scenario planning efficiency in real-world environments was improved by 23.5%, the motion stability index was improved by 7.2%, and the trajectory tracking energy consumption was reduced by 34.7%. In summary, the adaptive population initialization strategy of IDE and the segmented plus acceleration constraints of IASCV achieve a smooth transition of the velocity profile through the dynamic replanning mechanism in the overspeed region, breaking the limitation of the decoupling of the existing trajectory optimization and dynamics constraints. In conclusion, the research model reaches the advanced level in terms of convergence speed, environmental adaptability and industrial energy efficiency, providing a highly robust solution for complex industrial scenarios.

Povzetek: Razvit je adaptivni algoritem A-I-DE za načrtovanje poti industrijskih robotov, ki z integracijo IDE in IASCV doseže večjo natančnost, točnost, stabilnost in energetsko učinkovitost v kompleksnih okoljih.

1 Introduction

As the market demand for industrial products is increasing day by day. Mobile industrial robots instead of manual labor to participate in the inspection, sorting, handling and other production processes and gradually become the main development trend of intelligent industry [1]. Optimal path planning (OPP) is an important guarantee for industrial robots to complete their work successfully, and it is also the focus of current industrial robot research. Through the optimization of path planning (PP), the robot's work efficiency and work accuracy can be significantly improved, reducing energy consumption and ensuring the efficient completion of the task [2]. The two primary categories of robot trajectory planning algorithms now in use are joint space trajectory planning (STP) and Cartesian STP. Joint STP is concerned with the change in angle of each joint, whereas Cartesian STP is concerned with the end-effector's path in the workspace [3].

However, these traditional PP algorithms have many problems. For example, linear interpolation has limited planning ability for complex paths and cannot handle higher order curves or surfaces [4]. Inverse kinematics algorithms have no solution or multiple solutions in some cases, increasing the computational complexity [5]. Dynamic PP algorithms rely on accurate robot dynamics models and are computationally intensive with high real-time requirements [6]. Consequently, while extant methods optimize the global search path, they do not integrate velocity profile smoothing, resulting in mechanical stress accumulation. Moreover, they rely on data-driven approaches and lack explicit optimization of trajectory excitation parameters. Both of them do not synergistically solve the velocity step and excitation mechanism problems, leading to limited motion stability. The motion step problem refers to the accumulation of mechanical stresses caused by sudden velocity changes. It is a common challenge in dynamic PP (not inherent to DE algorithms), but directly affects the accuracy of robot motion. Therefore, this study focuses on the underlying logic and excellent performance of differential evolution (DE), proposes improved DE (IDE), and then establishes an OPP model for industrial robots based on adaptive IDE (A-I-DE). The objective of the research is to develop a dynamic coupling framework of heterogeneous algorithms to address the timing synergy problem between global trajectory optimization and local mechanical constraints through velocity field-driven multimodal real-time feedback. The aim is to achieve the simultaneous enhancement of PP accuracy and motion stability in complex industrial scenarios. The innovation of the research lies in the closed-loop collaborative optimization mechanism of IDE and improved asymmetric s-curve velocity profile (IASCV): It dynamically identifies the trajectory risk region through velocity extremum search, combines adaptive parameter adjustment and iterative correction of multi-speed regions, and realizes the global search capability and local trajectory smoothing of DE in deep coupling. The deep coupling of DE global search capability and local trajectory smoothing breaks through the unidirectional serial connection mode between traditional DE variants and trajectory post-processing.

There are four parts to the research. The first part introduces the current worldwide research on the logic and related algorithms of PP for industrial robots. The second part establishes an accurate and efficient OPP model for industrial robots from DE algorithm and IASCV. In the third part, the proposed PP algorithm model is analyzed with examples to verify its reliability. The last part summarizes and analyzes the article comprehensively.

2 Related works

The use of robots in the production of many industrial domains has demonstrated a fast-expanding tendency as large-scale development in numerous industries has advanced [7]. OPP is one of the core performances of industrial robots, and it is also an application direction that needs to be continuously expanded and deepened in smart industry [8]. However, in complex real-world environments, the PP of robots is often accompanied by various uncertainties, so many researchers are improving this problem. Aiming at the traditional A* algorithm's slow planning speed and paths close to obstacles, Li Y et al. proposed an improved A* algorithm based on extended distance and bi-directional search, which improved the algorithm's PP efficiency [9]. For the three-dimensional environmental PP problem, Dong L proposed an improved A* algorithm with multi-sensor fusion. It significantly improved the path search efficiency and environmental adaptability by optimizing the heuristic function and the cost factor [10]. To address the problems of industrial robots in terms of coordination constraints as well as poor performance in task allocation, Hartmann VN et al. proposed a PP method based on a sampling-based bi-directional spatio-temporal path planner. The method was highly robust and scalable in the long-time dimension [11]. Sun D et al. developed a PP approach based on temporal signposts to enhance the performance of multi-robot collaborative task PP in a long-time domain, aiming to address the issue of standard algorithms' restricted scalability in terms of planning time domain [12]. A new hybrid method based on the firefly algorithm and genetic algorithm was presented by Zhang T W et al. with the goal of improving the robot's computational ability and responsiveness in PP. The firefly algorithm would easily fall into the local optimal solution (OS) and other difficulties [13]. Zan J. suggested a hybrid PP algorithm based on genetic and whale optimization algorithms to increase the effectiveness of robot PP for issues like PP in dynamic and complicated situations [14]. Liu Y et al. proposed an enhanced jumping frog swarm algorithm to address issues like traditional PP algorithms' poor search capabilities and propensity to fall into local optimization. This algorithm enhanced the quality of the OS and the stability of both low and high dimensional searches [15].

Furthermore, to address the problem of planning optimal maneuvering trajectories in uncertain environments, Chai R et al. proposed a prioritized empirical playback algorithm with noise to improve the performance of robot motion planning and collision avoidance [16]. Chen P et al. developed a PP approach based on the soft actor-critic algorithm to address issues like the difficulties of PP in dynamic environments. This method was able to successfully avoid shifting impediments in the environment and finish the planning assignment with a high success rate [17]. The time required for the robot to visit a series of objects under complex operating conditions was significantly decreased by Nawaz F et al.'s suboptimal method, which was polynomial at each time step [18]. Khlif N et al. proposed a Q-learning PP method based on an improved exploration strategy for the utilization and exploration balance in PP algorithms. The method addressed the shortcomings of the algorithm in terms of reduced cumulative rewards in exploration and falling into local optimality in utilization [19]. Aiming at the problem of ship path smoothness and disturbance adaptation, Li D proposed a PP method based on Fermat curves. This study could optimize the motion trajectories between waypoints and compensate for the side-slip angle, and improve the navigation accuracy and energy efficiency [20]. In summary, many researchers around the world have considered the problems of PP for robots in different environments and have conducted several research efforts to solve these problems, as shown in Table 1.

Table 1: Recent advances in PP for robots.

Method	Clusters	Advantages	Disadvantages	Key findings	References
Improved A algorithm with dynamic window approach (IA-DWA)	Modification of the classical algorithm	Combine global and local planning to improve obstacle avoidance efficiency; support dynamic environment adaptation.	High computational complexity and limited real-time performance.	Bidirectional search strategy reduces path redundancy and dynamic window improves obstacle avoidance flexibility.	[9]

Bidirectional spatiotemporal path planner (Bi-STP)	Sampling-based path planning	Long-time domain task robustness; multi-robot collaboration support.	High-dimensional space computational complexity increases significantly.	Spatio-temporal separation strategy reduces planning dimensions and improves scalability.	[11]
Hybrid genetic-firefly algorithm (HGFA)	Hybrid optimization algorithms	Balancing global search and local optimization to avoid local optima.	Parameter sensitivity is high and requires complex parameter tuning.	The cross mutation mechanism of genetic algorithm enhances the convergence of firefly algorithm.	[13]
Polynomial-time suboptimal multi-target algorithm (PTSMA)	Multi-intelligent body collaborative planning	Polynomial time solutions to multi-objective access problems are efficient.	Suboptimal solutions may deviate from the global optimum.	Hierarchical optimization strategy based on Markov decision process shortens the task time.	[18]
Deep reinforcement learning with prioritized experience replay (DRL-PER)	Reinforcement learning for dynamic environments	Strong real-time obstacle avoidance performance; no need for accurate environment modeling.	Depends on a large amount of training data, high hardware resource demand.	Prioritized experience playback mechanism accelerates strategy convergence, noise injection improves robustness.	[16]
Modified Q-Learning with exploration strategy (MQL-ES)	Heuristic search optimization	Balance exploration and exploitation to avoid local optima.	Convergence is slow and relies on initial parameter settings.	Improved exploration strategy reduces invalid path search and improves cumulative reward efficiency.	[19]

In addition, the accuracy and stability of PP are crucial for industrial robots to work efficiently and industrial production to be conducted in a safe and orderly manner, and its importance is self-evident. However, the majority of the aforementioned methods prioritize global path optimization or dynamic obstacle avoidance, yet they do not systematically address the mechanical stresses induced by the velocity step. Moreover, these methods lack adaptive optimization mechanisms for the trajectory excitation parameters, which hinders the integration of motion stability and path accuracy. Therefore, based on the DE algorithm, the study proposes an OPP model for industrial robots based on the A-I-DE algorithm, taking the rationality of the incentive mechanism and the stability of the motion as the optimization objectives. The study provides a comprehensive and innovative solution to solve the uncertainty and efficiency problems of industrial robot PP in real complex environments.

3 Methods and materials

This section describes in detail the A-I-DE algorithm, which consists of IDE and IASCV. Among them, IDE is responsible for the implementation of the orbit excitation mechanism, and the performance of the model in solving the motion steps is further improved when combined with IASCV.

3.1 IDE algorithm based on track excitation design for industrial robots

In the industrial robot motion model, the track excitation mechanism can guide the robot to reach the destination accurately, optimize the path selection, and realize the industrial collaborative production operation of multi-robot system. It consists of trajectory length L, maximum velocity $v_{\max}$, acceleration $a_{\max}$, and additive acceleration $j_{\max}$. Moreover, it enhances path diversity to approximate the global optimum by dynamically adjusting the above parameters, which is mathematically expressed as minimizing the number of observation matrix conditions. The motion stability is related to whether the industrial robot can maintain a stable motion state and accurately perform the scheduled tasks in the complex and changing industrial production environment. Therefore, the study proposes an OPP model for industrial robots based on the A-I-DE algorithm with respect to the track excitation mechanism and motion stability performance of industrial robots. Among them, the DE algorithm is the basic framework for realizing the track excitation of the PP model. Its algorithm flow is shown in Figure 1.

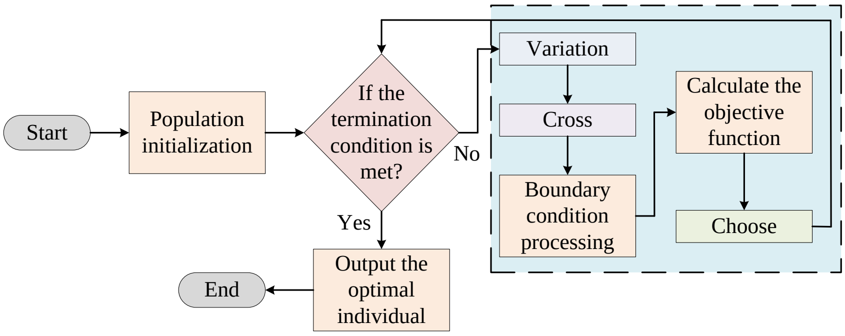

Start
Population initialization
If the termination condition is met?
No
Yes
Variation
Cross
Boundary condition processing
Calculate the objective function
Choose
Output the optimal individual
End

Figure 1: The process of the differential evolution algorithm.

In Figure 1, the DE generates candidate solutions by initializing the population, performs mutation, crossover, and selection operations sequentially, perturbs the individuals using difference vectors and greedily retains the superior solution, and iterates in a loop until convergence. The procedure first performs an initialization operation, giving each population member's dimension a value. Each individual is denoted by $x_{i,G}\left(i=1,2,\ldots NP\right)$. i denotes the number of the individual in the population. G is the number of evolutionary generations. NP represents the population size. Every population that has been randomly initiated is thought to follow a uniform distribution. It is set that the bounds of the parameter variables are $x_j^L < x_j < x_j^U$, at this time, the initialized individual assignments can be expressed by Equation (1).

$$x_{jx,0} = rand\left[0,1\right]*\left(x_j^U - x_j^L\right) + x_j^L \tag{1}$$

In Equation (1), $rand\left[0,1\right]$ denotes the real number that produces uniformity among $\left[0,1\right]$. x_j^U and x_j^L are the upper and lower limits of the parameter variables, respectively. After that, the mutation operation is performed, and for each target $x_{i,G}\left(i=1,2,\ldots NP\right)$, the mutation vector is generated as shown in Equation (2).

$$v_{i,G+1} = x_{r1,G} + F*\left(x_{r2,G} - x_{r3,G}\right) \tag{2}$$

In Equation (2), $r1$, $r2$, and $r3$ are randomly selected individual serial numbers. Each ordinal number is different from each other and cannot be the same as the vector ordinal number i , which must satisfy $NP \geq 4$. The operator for variation The scaling of the deviation variable is regulated by the real constant factor $F \in \left[0,2\right]$. DE adds a crossover operation to the interference parameter vectors to make them more diverse. The test vector is now displayed in (3).

$$\begin{cases} v_{i,G+1} = \left(u_{1i,G+1}, u_{2i,G+1}, \ldots u_{Di,G+1}\right) \\ u_{ji,G+1} = v_{ji,G+1} \quad randb\left(j\right) \leq CR \ or \ j = rnbr\left(i\right) \\ u_{ji,G+1} = x_{ji,G+1} \quad x! = 0 \ and \ j! = rnbr\left(i\right) \end{cases} \tag{3}$$

In Equation (3), The random number generator that generates $\left[0,1\right]$ has an j th estimate, which is represented by the $randb\left(j\right)$. To guarantee that $u_{i,G+1}$ receives at least one parameter from $v_{i,G+1}$, $rnbr\left(i\right) \in \left(1,2,\ldots,D\right)$ stands for a randomly chosen sequence, where D is the population dimension. CR denotes the crossover operator, which takes the value range $\left[0,1\right]$. The selection operation is performed after the variation and crossover operations are completed. In this process, the test vectors are compared with the target vector $x_{i,G}$ according to the greedy criterion and the superior one goes to the next generation. This process is inter-individual comparison, not group competition. If a solution outside the feasible domain, i.e., $u_{ji,G+1} < x_j^L$ or $u_{ji,G+1} > x_j^U$, is compiled during the mutation process, the operation shown in Equation (4) is performed.

$$u_{ji,G+1} = rand\left[0,1\right]*\left(x_j^U - x_j^L\right) + x_j^L \tag{4}$$

In Equation (4), the solution generated by the mutation is always within the bounds of the parameter variables by re-initializing the assignment. In summary, population initialization, mutation and crossover operations are the key to achieve computational accuracy and convergence speed [21]. Therefore, this study improves the three in the framework of DE and proposes the IDE algorithm. First, the population initialization is improved, and the distance between the optimal individuals and the initial population's members influences the DE algorithm's computation time. Random people don't have direction. The initial population is optimized by the optimal worst-case backward learning strategy to determine the direction of feasible solutions. Therefore, the initial individuals are close to the OS, which speeds up the convergence of the algorithm while maintaining the global search capability. The mathematical definition of the reverse learning strategy is shown in Equation (5).

$$x_{i,\text{opp}}^{(j)} = x_j^U + x_j^L - x_i^{(j)} \tag{5}$$

In Equation (5), $x_{i,\text{opp}}^{(j)}$ is the reverse solution of individual x_i in dimension j . x_j^U and x_j^L are the upper and lower bounds of the parameters, respectively. The study merges the original population P with the inverse population P_{opp} , and retains the highly adapted individuals as the optimized initial population. Second, the variation operator F 's is improved. The operator is too small, which reduces the population diversity. If it is too large, it will reduce the search efficiency and affect the accuracy of the OS. This study designs the adaptive variation operator, which is calculated as shown in Equation (6).

$$\lambda = e^{1-\frac{g_{\max}}{g_{\max}+1-g}}, F = F_0 * 2^{\lambda} \tag{6}$$

In Equation (6), F_0 is the initial mutation operator. $g_{\max}$ is the maximum evolutionary algebra. g is the current evolutionary generation. The initial adaptive variation operator, $F = 2F_0$, can maintain individual diversity at the beginning. As the number of evolutionary generations increases, F gradually decreases and approaches F_0. The linear decay design is predicated on a balance between global exploration and local exploitation of the algorithm. This is achieved by dynamically adjusting the perturbation strength, thereby avoiding local optimization in the early stages and suppressing oscillations in the later stages. In addition, the value of the crossover operator CR will directly affect the convergence time of the algorithm, resulting in missing the global OS or reducing the efficiency of the algorithm. Similarly, the study designs adaptive crossover operator, which is calculated as shown in Equation (7).

$$CR = CR_{\min} + \frac{k}{g_{\max}}\left(CR_{\max} - CR_{\min}\right) \tag{7}$$

In Equation (7), $R_{\max}$ and $R_{\min}$ denote the maximum and minimum values of the crossover operator, respectively. In terms of specific parameter settings, the study sets the initial value of the adaptive variance factor of IDE to 0.5 and gradually decreases to 0.3 with the number of iterations. The dynamic adaptive adjustment range of the crossover rate is 0.3-0.9. The adaptive variance factor enhances the global search ability in the initial stage and improves the convergence accuracy in the later stage. The dynamic crossover rate balances the diversity of the populations with the convergence speed. Through the hyperparameter adaptive dynamic adjustment mechanism in Equations (5) and (6), the model is able to avoid the subjectivity of manual parameter tuning, while adapting to the convergence requirement in different environments to ensure the robustness of the algorithm. Improving the identification accuracy and reducing the condition number of the observation matrix are the main goals of optimizing the PP algorithm's excitation trajectory criterion. Therefore, the flow of IDE to optimize the trajectory excitation parameters is shown in Figure 2.

As illustrated in Figure 2, IDE incorporates adaptive variance factors and dynamic crossover rates, integrates optimal backward learning to initialize the population, generates incentive parameters through constraint validation and cyclic optimization, and iteratively updates them until the termination conditions are met. First, IDE initializes the population, and the incentive track parameter is the population dimension. Then, the individual objective value of the current population is calculated, which is quantified by a weighted multi-objective function as shown in Equation (8).

$$f_{\text{obj}} = w_1 \cdot \frac{L}{L_{\max}} + w_2 \cdot \frac{T}{T_{\max}} + w_3 \cdot \sum| \frac{d^2\theta}{dt^2}| + w_4 \cdot \frac{1}{d_{\min}} \tag{8}$$

In Equation (8), L is the planning path length. $L_{\max}$ is the maximum theoretical path length of the scene. T is the path execution time. $T_{\max}$ is the maximum time threshold allowed by the task. $\sum| \frac{d^2\theta}{dt^2}|$ is the sum of the absolute values of the joint angular acceleration, which measures the smoothness of the trajectory. $d_{\min}$ is the minimum distance between the path and the obstacle. Furthermore, the weight coefficients satisfy $w_1 + w_2 + w_3 + w_4 = 1$, and the study is set to $w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, w_4 = 0.1$. If the current individual does not meet the constraints, the objective value is set. If it does, compare the initial value and the current individual's objective function value. If the initial value is greater, the initial value of the individual is updated, otherwise genetic operations are performed to generate new individuals, i.e. mutation, crossover, selection. If the new individual value is greater than the original individual value, the population is reinitialized. At this point, the number of evolutionary generations is increased by one, and the cycle optimization is restarted until the OS for the population is obtained, or the evolution reaches the maximum number of generations.

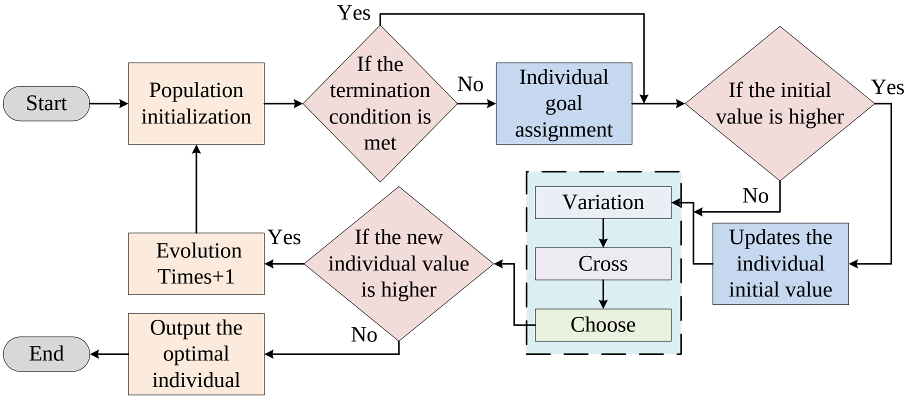

Start
Population initialization
If the termination condition is met
Yes
No
Individual goal assignment
If the initial value is higher
Yes
No
Variation
Cross
Choose
Updates the individual initial value
If the new individual value is higher
Yes
No
Evolution Times+1
Output the optimal individual
End

Figure 2: The process of optimizing the trajectory excitation parameters in the IDE.

3.2 A-I-DE path planning modeling for motion step problems

Orbital excitation mechanisms and motion stability are at the core of learning and adaptive path optimal planning for industrial robots. The study proposes that IDEs optimize the design of the excitation mechanism. However, in industrial dynamic environments or collaborative production operations, the operating speed of industrial robots often undergoes jumping changes. This process will generate huge mechanical stress, which affects the motion accuracy and stability of industrial robots, and thus poses a threat to industrial production [22]. IASCV makes the velocity profile of industrial robots smoother by processing the trajectory in segments, reduces the impact when starting and stopping, and improves the stability and accuracy of the motion. Therefore, the research addresses the velocity step problem, combines IASCV to further optimize the IDE, and proposes A-I-DE. Among them, the curve model of IASCV is shown in Figure 3.

In Figure 3, the IASCV is predicated on the enhancement of the ASCV, with the absence of mid-beginning and end-end acceleration. The red point demarcates the transition moment of the acceleration phase to ensure a seamless transition of motion. The green point identifies the critical state of uniform and variable speeds to regulate the magnitude of the velocity step. Furthermore, the blue point calibrates the point of cumulative inversion of the displacement to inhibit motion oscillations. IASCV restricts the second-order derivative continuity of the velocity profile by adding acceleration constraints to the segmented trajectory and dynamic time allocation. This ensures that the acceleration in each phase transitions smoothly to avoid mechanical shocks. The parameters when IASCV runs are mainly trajectory length L, initial velocity v_s and final velocity v_e, maximum acceleration $a_{\max}$, maximum deceleration $-a_{\max}$, initial acceleration a_s, and final acceleration a_e. It constructs a curve model by using the parameters and outputs, and the model parameter changes can be calculated based on the change curves when the curve trajectory runs. For example, when $0 \le t \le t_1$, the parameter changes are shown in Equation (9).

$$\begin{cases} a_{(t)} = a_s + j_{\max} * \tau_1 \\ v_{(t)} = v_s + a_s * \tau_1 + 0.5 j_{\max} * \tau_1^2 \\ S_{(t)} = v_s * \tau_1 + 0.5 a_s * \tau_1^2 + (j_{\max} * \tau_1^3)/6 \end{cases} \quad (9)$$

In Equation (9), $j_{\max}$ denotes the maximum additive acceleration. $t_k = \sum_{i=1}^{k} T_i, k = 1, 2, \ldots, 7$ denotes the elapsed point moments of each motion phase of the trajectory. τ denotes the local time coordinates, where $\tau_1 = t, \tau_k = t - t_{k-1}, k = 2, \ldots 7$. Similar to ASCV, IASCV finds the values of the running time T for different phases based on the input parameters. Its algorithm flow is shown in Figure 4.

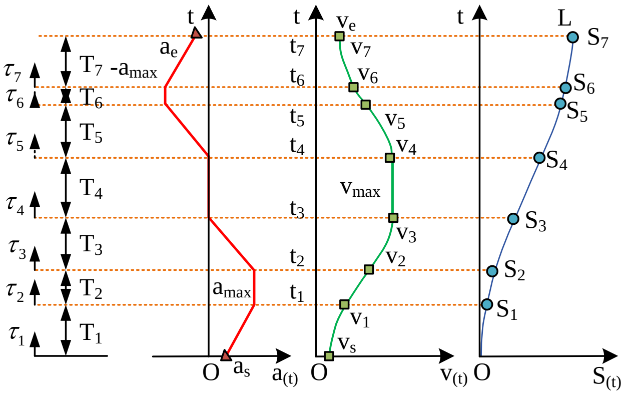


Figure 3: Improved asymmetric S-curve model.

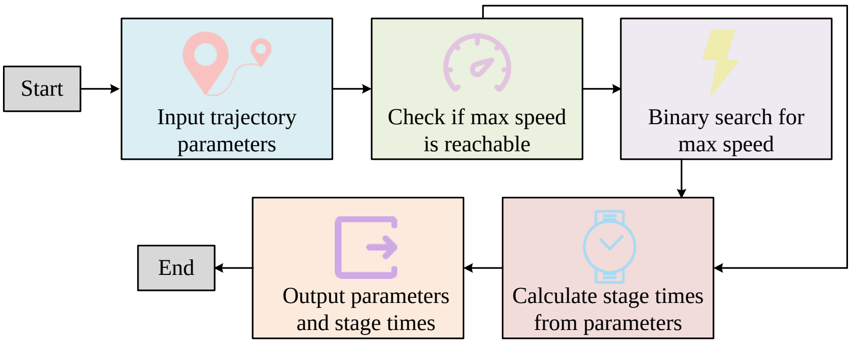

Start
Input trajectory parameters
Check if max speed is reachable
Binary search for max speed
Calculate stage times from parameters
Output parameters and stage times
End

Figure 4: The process of improved asymmetric s-curve velocity profile.

In Figure 4, the known trajectory parameters, including L , v_s , v_e , etc., are firstly inputted into the algorithm. After that, the segmental displacement trajectory $L = L_a + L_d$ is calculated for the initial and terminal velocities to reach the maximum uniform velocity $v_{\max}$. In this case, L_a and L_d are calculated as shown in Equation (10).

$$\begin{cases} L_a = \dfrac{0.5\left(v_{\max}^2 - v_s^2\right)}{a_{\max}} + \dfrac{0.5\left(v_{\max}^2 + v_s\right)}{j_{\max}} & v_{\max} - v_s \geq \dfrac{a_{\max}^2}{j_{\max}} \\ L_a = \left(v_{\max} + v_s\right) * \left(\dfrac{v_{\max} - v_s}{j_{\max}}\right)^{0.5} & v_{\max} - v_s < \dfrac{a_{\max}^2}{j_{\max}} \\ L_d = \dfrac{0.5\left(v_{\max}^2 - v_e^2\right)}{d_{\max}} + \dfrac{0.5\left(v_{\max}^2 + v_e\right)}{j_{\max}} & v_{\max} - v_e \geq \dfrac{d_{\max}^2}{j_{\max}} \\ L_d = \left(v_{\max} + v_e\right) * \left(\dfrac{v_{\max} - v_e}{j_{\max}}\right)^{0.5} & v_{\max} - v_e < \dfrac{d_{\max}^2}{j_{\max}} \end{cases} \tag{10}$$

In Equation (10), $d_{\max}$ is the maximum acceleration in the deceleration phase. If the shortest trajectory required to change v_e from v_s is less than L , it means that v_s and v_e satisfy the planning conditions. Then, based on the displacement length and the actual v_s and v_e , the actual $v_{\max}$ is calculated by dichotomization and proceeds to the next step. Otherwise, the bisection operation is skipped, and the different stages running time T is calculated directly based on the actual track parameters. Then the different phase running times T are summed up to obtain the shortest total running time. Finally, the actual values of v_s , v_e , and $v_{\max}$ are output, as well as the values of each phase and the shortest total running time. For IASCV, the study sets the maximum additive acceleration to be set at 1000 cm/s3, the maximum acceleration to be 500 cm/s2, and the initial acceleration to be 50 cm/s2 during PP. The IASCV parameter ensures the smooth transition of the velocity profile by constraining the upper limit of additive acceleration and acceleration to reduce the mechanical shocks and improve the stability of the motion. On this basis, the study combines IASCV with IDE to propose the A-I-DE PP model. Its algorithm flow is shown in Figure 5.

In Figure 5, A-I-DE achieves collaborative optimization through a bidirectional closed-loop feedback mechanism. In this mechanism, IDE generates path parameters, and then IASCV detects velocity anomalies in real time and triggers dynamic adjustment of the parameters with feedback. Both of these processes are iteratively corrected between path generation and velocity smoothing. This ensures a dynamic balance between global optimization and local motion constraints. That is, A-I-DE is divided into two main components: the computation of the overspeed region and the replanning of the trajectory. The former is used for the preliminary detection of individual overspeeds, while the latter is based on multi-area computation to realize accurate trajectory correction, taking into account both global safety and local optimization needs. The A-I-DE algorithm first generates the initial path through IDE PP and identifies the peaks of velocity mutation through the velocity extreme value search module. The overspeed determination link is responsible for detecting any potential violations of the dynamics constraints. In the event of overshoots, the continuous risk period is divided into single or multiple overspeed regions. The trajectory correction module is tasked with dynamically adjusting the excitation parameters and relaying these adjustments back to the IDE for iterative optimization. The corrected trajectory is looped again until all overspeed regions are eliminated, and the final output is a smooth velocity profile. This sequence of operations helps to ensure the safety and efficiency of the robot's motion while optimizing time.

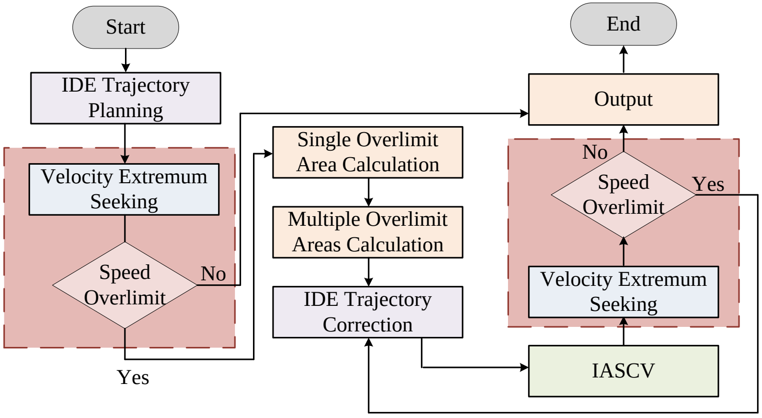

Start
IDE Trajectory Planning
Velocity Extremum Seeking
Speed Overlimit
No
Yes
Single Overlimit Area Calculation
Multiple Overlimit Areas Calculation
IDE Trajectory Correction
IASCV
Velocity Extremum Seeking
Speed Overlimit
No
Yes
Output
End

Figure 5: The structure of A-I-DE.

4 Results

For verifying the effectiveness and superiority of the A-I-DE algorithm proposed in the study, the study synthesizes the theoretical foundation and algorithm analysis mentioned above, and carries out simulation simulation running experiments and real environment motion experiments on the industrial robot model under different algorithms. Moreover, the experimental results are analyzed in detail to compare their PP accuracy and motion stability.

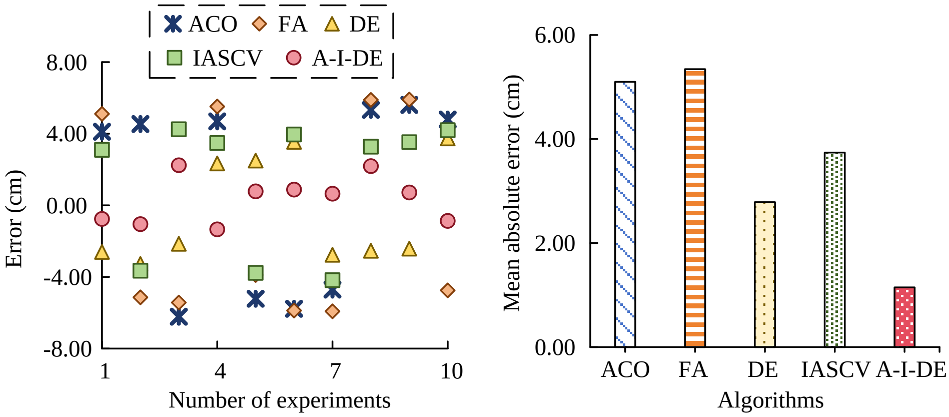
4.1 Simulation simulation operation experiment

The industrial robot PP model's application environment is combined with the simulation running experiments to provide a compatible system development environment that is separated into a software environment and a hardware environment. The detailed configuration is shown in Table 2.

In Table 2, the study selects Windows 10 as the operating system and the robot operating system as the software platform. Gazebo 3D physical simulation platform is used to build the simulation scene, and the industrial robot model is constructed through the unified robot description format. The simulation environment is a 10m×6m rectangular space containing 12 obstacles: 8 static cylinders (diameter 0.5-1.2m) and 4 dynamic moving cubes (side length 0.8m). The robot has a maximum linear velocity of 2m/s, a safety distance threshold of 0.3m, and a joint torque limit of 80%. Moreover, the constraint values are set based on the robot dynamics model and industry standards. Furthermore, the study selects ant colony optimization algorithm, firefly algorithm, DE, and IASCV as the comparison methods and named ACO, FA, DE, and IASCV respectively. A-I-DE is taken as the object of the study and named A-I-DE. These baseline methods can systematically verify the comprehensive performance of this algorithm in multi-dimensional performance metrics. They represent typical schemes in the fields of traditional heuristic search, population intelligence optimization, and speed smoothing, respectively, and cover the mainstream technological routes of PP. Moreover, the pheromone volatility of ACO is 0.5 and the heuristic factor is 2.0.The attractiveness of FA is 0.9 and the step size is 0.3.The variance factor of DE is 0.6 and the crossover rate is 0.8.The maximum acceleration of IASCV is 500 cm/s² and the acceleration of additive acceleration is 1000 cm/s³. Moreover, the parameters are set based on the standard literature recommended values to ensure the fairness. In this study, two points are firstly selected as the starting point (0,0) and the target point (5m, 3m) of the industrial robot in the complex environment simulated by the simulation. Then, the robot is controlled to start from the starting point and finally reach the target point, which is repeated 10 times with the confidence interval set to 95%. Among these, Figure 6 displays the outcomes of the distance inaccuracy between the industrial robot and the TP point once it has stopped operating.

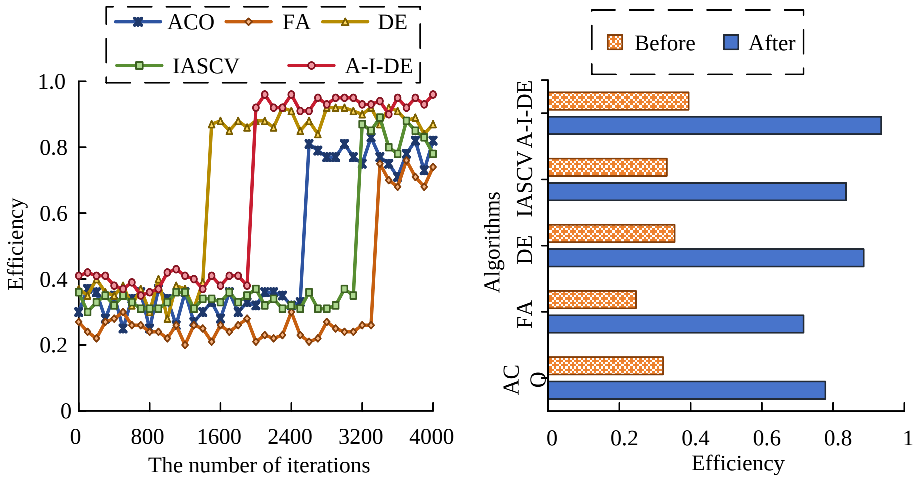
Table 2: System development environment.

	System development environment
Hardware environment	Intel(R) Core(TM) i7-8700 Installed memory 16.00GB
	NVIDIA RTX2080Ti
Software environment	Windows 10 is running on a 64-bit operating system
	Robot Operate System
	Gazebo
	Unified Robot Description Format



(a) Target point error (b) Mean absolute error

Figure 6: Difference in target point error.

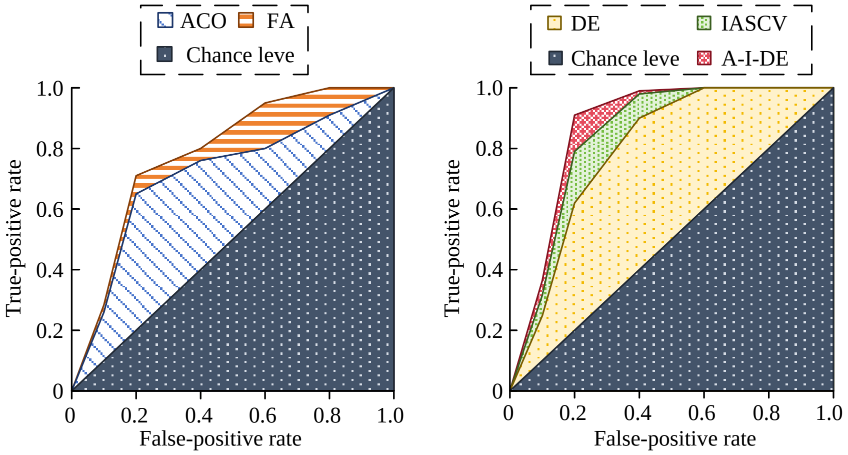


(a) Differences in learning performance (b) Average precision differences

Figure 7: Iterative learning performance experiments.

In Figure 6(a), when the industrial robot stops moving, ACO has the largest change in error from the TP. The maximum error magnitude of change is 2.11 cm, followed by FA. Among them, the error fluctuates between 3.89 cm and 5.92 cm. The errors of DE and IASCV varied less, with the maximum change magnitude of 1.59 cm for both. The error of A-I-DE varies the least, with the error fluctuating between 0.65 cm and 2.24 cm. In Figure 6(b), in terms of the average error, A-I-DE has the smallest error of 1.15 cm. It is followed by DE with an average error of 2.79 cm. IASCV has an average error of 3.74 cm. ACO and FA have the largest errors, with a mean value of 5.10 cm and 5.34 cm, respectively. The outcome displays that the accuracy of the A-I-DE's PP is much higher than the traditional algorithm. In addition, the study selects the operational dataset as the training set and conducts iterative experiments on the model. The study verifies its learning performance by comparing its PP efficiency (path success rate).The results of which are shown in Figure 7.

In Figure 7(a), DE completes convergence at 1500 iterations and has the fastest convergence rate. ACO completes convergence at 2600 iterations. FA and IASCV converge the slowest, completing convergence at 3400 and 3200 iterations, respectively. Whereas, A-I-DE has the second fastest convergence rate after DE, completing the convergence at 2000 iterations. In Figure 7(b), before and after convergence, FA has the lowest average efficiency of 0.25 and 0.72, respectively. It is followed by ACO, with average efficiencies of 0.32 and 0.78 before and after convergence, respectively. The efficiencies before and after convergence of DE and IASCV are in the ranges of 0.28-0.40 and 0.78-0.92, respectively. Whereas A-I-DE has the highest efficiency before and after convergence with mean values of 0.39 and 0.94 respectively. Compared to the fast local convergence of DE, A-I-DE prioritizes the strengthening of the global search capability and expands the exploration area of the solution space through dynamic parameter adjustment. This ensures that the local optimum in complex PP is avoided, and the moderate iteration cost is traded for high accuracy and stability. For the overall PP performance of the model, the study conducts receiver operating characteristic curve (ROC) and area under curve (AUC) analysis. The results are shown in Figure 8.



(a) ROC for M1 and M2 (b) ROC for DE, IASCV and A-I-DE

Figure 8: Analysis of ROC for different algorithms.

Table 3: Differences in computational complexity.

Method	ACO	FA	DE	IASCV	A-I-DE
Single planning time（ms）	15.2	12.7	9.4	8.1	6.3
peak memory usage（MB）	48.3	40.6	35.2	30.5	26.8
CPU utilization（%）	82.5	75.8	68.3	62.4	58.9

Table 4: Comparison of model motion stability.

Method	TTE (cm)	SSF (time)	ASF (time)	PSI (%)	TS (°/m)
ACO	6.32	9	12	68.5	8.2
FA	6.78	8	11	70.2	7.8
DE	4.15	6	8	82.7	5.4
IASCV	3.89	5	7	88.3	4.9
A-I-DE	1.05	2	3	95.6	2.1

In Figure 8, in the ROC, the false positive rate (FPR) grows fastest at 0-0.2 for the true positive rate (TPR). At 0.2-0.6, the growth of TPR becomes slower and gradually stabilizes after 0.6. At the same FPR, ACO had the lowest TPR, followed by FA. At this time, the TPRs of DE and IASCV are slightly higher than that of FA, but the TPR of A-I-DE is the largest. In addition, there is a similar trend of difference in the AUC values of the different models. IASCV has an AUC value of 0.85, whereas A-I-DE has the highest AUC value of 0.87. The AUCs of DE and FA are 0.80 and 0.78, respectively, and the AUC of ACO is the lowest value of 0.72. The benchmark value of AUC in PP tasks is typically 0.8-0.9 [23], and experiments show that A-I-DE has high reliability and outperforms most traditional algorithms.

To further explore the computational complexity of the proposed algorithm, the study selects the single planning elapsed time (ms), peak memory occupancy (MB), and CPU utilization (%) metrics for experimentation. The results are shown in Table 3.

In Table 3, the single planning elapsed time of A-I-DE (6.3ms) is 58.5% lower than that of ACO (15.2ms), due to the low computational overhead of the adaptive mutation and crossover operators. The peak memory usage (26.8MB) is 23.9% lower than that of DE (35.2MB), due to the optimization of IDE's population initialization to reduce redundant storage. The CPU utilization (58.9%) is significantly lower than the comparison methods, thanks to IASCV's speed smoothing mechanism, which reduces the frequency of real-time replanning. The results show that A-I-DE has significant advantages in scenarios with limited computational resources.

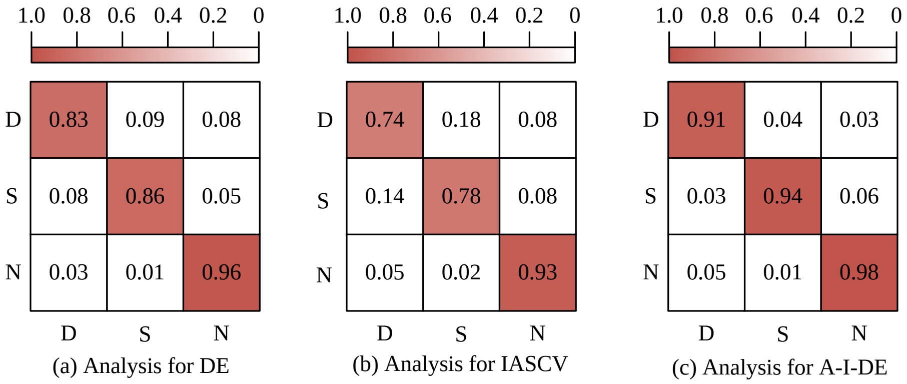
In addition, to verify the motion stability of the model in the simulated environment, the study selects trajectory tracking error (TTE), speed step frequency (SSF), acceleration step frequency (ASF), posture stability index (PSI) and trajectory smoothness (TS) metrics for comparison. The PSI and TS metrics are compared and the results are shown in Table 4.

In Table 4, the TTE of A-I-DE (1.05 cm) is 83.4% lower than that of ACO (6.32 cm), due to the closed-loop parameter coupling of IDE and IASCV, which suppresses the cumulative displacement deviation. Its SSF (2 times) is only 33% of that of DE (6 times). ASF (3 times) is 57% less than that of IASCV (7 times), attributed to the adaptive parameters of IDE suppressing the path abruptness. In addition, the PSI of the A-I-DE is 95.6%, which is significantly higher than that of the IASCV (88.3%) due to the IDE's optimization of path excitation parameters to reduce fuselage inclination. Trajectory smoothness (TS=2.1°/m) is improved by 61.1% compared to DE (5.4°/m) thanks to the segmentation plus acceleration constraints of IASCV, confirming the overall advantages of A-I-DE in terms of kinematic stability and dynamic performance.

4.2 Real environment motion experiments

Robot operation in simulation is an important standard for measuring the operation of industrial robots. However, influenced by many uncontrollable factors, the operation status of industrial robots in the actual environment often differs from the simulation simulation. Therefore, the study uses an industrial production workshop as the experimental environment, with workshop equipment as static obstacles and workers and other industrial robots as dynamic obstacles, to conduct experiments. Each set of experiments is repeated 10 times to verify the stability. Moreover, the study selects Universal Robots UR5 collaborative robot (public parameters, https://www.universal-robots.com/products/ur5-robot/) as the actual test robot. The starting point is located at the origin of the workshop coordinates (0,0), the target point is set as the material loading and unloading area (12.5m, 8.2m), and the theoretical shortest path between the two points is 15.3 m. The actual running path is recorded in real time by the laser tracking instrument (API T3) and the accumulated mileage is calculated, and the deviation of the path length is quantified by the Euclidean distance integration method. In addition, the study selects DE and IASCV, which perform better in simulation and simulation running experiments, as the comparison method, and A-I-DE is used as the object of the study. The study firstly verifies the operational efficiency of the model when facing no obstacles (No, N), static obstacles (Static, S), and dynamic obstacles (Dynamic, D). The static obstacles are set as 8 fixed cylinders, and the dynamic obstacles are set as 5 randomly moving spheres (with operational constraints consistent with those of the robot). Furthermore, the experiments use 10 sets of random seeds to generate the initial positions of the obstacles to ensure the generalizability of the results. Figure 9 displays the outcomes of the experiment.

In Figure 9, the model has the highest PP efficiency in an accessible environment, followed by static barriers, and the model has the lowest efficiency in a dynamic barrier environment. In Figure 9(a), the efficiency of DE is 0.96 in the accessible environment, and 0.83 and 0.86 in static and dynamic barriers, respectively. In Figure 9(b), IASCV has the lowest efficiency of PP in the real environment. Its planning efficiency is 0.93, 0.78, and 0.74 in the three barrier environments, respectively. In Figure 9(c), the planning efficiency of A-I-DE in all three obstacle environments is high, and the overall efficiency is above 0.91. Among them, the model has the highest PP efficiency in the barrier-free environment, reaching 0.98. This means that the PP success rate of A-I-DE reaches 98%, indicating a nearly redundancy-free search, which significantly improves the efficiency of continuous operation in industrial scenarios. Moreover, the critical tilting angle of conventional industrial robots is about 25°, and they will fall down if the critical tilting angle is exceeded. Therefore, the study selects 10 time points during the movement of the industrial robot. Moreover, the angle ∂ between the body of the industrial robot and the horizontal plane at that time point is used as the basis for judgment. The smaller the tipping angle during the motion of the industrial robot, the higher the motion stability. Table 5 displays the findings.

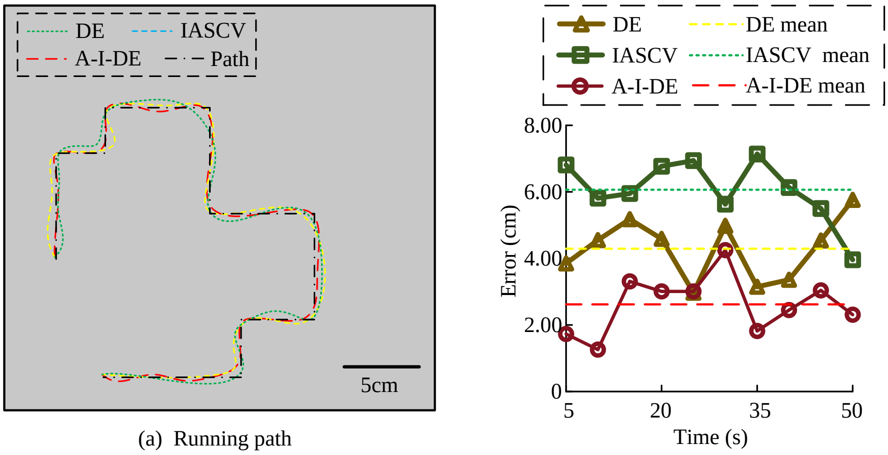


(a) Analysis for DE (b) Analysis for IASCV (c) Analysis for A-I-DE

Figure 9: Confusion matrix analysis for operational efficiency.

Table 5: Athletic stability of different algorithms running.

Points	Dump angle (°)			Motion stability (%)		
	DE	IASCV	A-I-DE	DE	IASCV	A-I-DE
1	11.85	5.90	4.82	52.60	76.40	80.72
2	13.76	7.75	7.75	44.96	69.00	69.00
3	12.74	8.75	4.50	49.04	65.00	82.00
4	12.90	8.16	6.17	48.40	67.36	75.32
5	13.78	7.92	7.42	44.88	68.32	70.32
6	11.48	8.85	6.73	54.08	64.60	73.08
7	14.10	5.95	4.63	43.60	76.20	81.48
8	12.31	6.83	4.52	50.76	72.68	81.92
9	13.63	6.41	6.45	45.48	74.36	74.20
10	11.51	6.17	7.73	53.96	75.32	69.08
Mean	12.81	7.27	6.07	48.78	70.92	75.71



(a) Running path

(b) Path error

Figure 10: Path trajectory and error.

In Table 5, DE has the largest dumping angle ranging from 11.48°-14.1°0 with a mean value of 12.81°. IASCV has a range of dumping angle from 5.90° to 8.85° with a mean value of 7.27°. The mean dumping angle of A-I-DE is 6.07° and the range fluctuates between 4.50°-7.75°. In addition, according to the calculated equation $Ms = \frac{25-\partial}{25} * 100\%$, the A-I-DE has the highest motion stability ranging from 69.00% to 82.00% with a mean value of 75.71%. This is followed by IASCV with stability ranging from 64.60%-76.40% with a mean stability of 70.92%. The stability of DE is the lowest, with an overall range of 43.60%-54.08% and a mean value of 48.78%. The experimental results show that the robot under A-I-DE control operates more stably with a low probability of error in real environment applications. The study also examines the discrepancy between the industrial robot's real motion path and its preset path in real time to confirm the model's correctness and stability in real-world operations. Figure 10 displays the outcomes of the experiment.

In Figure 10(a), the IASCV has a large offset relative to the preset trajectory in the control motion, and the actual motion trajectory is curved. DE has a smaller offset relative to the preset trajectory, while the actual trajectory of A-I-DE fits the preset path better. In Figure 10(b), the errors of the actual paths of DE and IASCV with respect to the preset paths are between 2.97cm-5.76cm and 3.96cm-7.14cm, respectively. Their mean values are 4.29cm and 6.06cm, respectively. At this time, the errors of the paths of A-I-DE are between 1.26-4.25cm, and the mean error is 6.62. Moreover, except for some external time points, the errors of the other time points are smaller than those of DE and IASCV. The experimental results show that the industrial robot under A-I-DE control has the best operational accuracy and stability in the real environment. It can be concluded from the aforementioned findings that the study on the improvement of the DE algorithm in the operations of initializing population, mutation, and crossover, as well as combining it with the IASCV algorithm for optimization and other improvement measures, is effective. The implementation of these measures can lead to enhanced performance of industrial robots in terms of efficiency and stability of PP.

Moreover, to validate the comprehensive performance advantages of the proposed methods of the study in complex environments, the study selects methods from the literature [9], [11], [13], [16], [18], and [19] for comparison with A-I-DE. These methods are IA-DWA, Bi-STP, HGFA, PTSMA, DRL-PER, and MQL-ES and the results are shown in Table 6.

Table 6: Comparison of A-I-DE and recent advances.

Method	IA-DWA	Bi-STP	HGFA	PTSMA	DRL-PER	MQL-ES	A-I-DE
Convergence iterations	2600	3400	2800	2100	3200	2500	2000
Path error (cm)	5.1	3.74	5.34	4.29	6.06	4.15	1.15
Motion stability (%)	48.78	70.92	65.2	72.5	68.3	74.8	75.71
Computational efficiency (times/s)	38.5	22.1	45.3	50.6	18.9	40.7	65.4
AUC	0.72	0.85	0.78	0.82	0.87	0.83	0.94

Table 7: Comparison of model integrated energy efficiency performance.

Method	MOEC （W·h）	TTEC （J/m）	HRPEC （W）	SPTEC （W·s）	TSEC （kW·h）

IA-DWA	12.5	8.2	95.3	1.9	1.28
Bi-STP	10.8	7.5	88.7	1.7	1.15
HGFA	11.2	7.8	92.1	1.8	1.21
PTSMA	9.6	6.9	84.5	1.5	0.98
DRL-PER	15.3	9.1	102.4	2.3	1.45
MQL-ES	8.9	6.3	78.9	1.3	0.85
A-I-DE	7.2	5.1	72.4	1	0.68

As shown in Table 6, the number of convergence iterations (2000) of A-I-DE is significantly lower than that of the comparison methods (2100-3400), thanks to the dynamic parameter adjustment strategy of the adaptive variational and crossover operators. Its path error (1.15 cm) is 72.3% lower than that of the optimal comparison method (4.15 cm for MQL-ES), which is due to the IDE's orbit excitation mechanism that optimizes the parameter. Motion stability (75.71%) is better than other algorithms because IASCV's smooth velocity profile effectively suppresses mechanical stress. The computational efficiency (65.4 times/sec) and the TPR (AUC=0.94) are attributed to the improvement of the population initialization and the dynamic replanning mechanism in the overspeed range, which avoids the redundant search.

Finally, the study validates the comprehensive model by comparing the research model with the latest work of model operation energy consumption (MOEC), trajectory tracking energy consumption (TTEC), hardware resource peak energy consumption (HRPEC), single planning time energy consumption (SPTEC) and total system energy consumption (TSEC). Hardware resource peak energy consumption (HRPEC), single planning time energy consumption (SPTEC), and Total system energy consumption (TSEC) are introduced to verify the comprehensive energy efficiency performance and industrial applicability of the model. The experimental results are shown in Table 7.

In Table 7, the model running energy consumption of A-I-DE (7.2W-h) is reduced by 19.1% compared to the latest work MQL-ES (8.9W-h) due to the adaptive parameters to reduce the redundant computation. The trajectory tracking energy consumption (5.1J/m) is reduced by 18.9% thanks to the IASCV speed smoothing to reduce the frequent motor starts and stops. The peak power consumption of hardware resources of A-I-DE (72.4W) is 29.3% less than DRL-PER (102.4W) due to IDE population optimization to reduce memory and CPU load. The energy consumption (1.0 W-s) of a single design is only 52.6% of IA-DWA, and the total system energy consumption (0.68 kW-h) is optimized by 20.0% compared to MQL-ES (0.85 kW-h). The experimental results confirm the energy saving advantages of the A-I-DE.

5 Discussion and conclusion

The IDE algorithm and the IASCV algorithm were integrated in the study to address the issues of low efficiency and poor motion stability of conventional PP algorithms for industrial robots. The A-I-DE PP model was then created. The model improved the accuracy and stability of industrial robot operation by optimizing the track excitation mechanism and solving the velocity step problem. The experimental results revealed that the average absolute error of A-I-DE was 1.15cm in the simulation simulation experiments, with an error range between 0.65cm-2.24cm. The errors of the other algorithms ranged from 2.15cm-5.34cm. When iterating in the operational data set, the average efficiency of A-I-DE before and after convergence was 0.39 and 0.94, respectively. Whereas the efficiencies of other algorithms before and after convergence were in the range of 0.20-0.43 and 0.68-0.92, respectively. Moreover, in the ROC analysis, the AUC value of A-I-DE was 0.87, whereas the other algorithms' AUC values were in the range of 0.70-0.85. In addition, in the actual detection, the detection efficiency of IRAU-DSC was above 0.91, while the efficiency of other algorithms ranged from 0.71-0.85. In addition, in the actual environmental motion experiments, the efficiency of A-I-DE in different environments ranged from 0.91-0.98, while the efficiency of other algorithms ranged from 0.74-0.96. However, the average tipping angle and average stability of A-I-DE were 6.07° and 75.71%, respectively, during the actual motion. For other algorithms, the dumping angle and stability were in the range of 5.90°-14.10° and 43.60%-76.40%, respectively. Moreover, the real-time error of the path of A-I-DE was between 1.26cm-4.25cm. The average errors of other algorithms were 4.29cm and 6.06cm, respectively. In conclusion, the study offers practical applications for enhancing the precision and reliability of industrial robot PP. However, the PP model proposed in the study focuses on single-segment paths, which may lead to cumulative errors at trajectory breakpoints due to the temporal constraint transfer characteristic of the parameter coupling mechanism in the continuous operation of multiple goal points. Additionally, the model suffers from the problem of nonlinear growth of computational complexity in complex topological environments. To address this limitation, in future work, the dynamic parameter decoupling strategy of trajectory articulation between target points, combined with incremental learning to optimize the subpath incentive mechanism, will be used to construct a hierarchical optimization framework to achieve global stability control of multistage continuous planning.

References

[1] Fushu Luan, Xinhui Yang, Yang Chen, and Paulo José Regis. Industrial robots and air environment: A moderated mediation model of population density and energy consumption. Sustainable Production and Consumption, 30(1):870-888, 2022. https://doi.org/10.1016/j.spc.2022.01.015

[2] Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, and Enrique Fernandez-Blanco. A review of artificial intelligence applied to path planning in UAV swarms. Neural Computing and Applications, 34(1):153-170, 2022. https://doi.org/10.1007/s00521-021-06569-4

[3] Lin Xu, Maoyong Cao, and Baoye Song. A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm. Neurocomputing, 473(1):98-106, 2022. https://doi.org/10.1016/j.neucom.2021.12.016

[4] Zhibin Li, Shuai Li, and Xin Luo. Using quadratic interpolated beetle antennae search to enhance robot arm calibration accuracy. IEEE Robotics and Automation Letters, 7(4):12046-12053, 2022. https://doi.org/10.1109/LRA.2022.3211776

[5] Steffan Lloyd, Rishad A. Irani, and Mojtaba Ahmadi. Fast and robust inverse kinematics of serial robots using Halley's method. IEEE Transactions on Robotics, 38(5):2768-2780, 2022. https://doi.org/10.1109/TRO.2022.3162954

[6] Xiaolong Wang, Jianfu Cao, Ye Cao, and Feng Zou. Energy-efficient trajectory planning for a class of industrial robots using parallel deep reinforcement learning. Nonlinear Dynamics, 113(8):8491-8511, 2024. https://doi.org/10.1007/s11071-024-10510-4

[7] QingyangWu. Sustainable growth through industrial robot diffusion: Quasi-experimental evidence from a Bartik shift-share design. Economics of Transition and Institutional Change, 31(4):1107-1133, 2023. https://doi.org/10.1111/ecot.12367

[8] Chengmin Zhou, Bingding Huang, and Pasi Fränti. A review of motion planning algorithms for intelligent robots. Journal of Intelligent Manufacturing, 33(2):387-424, 2022. https://doi.org/10.1007/s10845-021-01867-z

[9] Yonggang Li, Rencai Jin, Xiangrong Xu, Yuandi Qian, Haiyan Wang, Shanshan Xu, and Zhixiong Wang. A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach. IEEE Access, 10(1):57736-57747, 2022. https://doi.org/10.1109/ACCESS.2022.3179397

[10] Lisha Dong. Improved A* algorithm for intelligent navigation path planning. Informatica, 48(10):181-194, 2024. https://doi.org/10.31449/inf.v48i10.5693

[11] Valentin N. Hartmann, Andreas Orthey, Danny Driess, Ozgur S. Oguz, and Marc Toussaint. Long-horizon multi-robot rearrangement planning for construction assembly. IEEE Transactions on Robotics, 39(1):239-252, 2022. https://doi.org/10.1109/TRO.2022.3198020

[12] Dawei Sun, Jingkai Chen, Sayan Mitra, and Chuchu Fan. Multi-agent motion planning from signal temporal logic specifications. IEEE Robotics and Automation Letters, 7(2):3451-3458, 2022. https://doi.org/10.48550/arXiv.2201.05247

[13] Ting-Wei Zhang, Guang-Hui Xu, Xi-Sheng Zhan, and Tao Han. A new hybrid algorithm for path planning of mobile robot. The Journal of Supercomputing, 78(3):4158-4181, 2022. https://doi.org/10.1007/s11227-021-04031-9

[14] Jie Zan, Pengtao Ku, and Shoufeng Jin. Research on robot path perception and optimization technology based on whale optimization algorithm. Journal of Computational and Cognitive Engineering, 1(4):201-208, 2022. https://doi.org/10.1109/ACAIT53529.2021.9731150

[15] Yun Liu, Ali Asghar Heidari, Zhennao Cai, Guoxi Liang, Huiling Chen, Zhifang Pan, Abdulmajeed Alsufyani, and Sami Bourouis. Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. Neurocomputing, 503(3):325-362, 2022. https://doi.org/10.1016/j.neucom.2022.06.075

[16] Runqi Chai, Hanlin Niu, Joaquin Carrasco, Farshad Arvin, Hujun Yin, and Barry Lennox. Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment. IEEE Transactions on Neural Networks and Learning Systems, 35(4):5778-5792, 2022. https://doi.org/10.1109/TNNLS.2022.3209154

[17] Pengzhan Chen, Jiean Pei, Weiqing Lu, and Mingzhen Li. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. Neurocomputing, 497(1):64-75, 2022. https://doi.org/10.1016/j.neucom.2022.05.006

[18] Farhad Nawaz, and Melkior Ornik. Multiagent, Multitarget path planning in markov decision processes. IEEE Transactions on Automatic Control, 68(12):7560-7574, 2023. https://doi.org/10.1109/TAC.2023.3286807

[19] Nesrine Khlif, Khraief Nahla, and Belghith Safya. Reinforcement learning with modified exploration strategy for mobile robot path planning. Robotica, 41(9):2688-2702, 2023. https://doi.org/10.1017/S0263574723000607

[20] Daoke Li. Fermat curve path planning method for ship trajectory tracking. Informatica, 48(8):193-206, 2024. https://doi.org/10.31449/inf.v48i8.5735

[21] Javier Muñoz, Blanca López, Fernando Quevedo, Ramón Barber, Santiago Garrido, and Luis Moreno. Geometrically constrained path planning for robotic grasping with differential evolution and fast marching square. Robotica, 41(2):414-432, 2023. https://doi.org/10.1017/S0263574722000224

[22] Ruihua Han, Shengduo Chen, Shuaijun Wang, Zeqing Zhang, Rui Gao, Qi Hao, and Jia Pan. Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. IEEE Robotics and Automation Letters,

7(3):5896-5903, 2022. https://doi.org/10.1109/LRA.2022.3161699

[23] Zhixian Li, Nianfeng Shi, Liguo Zhao, and Mengxia Zhang. Deep reinforcement learning path planning and task allocation for multi-robot collaboration. Alexandria Engineering Journal, 109(1):408-423, 2024. https://doi.org/10.1016/j.aej.2024.08.102