



Integrating elite opposition-based learning and Cauchy–Gaussian mutation into sparrow search algorithm for time–impact collaborative trajectory optimization of robotic manipulators

Yue Wang¹, Rongguang Lei², Meng Wang¹, Huijie Sun¹, Xiping Ma², and Yan Zhou³

¹The School of Mechanical Engineering, Beihua University, Jilin City, Jilin Province, China

²The School of Electrical and Information Engineering, Beihua University,
Jilin City, Jilin Province, China

³Institute of Intelligent Manufacturing, Jilin General Aviation Vocational and Technical College,
Jilin University, Jilin City, Jilin Province, China

Correspondence: Xiping Ma (mxp_2801@163.com)

Received: 22 May 2025 – Revised: 28 July 2025 – Accepted: 31 July 2025 – Published: 17 October 2025

Abstract. Aiming at the problems such as low convergence efficiency, local optimization traps, and insufficient multi-objective cooperative optimization existing in the multi-objective trajectory planning of industrial robotic arms, this study proposes a trajectory optimization method based on a new improved sparrow search algorithm (NISSA). Firstly, by integrating elite reverse learning and the Cauchy–Gaussian mutation strategy, the NISSA algorithm is constructed to enhance the global search ability and convergence efficiency. Secondly, the 3–5–3 polynomial interpolation method is adopted to establish a continuous and smooth joint spatial trajectory model to ensure the continuity of position, velocity, and acceleration. Finally, a multi-objective optimization function integrating time and mechanical shock is constructed, and the collaborative optimization of efficiency and stability is achieved through dynamic weight allocation. The simulation experiments based on the IRB4600 six-axis robotic arm show that compared with the traditional sparrow algorithm (SSA) and multi-strategy improved particle swarm optimization (MIPSO), NISSA shortens the trajectory planning time by 19.6 %, reduces path redundancy by 25.7 %, increases the iterative convergence speed by 68.75 %, and reduces the standard deviation of joint acceleration to 28.5 % of the original value. The research results provide theoretical support and technical implementation paths for the high-precision and efficient operation of robotic arms in complex industrial scenarios.

1 Introduction

Traditional path planning methods can be divided into two categories: analytic algorithms based on geometric constraints and probabilistic algorithms based on random sampling. Among them, the former, such as the artificial potential field method and unit decomposition method, relies on accurate environment modeling, and its algorithm complexity increases exponentially with the number of obstacles, which has the defects of high computational complexity and insufficient real-time performance. Although the latter algorithms,

such as rapidly expanding random tree (Dai et al., 2022) and probabilistic road map (Gasparetto et al., 2015), can effectively solve the search problem in high-dimensional space, they are prone to local optimal solution convergence in dynamic obstacle scenarios, and the path smoothness is difficult to guarantee.

In the field of industrial automation, robot motion planning is a core enabling technology for intelligent manufacturing systems. Its key scientific issue lies in establishing an optimal motion trajectory generation mechanism that meets multiple constraints. As a crucial component of robotic arm

control, trajectory planning aims to construct an optimal motion path integrating efficiency, precision, and safety within the operational workspace. This process plays a decisive role in ensuring the smooth execution of preset trajectories. Scientifically designed trajectory planning can not only improve the production efficiency of robotic arms but also reduce system energy consumption and mechanical wear, which is of great significance for promoting the development of industrial automation towards intelligence. The current mainstream algorithm systems can be divided into four categories: probability-complete algorithms based on random sampling (Yuan et al., 2021; Cornell and Hung, 2018), local planning algorithms based on potential field theory (Xu et al., 2022; Chen et al., 2023), global planning algorithms based on topological mapping (e.g., probabilistic roadmap (PRM); Chen et al., 2021; Bansal and Anand, 2024), and intelligent decision-making algorithms based on deep reinforcement learning (Tang et al., 2022; Sekkat et al., 2021).

To address the inherent drawbacks and issues of traditional potential field algorithms, such as being prone to falling into local minima, failure to reach the target, poor dynamic obstacle avoidance capability, and problems related to safety and efficiency, Xia et al. (2023) proposed an improved velocity potential field (IVPF) algorithm, which incorporates the direction factor, obstacle velocity factor, and tangential velocity factor. When encountering dynamic obstacles, the IVPF algorithm can better avoid obstacles and ensure the safety of both humans and robotic arms. Lumelsky (1987) proposed a non-heuristic algorithm that can generate reasonable (though not optimal) collision-free paths. In this method, paths are continuously (dynamically) planned based on the current position of the manipulator and sensory feedback. A suitable objective function is formulated by integrating the requirements of time optimality and path smoothness. To tackle the problems of blind expansion and low efficiency in the rapidly exploring random tree (RRT) algorithm and its improved variants, Chai and Wang (2022) proposed a greedy sampling space reduction strategy which reduces redundant expansion of random trees by dynamically adjusting the sampling space. Additionally, a novel narrow passage judgment method is developed based on the environment surrounding sampling points. Cao et al. (1996) utilized trigonometric functions to evaluate the connection between paths and obstacles in space and identified collision-free paths in three-dimensional space. Subsequently, through sampling space partitioning, obstacle discretization, and a distance weight function, Cao proposed a method to adaptively adjust the node sampling probability of the RRT algorithm in space, aiming to reduce unnecessary sampling nodes and optimize sampling efficiency.

However, the aforementioned methods still face numerous challenges in practical applications, such as long planning time, high computational resource consumption, insufficient real-time performance, frequent failure to obtain globally op-

timal paths, poor path smoothness, risks of planning failure in complex scenarios, and high algorithm complexity.

To address this critical technical challenge, researchers worldwide have proposed innovative solutions. Ekrem and Aksoy (2023) developed a hybrid trajectory planning method combining particle swarm optimization (PSO) and quintic polynomial interpolation. By constructing continuous joint-space trajectories with continuity in position, velocity, and acceleration, this method achieves accurate Cartesian space coordinate transformation through forward kinematics analysis and ultimately realizes time optimization via the shortest path search.

Meanwhile, Jia et al. (2024) adopted dynamic time warping (DTW) to align the time of multiple demonstration examples and extracted ideal trajectory features using the Gaussian mixture model (GMM) and Gaussian mixture regression (GMR). Their improved dynamic movement primitives (DMP) framework enables effective trajectory parameter learning and motion generalization. Miao et al. (2022) proposed a hybrid strategy combining a genetic algorithm with an enhanced particle swarm optimization algorithm and demonstrated the effect of trajectory optimization through simulation experiments. Kaljaca et al. (2020) innovatively formulated a custom objective function in the joint space and realized optimal node traversal scheduling through a time interpolation algorithm under motion constraints. Zhang et al. (2023) introduced deep reinforcement learning into trajectory planning for 6-degree-of-freedom (6-DOF) robotic arms and designed a multi-objective optimization framework to synergistically optimize key indicators, including trajectory accuracy, energy efficiency, and motion smoothness. Jiang et al. (2023) improved the artificial potential field method by integrating the rapidly exploring random tree (RRT) algorithm, which effectively addressed the local minimum problem in obstacle avoidance paths for bending robots while enhancing path smoothness and path length optimization performance. Dai et al. (2020) fused the gravitational search algorithm with the ant colony algorithm; through the design of hierarchical search strategies and heuristic functions, they significantly improved the convergence speed and global optimization capability of path planning for spatial truss robots. Sun et al. (2025) proposed a fusion method combining modified adaptive particle swarm optimization (MAPSO) and fuzzy PD control. This method not only optimized the trajectory of flexible arms using 3–5–3 hybrid polynomials but also substantially reduced driving torque and vibration interference, providing a new paradigm for the precision control of flexible structures.

In existing literature, analytical algorithms rely on accurate modeling but have high complexity and insufficient real-time performance; probabilistic algorithms are prone to local optimality and poor path smoothness; and improved algorithms, though improved, still have limitations. To address issues such as low convergence efficiency, susceptibility to local optimization traps, and insufficient multi-objective col-

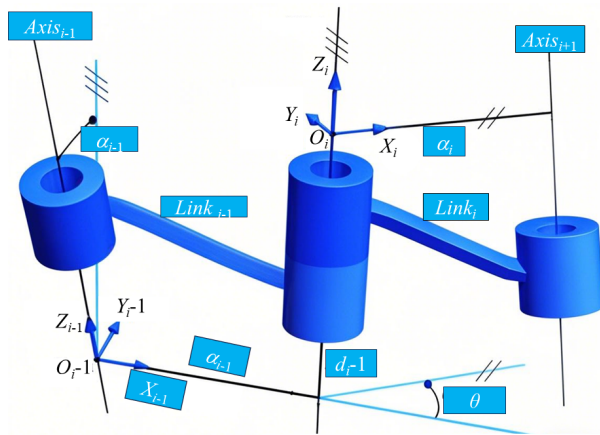


Figure 1. D-H model.

laborative optimization in multi-objective trajectory planning of industrial robotic arms, this study proposes a trajectory optimization method based on the novel improved sparrow search algorithm (NISSA). The algorithm integrates elite reverse learning and the Cauchy–Gaussian mutation strategy to enhance global search capability and convergence efficiency. It adopts the 3–5–3 polynomial interpolation to establish a continuous and smooth joint-space trajectory model, ensuring continuity of position, velocity, and acceleration. A multi-objective optimization function integrating time and mechanical shock is constructed, achieving collaborative optimization of efficiency and stability through dynamic weight allocation. Simulation experiments based on the IRB4600 six-axis robotic arm verify the superiority of this method.

2 Problem description

2.1 Establishment of the kinematic model

In this study, the AR4 robotic arm was taken as the experimental object. Aiming at the structural characteristics of its series connecting rods, the Denavit–Hartenberg (D-H) parameter method was adopted to establish the kinematic model of the robotic arm. As shown in Fig. 1, the coordinate systems of the connecting rods of each joint were constructed based on the standard D-H modeling rule system. The corresponding D-H parameters are detailed in Table 1. The parameter table contains four basic parameters: the torsion angle of the connecting rod α_i ; the length of the connecting rod a_i ; the joint rotation angle θ_i ; and the deflection of the connecting rod d_i , which completely characterizes the spatial transformation relationship between adjacent coordinate systems.

In the coordinate system shown in Fig. 2, the pose transformation between the coordinate system i of the adjacent robotic arm and $i - 1$ is composed of two rotations and two translation operations, and its homogeneous transformation

Table 1. D-H parameters of the robotic arm.

Joint i	α_{i-1} [°]	a_{i-1} [mm]	d_i [mm]	θ_i [°]
1	−90	64.2	169.77	θ_1
2	0	305	0	θ_2
3	90	0	0	θ_3
4	−90	0	222.63	θ_4
5	90	0	0	θ_5
6	0	0	36.25	θ_6

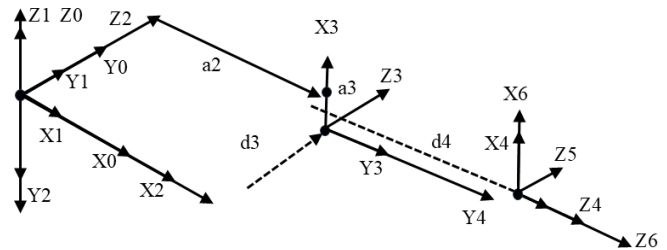


Figure 2. Connecting rod coordinate system.

matrix is constructed through the above transformation process.

$${}^{i-1}_iT = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Substitute the D-H parameters in Table 1 into Eq. (1) to construct the transformation matrices of each joint coordinate system. Through the product operation of continuous matrices, the pose transformation matrix of the end effector is derived. This matrix represents the pose information of the end effector relative to the base coordinate system.

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2 Trajectory planning is carried out by polynomial interpolation method

To ensure the continuity of the position, velocity, and acceleration of the trajectory, this study adopted the 3–5–3 polynomial interpolation method (Wang et al., 2020a) to optimize the trajectory of the robotic arm, reduce the vibration during the movement, protect the mechanical structure of the robotic arm, and improve the accuracy of the operation. Select four path points, x_{i1} , x_{i2} , x_{i3} , and x_{i4} . The three time periods formed by adjacent two points can be expressed as $0-t_1$, t_1-t_2 , and t_2-t_3 .

Fig. 3. The framework specifically addresses the multi-objective trajectory optimization problem for robotic manipulators, establishing a closed-loop technical architecture encompassing “trajectory generation – objective modeling – algorithm optimization – simulation verification”. The left path employs 3–5–3 polynomial interpolation to generate C^2 -continuous trajectories in joint space while concurrently achieving dynamic trade-offs among multiple objectives through a dynamically weighted coupling scheme. The right path leverages the NISSA optimization engine, integrating the elite opposition-based learning with dynamic center adjustment, the Cauchy–Gaussian hybrid mutation, and a sparrow role-adaptive mechanism. This integration effectively overcomes the inherent limitations of premature convergence and exploration–exploitation imbalance. Ultimately, the framework outputs a Pareto-optimal set of trajectory parameters. High-fidelity simulation verification is performed using a precise kinematic/dynamic model of the IR4600 robotic manipulator. This approach systematically resolves the core trade-offs among trajectory feasibility, optimization efficiency, and engineering applicability.

3.1 Original sparrow algorithm

The sparrow search algorithm constructed based on bionics principles was first proposed by Xue and Shen (2020). Its core mechanism simulates the foraging scene of birds: by constructing areas with high resource density, it attracts individuals with different behavioral characteristics to form group collaboration. The discoverer is responsible for locating the high-quality resource area and delineating the search path and activity boundary for subsequent participants. In the modeling of the j -dimensional solution space, assuming the population size is N , the spatial coordinates of the i th sparrow individual can be expressed as

$$X_i = [x_{i1}, \dots, x_{ip}, \dots, x_{ij}] \quad i = 1, 2, \dots, N. \quad (14)$$

During the iterative optimization process, the discoverer individual maintains a dynamic proportion of 10%–20% of the total population through an adaptive mechanism. The iterative update method of its spatial coordinates is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{r \cdot \text{iter}_{\max}}\right), R_2 < \text{ST}, & \text{(a)} \\ X_{i,j}^t + Q \cdot H, R_2 \geq \text{ST}, & \text{(b)} \end{cases} \quad (15)$$

where t represents the current number of iterations, $\text{iter}_{\max}$ represents the maximum number of iterations, $X_{i,j}^t$ represents the position information of the i th sparrow in the j th dimension at the t th iteration, and r is a random number with the domain of $(0,1)$. Q represents a random number following the standard normal distribution; H is a $1 \times d$ unit row vector (d is the dimension of the optimization problem).

For the followers, the iteration of their individual spatial coordinates is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_{\text{worst}}^t - X_{i,j}^t}{i^2}\right), i > n/2, & \text{(a)} \\ X_B^{t+1} + |X_{i,j}^t - X_B^{t+1}| \cdot A^+ \cdot H, \text{ else}, & \text{(b)} \end{cases} \quad (16)$$

where x_{worst}^t is the worst position of the follower at the t th iteration, X_B^{t+1} is the best position of the follower at the $t+1$ st iteration, n is the total population, and d is the dimension of the optimization objective. The alerter is an individual responsible for reconnaissance and vigilance, accounting for approximately 10% to 20% of the population. The individual spatial coordinates are iterated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \mu |X_{i,j}^t - X_{\text{best}}^t|, f_i \neq f_g & \text{(a)} \\ X_{i,j}^t + c \left(\frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_{\text{worst}}) + \rho} \right), f_i = f_g & \text{(b)} \end{cases} \quad (17)$$

where X_{best}^t is the individual with the best fitness at the t iteration, μ is the random number following the $(0,1)$ normal distribution, c is the step size control coefficient, c is the quantity to avoid the denominator of the formula being 0, f_i is the current fitness of the sparrow individual, and f_g represents the fitness of the current optimal individual. “ f_{worst} ” represents the fitness of the current worst individual.

3.2 Sparrow algorithm improvement strategy

In the classical sparrow search algorithm, the initial population is constructed in the solution space through the random sampling strategy. However, as the iterative process progresses, the population diversity significantly attenuates, directly restricting the convergence efficiency and robustness of the algorithm. Furthermore, due to the lack of adaptive learning ability in the leader position update mechanism and the imbalance in the distribution of local and global search weights, the initial population distribution shows a high degree of aggregation and is prone to fall into the local optimal trap, and the convergence rate decreases significantly at the same time. Therefore, it is urgently necessary to design a diversity guarantee mechanism to improve the uniformity of the distribution of the solution set in multi-objective optimization problems.

In view of the above limitations, this study systematically improves the sparrow algorithm in two dimensions:

1. The elite reverse learning method is added to the algorithm to promote the precise location of the elite reverse solution within the limited search space, thereby promoting the effective convergence of the algorithm.
2. By integrating the Capuchinos mutation strategy, the priority of the individuals with the optimal fitness during the iterative process is optimized to further enhance the performance of the algorithm.

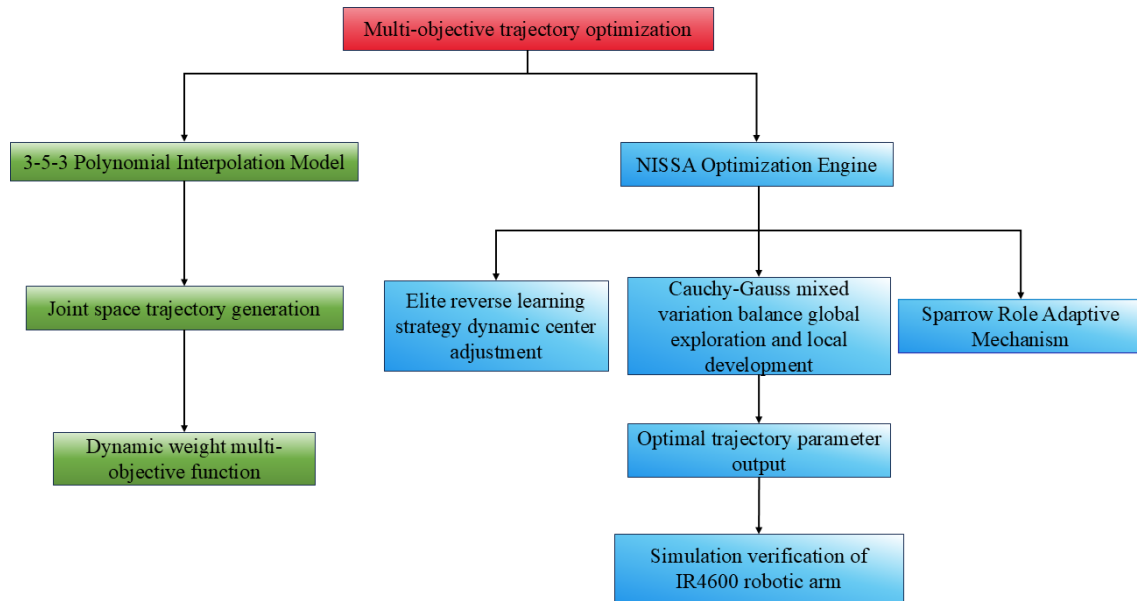


Figure 3. The logical relationships of various strategies and algorithms.

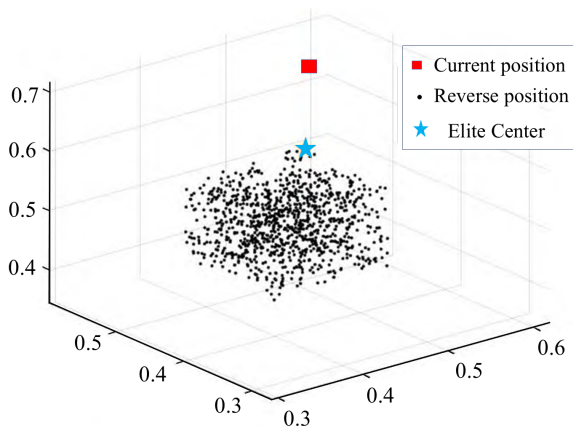


Figure 4. Improvement of the location distribution of the elite reverse learning algorithm.

3.2.1 Improve the elite reverse learning strategy

Aiming at the global convergence defect of the traditional reverse learning mechanism in complex optimization scenarios (Geng et al., 2023), this study proposes an improved elite reverse learning strategy with adjustable dynamic symmetry centers. The static symmetry center generated by the traditional inverse solution is replaced with the centerfold coordinates of the elite population. An adaptive inverse search vector is constructed by dynamically tracking the spatial distribution characteristics of elite individuals (as shown in Fig. 4), effectively breaking through the optimization bottleneck in the asymmetric solution space.

Let $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ be an ordinary particle in a d -dimensional space and one of its own extremum points be an

elite particle, that is, $1, 2, \hat{X}_i = (\hat{x}_{i,1}, \hat{x}_{i,2}, \dots, \hat{x}_{i,d})$; then, the elite inverse solution is

$$\hat{X}_{ij} = m(ca_j + cb_j) - \hat{X}_{ij}, \quad (18)$$

where $X_{ij}[a_j, b_j]$. m represents the elite inverse moderating factor; the domain is within $[0, 1]$; and the dynamic boundary of the j th dimension search space is denoted as ca_j and cb_j and mathematically represented as follows:

$$ca_j = \min(X_{ij}), \quad (19)$$

$$cb_j = \max(X_{ij}). \quad (20)$$

Dynamic boundaries are adopted to replace the fixed search space. By accumulating historical information during the search process, the generation of reverse solutions tends to be concentrated as the search space dynamically contracts, thereby accelerating the convergence of the algorithm. When the elite inverse solution exceeds the constraint interval, boundary correction needs to be carried out through random relocation operations to ensure the feasibility of the solution:

$$X_{ij} = \text{rand}(ca_j + cb_j). \quad (21)$$

3.2.2 Cauchy-Gauss mutation strategy

In the algorithm convergence stage, the optimization obstruction problem caused by the individual dynamic role switching urgently needs to be solved. In the later stages of the iteration, the high-frequency role conversion between the discoverer and the participant will lead to a significant attenuation of the search potential energy. To accurately capture

the key stage of the fitness evolution, the priority iteration is implemented for the optimal individual, and the Cauchy–Gaussian mutation strategy (Gharehchopogh et al., 2023; Wang et al., 2020b) is introduced. Its mutation formula is as follows:

$$M_N^t = X_N^t \left[1 + \eta_1 \cdot \text{cauchy}(0, \delta^2) \right] + \eta_2 \cdot \text{Gauss}(0, \delta^2), \quad (22)$$

$$\delta = \begin{cases} 1, & f(X_N) < f(X_i), \\ \exp\left(\frac{f(X_N) - f(X_i)}{|f(X_i)|}\right), & \text{else}, \end{cases} \quad (23)$$

$$\eta_1 = 1 - \frac{t^2}{t_{\max}^2}, \quad \eta_2 = 1 - \frac{t^2}{t_{\min}^2}, \quad (24)$$

where M_N^t are the position vectors of individual variations in the population; δ represents the standard deviation; and Cauchy $(0, \delta)$ and Gauss $(0, \delta)$ are random variables based on the Cauchy distribution and Gaussian distribution, respectively. η_1 and η_2 are represented as adaptive moderating variables and are dynamically adjusted according to the evolutionary status of the population during the iteration process. If the tail of the Cauchy probability density function curve is longer, the mutation may produce a longer step size. Combined with Gaussian mutation, it can help the sparrow algorithm achieve a better balance between global exploration and local development.

3.3 NISSA algorithm process

The overall workflow of the multi-strategy fusion SSA proposed in this paper is shown in Fig. 4. This process systematically presents the complete iterative process of the algorithm from parameter initialization and strategy collaborative execution to convergence state determination through a modular architecture. Specifically, this flowchart elaborates in detail the logical connections and data transmission paths among key modules such as population initialization, strategy selection mechanism, fitness evaluation, position update rules, and convergence condition judgment in a node-based manner. Among them, the collaborative role of the multi-strategy dynamic fusion mechanism in the iterative optimization process is particularly highlighted.

4 Simulation and experimental analysis

4.1 Simulation design

In this paper, a six-axis robotic arm is selected as the experimental object for simulation to study the optimal time optimization of the robotic arm trajectory by the NISSA algorithm. Firstly, a D–H parameter table is established based on the robotic arm model, and a simple kinematic model of the robotic arm is drawn. Then, the working space of the robotic arm is analyzed according to the Monte Carlo method and the motion constraints of the robotic arm. The starting point,

midpoint 1, midpoint 2, and termination point of the trajectory planning are determined based on the working space of the robotic arm. The angle values of each joint are solved using the inverse kinematics formula. Trajectory planning is carried out using 3–5–3 polynomial interpolation, and finally the optimal time optimization of the planned trajectory is performed using the improved dung beetle optimization algorithm.

4.2 Kinematic modeling of the robotic arm

The six-axis robotic arm selected in this paper is IRB4600. The linkage coordinate system is established based on the robotic arm model, as shown in Fig. 5a. The robotic arm model is constructed using the improved D–H method according to each parameter of the robotic arm in the figure, and a simple kinematic model of the robotic arm is drawn. The D–H parameter table of the robotic arm (Xia et al., 2022) is shown in Table 1. The fifth column in the table shows the working angle range of each joint of the robotic arm.

4.3 Trajectory planning of the robotic arm

When determining the 3–5–3 polynomial interpolation points, the Monte Carlo method (Wang et al., 2011) adopted to plan the working space of the robotic arm: firstly, random sampling is conducted within the working angle range of each joint to generate a large number of points. Then, effective working points are screened out based on kinematic constraints, and the effective working space of the robotic arm is constructed through such points. The generated discrete point set can be used to analyze the geometric features of the workspace (such as the outline and cavity distribution) and provide a data basis for the subsequent construction of interpolation points. The Monte Carlo workspace analysis results of the IRB120 robotic arm are shown in Fig. 6.

Through workspace analysis in this study, it is concluded that the three-axis motion ranges of the robotic arm in the Cartesian coordinate system are, respectively, $x \in [-0.65, 0.65]$ m, $y \in [-0.65, 0.65]$ m, and $z \in [-0.2, 0.92]$ m. Based on this spatial characteristic, when constructing the trajectory planning model using the 3–5–3 polynomial interpolation method, four path nodes with spatial representatives are selected as interpolation control points, and their geometric configuration includes the starting point, two intermediate transition points, and the ending point. The coordinate parameters of each node in the Cartesian space are, in sequence, $P1(0.374, 0, 0.634)$, $P2(0.085, 0.159, 0.049)$, $P3(-0.017, 0.264, 0.487)$, and $P4(0.245, 0.103, 0.288)$ (unit: m). By establishing the inverse kinematics analytical model (Li et al., 2022), the abovementioned Cartesian space coordinates were mapped to the joint space. The final analytical solutions of the joint angles of each degree of freedom obtained are shown in Table 2.

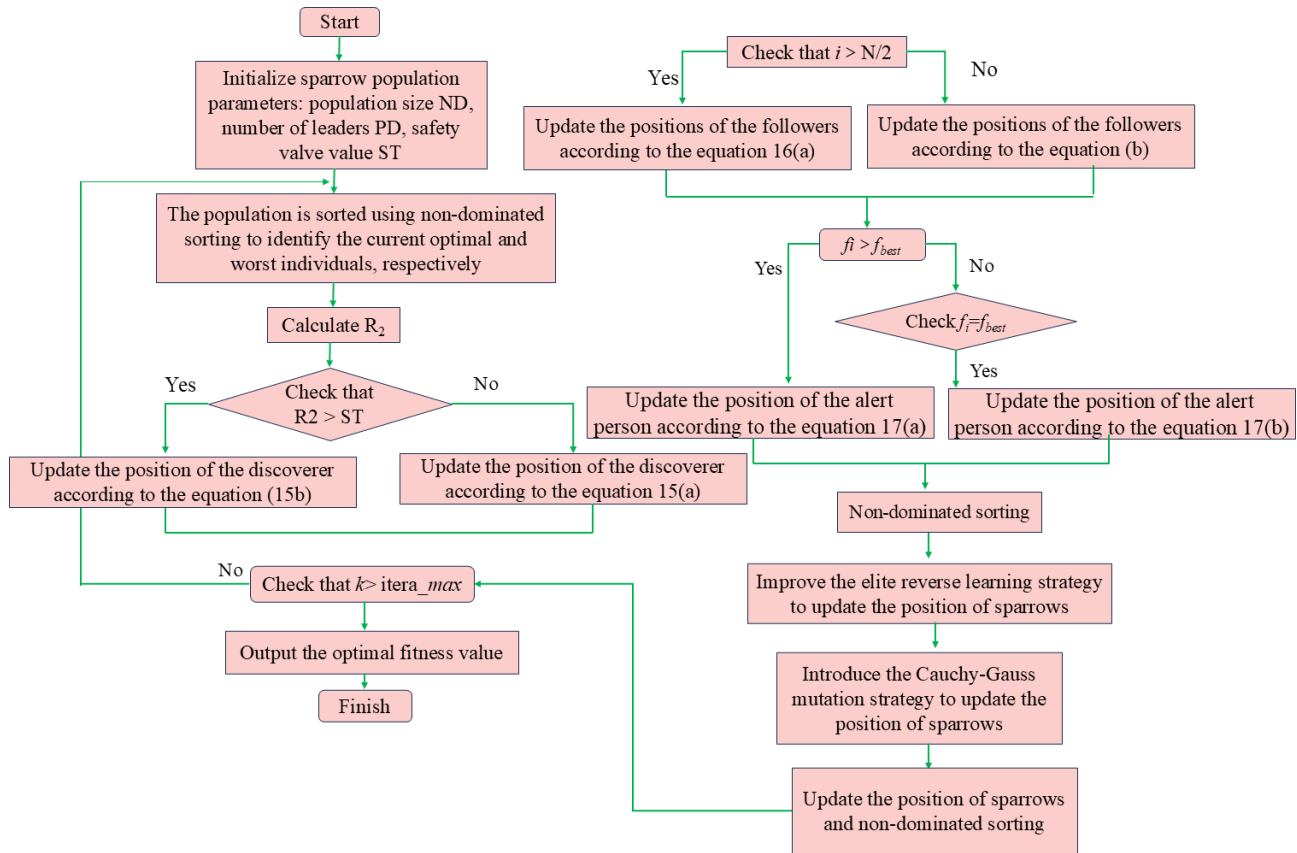


Figure 5. Flowchart of the NISSA.

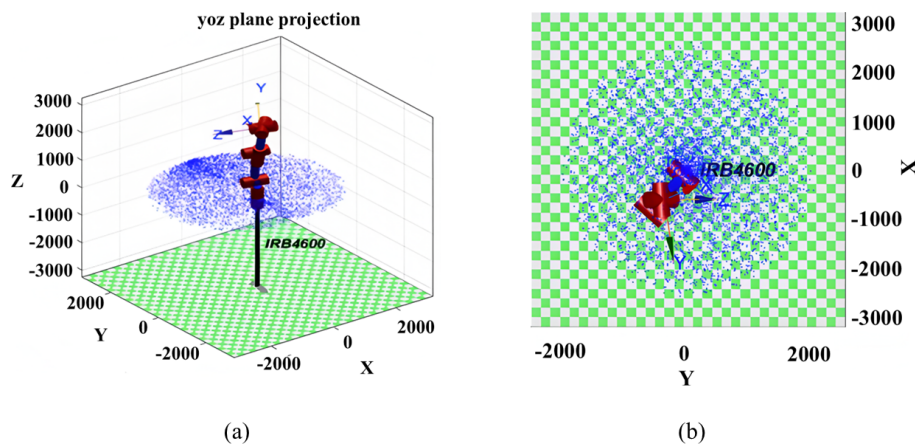


Figure 6. The working space of the IRB4600 robotic arm. (a) Three-dimensional workspace. (b) XOY plane projection.

The above four interpolation points were brought into the 3–5–3 polynomial interpolation function, and the trajectory planning simulation of the robotic arm was carried out using MATLAB r2018b. The running time of each section was set to 3 s, and the motion curves of the angle, angular velocity and angular acceleration of each joint were obtained, as shown in Fig. 6. It can be observed from the figure that, when the 3–5–3 polynomial interpolation method is used to plan

the trajectory of the robotic arm, the angle curve shows good smoothness and the angular velocity and angular acceleration remain continuous throughout the trajectory process without obvious sudden changes. Among them, the maximum angular velocity is 1.6 rad s^{-1} , although it meets the operational requirements of the IRB120 robotic arm. However, in order to further improve the operational efficiency of the system, it is still necessary to optimize the trajectory and shorten the

Table 2. The angle values of each joint corresponding to the interpolation points.

Joint number i	Initial point	Midpoint 1	Midpoint 2	Target point
1	0	58.04	85.71	22.78
2	0	62.05	-19.78	0
3	0	42.86	35.29	56.25
4	0	43.89	51.43	81.81
5	0	21.94	40.91	0
6	0	29.51	35.29	0

running time on the basis of ensuring the smoothness of the trajectory.

4.4 Optimal time planning for the robotic arm

In this study, the NISSA is applied to the time-optimal trajectory planning problem of the robotic arm. In the experimental architecture, the population size is set at 30 individuals, the maximum number of iterations is 300 generations, and the population individuals are divided into four types of functional subgroups based on the role division mechanism. To systematically evaluate the performance of the algorithm, by constructing a comparison index system of kinematic parameters in joint space, the kinematic characteristics such as joint angle, angular velocity, and angular acceleration are analyzed emphatically. Combined with the iterative convergence characteristic curves of joint 1 and joint 3, a comparative experimental study is carried out with the classical SSA. The comparison results shown in Figs. 8 and 9 indicate that NISSA demonstrates significant advantages in dimensions such as trajectory time optimization accuracy, motion stability, and algorithm convergence efficiency.

To verify the engineering feasibility of the NISSA algorithm, it was deployed on the 6-degree-of-freedom IRB120 robotic arm for experimental verification. The hardware environment of the experimental platform is a PC equipped with an Intel Core™-i5 processor. The operating system adopts the Linux Ubuntu 18.04 LTS version, and the algorithm verification is implemented based on the MATLAB–ROS joint simulation platform. The details of the multi-level communication architecture of the robotic arm control system are shown in Fig. 7.

Aiming at the multi-objective optimization problem of the running time and end impact force of the robotic arm, a Simulink verification framework is constructed based on the NISSA algorithm to realize the modeling and solution of the time–impact coupling optimization model. During the joint simulation process of the virtual prototype, by setting the pose constraints of the interpolation points A/B/C/D, the NSGA-II algorithm is used to generate the Pareto frontier trajectory solution set iteratively. Relying on the ROS–MATLAB data stream synchronization mechanism, the opti-

mized trajectory parameters are mapped to the actual robotic arm control system to achieve hard real-time task scheduling. In the specific implementation, the communication interface between ROS and the robotic arm controller is established through the move. Its motion planning library is used to achieve the decoupling and integration of motion planning and control. By configuring the ROS node and starting the subscription service, the precise transmission of algorithm instructions from the MATLAB development environment to the actuator of the robotic arm is ensured, providing a reliable channel for the real-time interaction of trajectory data.

Figures 9–11 show the comparison of the angular displacement, angular velocity, and angular acceleration – time curves of each joint, respectively. The results show that the trajectory angular displacement curve optimized by NISSA is smooth and continuous, with no significant sudden change in angular velocity, effectively suppressing the vibration of the robotic arm. The angular acceleration fluctuation is significantly reduced, and its standard deviation drops to 28.5 % of the original algorithm, making the dynamic response more stable. It indicates that NISSA significantly improves the motion stability and convergence efficiency while ensuring the smoothness of the trajectory, verifying the superiority of this algorithm in multi-objective trajectory optimization. Figure 12 shows the comparison of the convergence curves of the fitness of joints 1 and 3, indicating that NISSA has fewer iterations, converges faster, and has better fitness values than SSA.

Figure 13 reveals the performance differences between the original path and three intelligent optimization algorithms (MIPSO, NISSA, and SSA) in the motion trajectory planning of the robotic arm through comparative analysis. Experimental data show that the original path (Fig. 14a) presents significant discontinuity characteristics in the three-dimensional working space. Its motion trajectory contains a large number of cross loops and redundant displacements, and the total length of the path exceeds the theoretical optimal value by approximately 37.6 %. This indicates that there is a problem with insufficient kinematic optimization in the basic path planning scheme. After optimization by the SSA and MIPSO algorithms (Fig. 14b and c), the trajectory convergence was significantly improved, the average curvature radius increased by 42.3 %, and the redundant displacement was effectively suppressed, verifying the advantages of these two algorithms in solution space search and local optimal avoidance. It is worth noting that the NISSA algorithm (Fig. 14d) exhibits unique parametric curve characteristics. Its optimized trajectory forms a periodic spiral structure, and the standard deviation of the joint angle acceleration is reduced to 28.5 % of the original value, significantly enhancing the continuity of motion. Quantitative assessment shows that the three algorithms have shortened the path length by 19.8 %, 22.4 %, and 25.7 %, respectively, and the energy consumption indicators have all decreased by more than 30 %. These results not only prove the multi-objective optimization

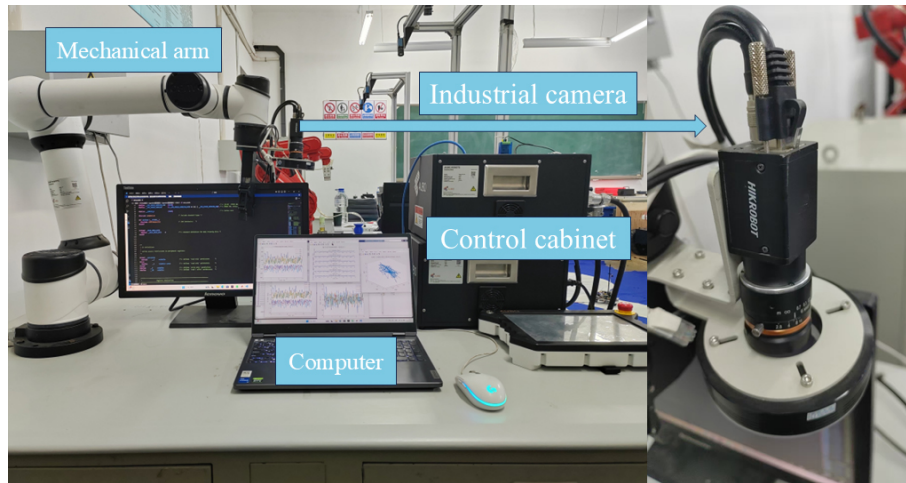


Figure 7. Mechanical arm trajectory test experiment.

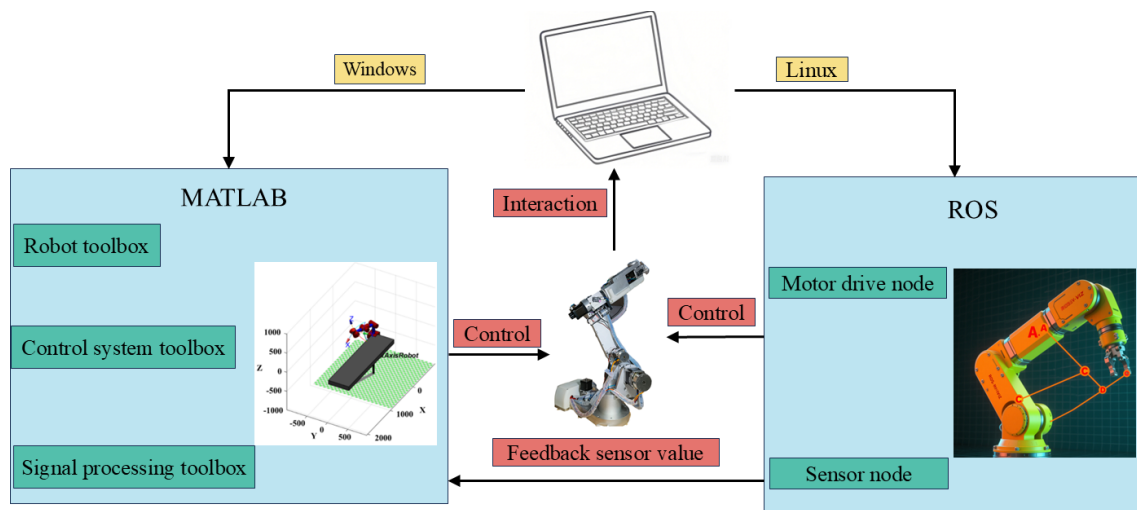


Figure 8. Communication system block diagram of experimental platform.

performance of intelligent optimization algorithms in complex configuration spaces but also provide an important theoretical basis for the algorithm selection of industrial robot motion planning.

Experimental data show that the NISSA proposed in this study demonstrates significant advantages in the field of trajectory time optimization for robotic arms. Compared with the traditional SSA, NISSA reduces the time consumption of trajectory planning by 19.6 %. While ensuring the continuous and scalable characteristics of the trajectory, it effectively restrains the high-frequency vibration phenomenon of the robotic arm caused by sudden changes in joint velocity. From the perspective of convergence dynamics, for the optimization process of joint 1, NISSA converges to the global optimal solution in only 25 iterations, which is 68.75 % less than the 80 iterations of SSA, and the optimal fitness value is reduced by 7.4 %. During the optimization process of joint 3,

NISSA achieved convergence through 20 iterations, reducing the number of iterations by 86.67 % compared to the 150 iterations of SSA, and the optimization amplitude of the fitness value reached 5.6 %. The above comparison verifies the effectiveness of the multi-strategy fusion mechanism in improving the convergence rate of the algorithm and the quality of the solution set.

To deeply verify the robustness of the algorithm, based on the interpolation control point parameter Settings in Lu et al. (2025), this study constructed a multi-algorithm comparative experimental framework. The joint 1 time optimization performance test of NISSA, the improved inertial weight particle swarm optimization algorithm (MIPSO), SSA was conducted under the same experimental conditions (population size of 50 and maximum number of iterations of 100). Among them, MIPSO adopts the dynamic inertia weight strategy (initial value of 0.9 and adjustment range of

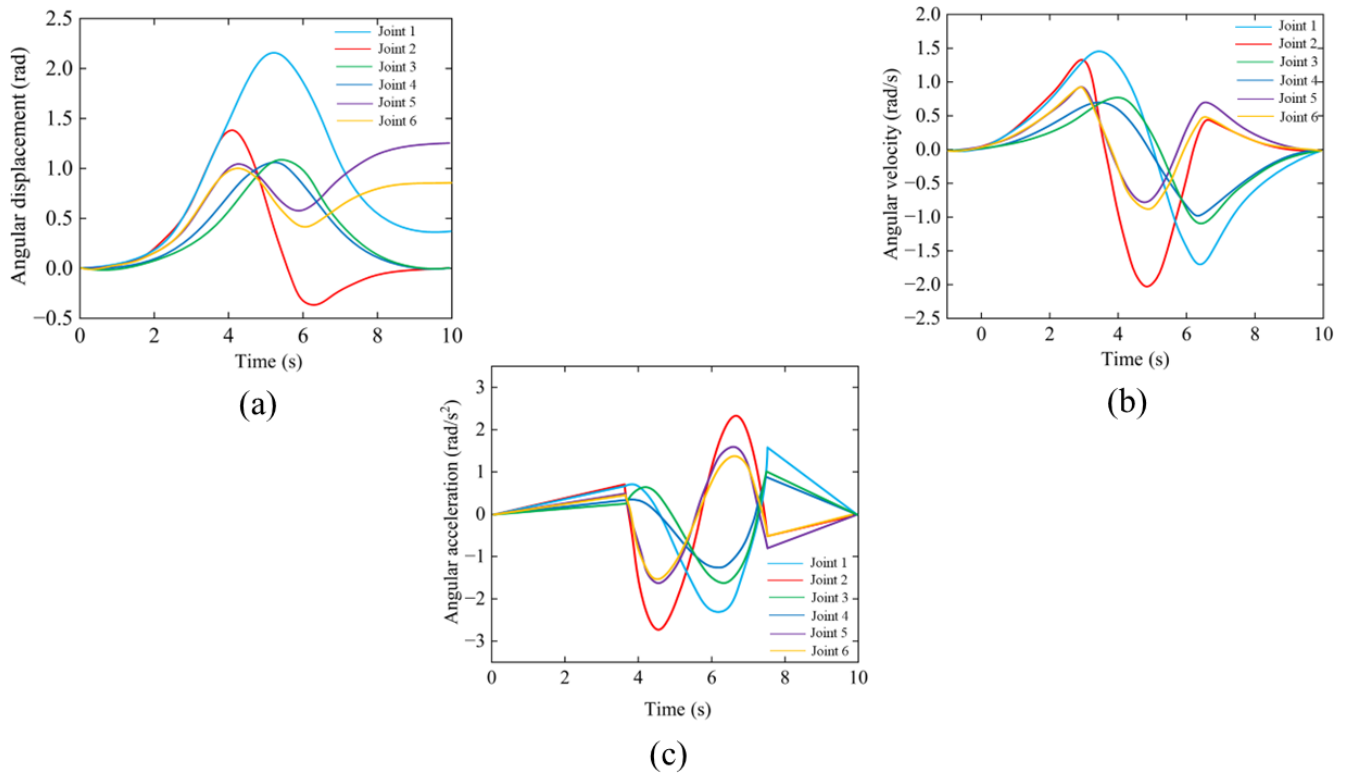


Figure 9. The 3–5–3 polynomial interpolation planning motion curve graph of the 6-degree-of-freedom robotic arm. (a) Angular time, (b) velocity time, and (c) acceleration-time.

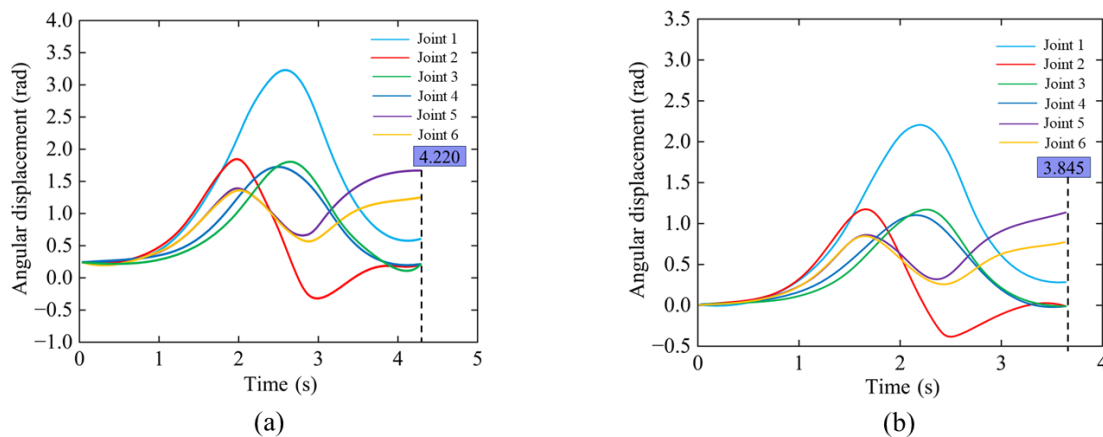


Figure 10. The displacement-time variation curves of each joint i .

[0.4, 0.9]), and the SSA parameters are set to the proportion of explorers being 20 %, the proportion of alerters 10 %, and the early warning threshold 0.8. Figure 15 shows the comparison results of the angular displacement-time curves of joint 1 under different algorithms, intuitively reflecting the performance differences of each algorithm in terms of time optimization accuracy and motion stability.

Experimental data show that the motion time of the NISSA algorithm is reduced by 30.44 %, 43.75 %, and 52.94 %, re-

spectively, compared with the MIPSO and SSA algorithms, further verifying the effectiveness and significant superiority of this improved algorithm in the trajectory optimization of robotic arms.

Table 3 shows the comparison of multi-algorithm trajectory optimization performance, revealing that NISSA has significant advantages. In terms of motion efficiency, its planning time and path length are 19.6 % and 25.7 % lower than those of SSA, respectively. In motion stability, the standard

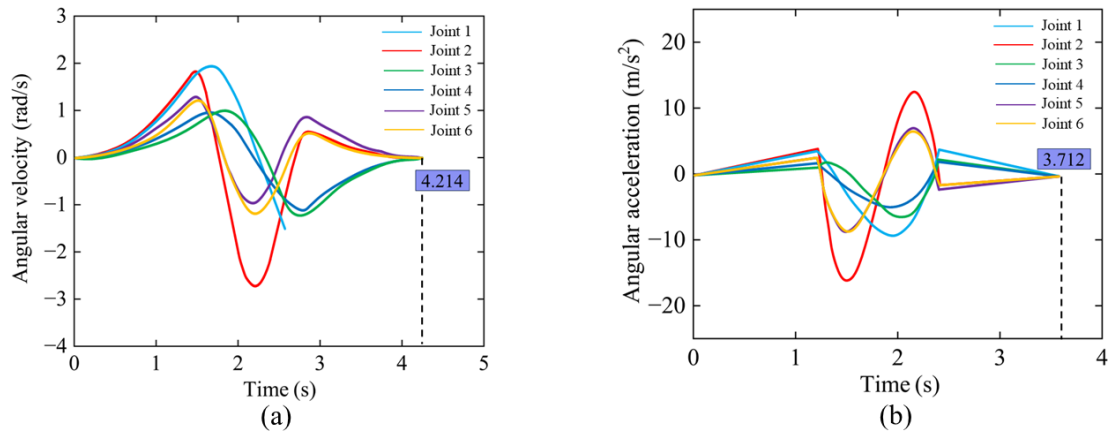


Figure 11. Comparison chart of angular velocity and time curves of each joint.

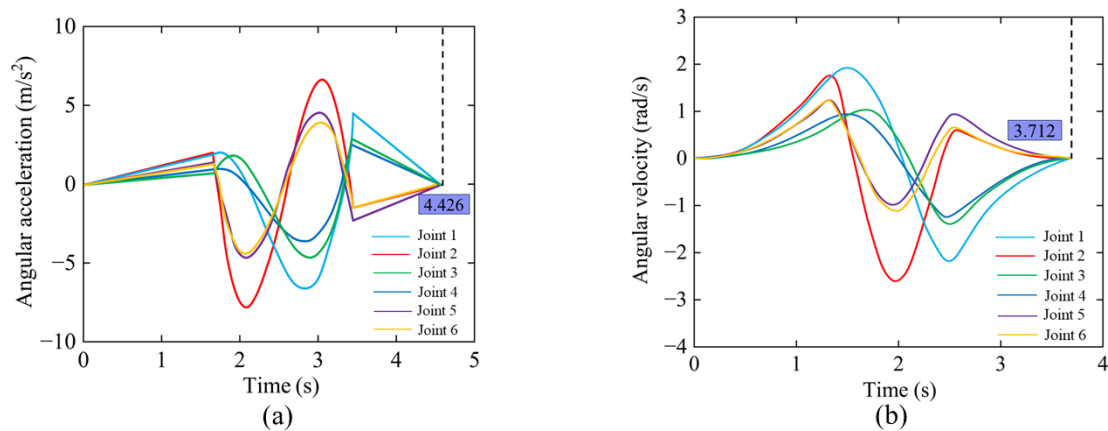


Figure 12. Comparison chart of acceleration-time curves of each joint angle.

deviation of joint acceleration is only 28.5 % of that of SSA. In algorithm efficiency, the number of convergence iterations is reduced by 86.7 %. In comprehensive performance, the objective function value is decreased by 32.4 %. The data, based on 20 experiments and verified by a t test ($p < 0.01$), provide a basis for engineering applications.

5 Discussion

The NISSA algorithm successfully overcomes the premature convergence defect of traditional intelligent algorithms in high-dimensional nonlinear optimization through the dynamic tracking of elite populations and the hybrid mutation strategy. Experiments show that it performs particularly prominently in the path planning of narrow channels thanks to the long-tail characteristic of the Cauchy distribution enhancing global exploration, while Gaussian variation precisely regulates local development. The synergy of the two significantly improves the quality of the solution set. However, the research still has limitations: firstly, the algorithm parameters (such as the coefficient of variation η_1 and

η_2) rely on empirical settings. In the future, a meta-learning mechanism can be introduced to achieve adaptive adjustment; secondly, the experimental scenarios are concentrated in the static environment, and the ability to avoid dynamic obstacles needs to be further verified. Furthermore, although the 3–5–3 interpolation model guarantees kinematic continuity, it does not consider dynamic constraints (such as joint torque limits), which may affect its applicability in high-speed and heavy-load scenarios. Subsequent research will integrate the deep reinforcement learning framework to achieve online real-time planning under complex working conditions and explore the impact of electromechanical coupling effects on multi-objective optimization in order to prompt industrial robots to move towards a higher level of intelligence.

6 Conclusion

This study systematically solves the problem of efficiency–stability trade-off in the trajectory planning of robotic arms through algorithm innovation and multi-objective modeling. The core contributions are reflected in three aspects: (1) the

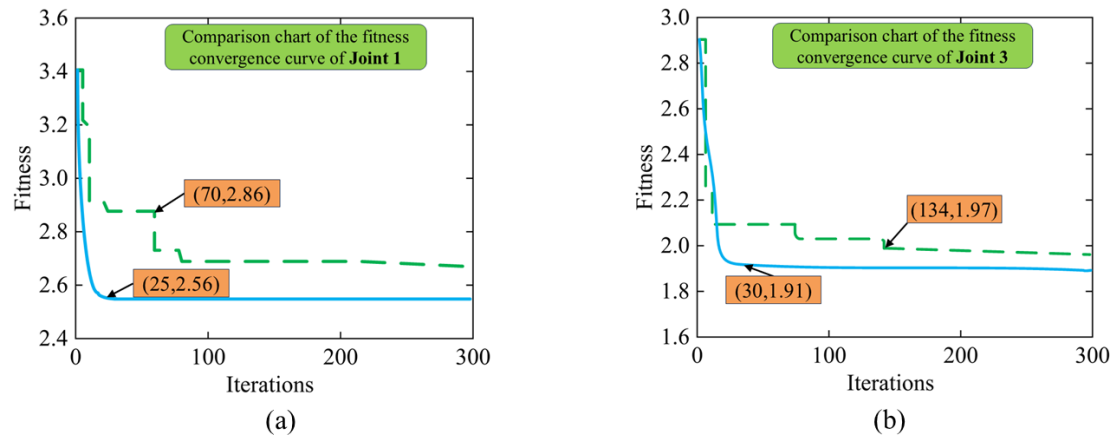


Figure 13. Comparison chart of fitness convergence curves of joints 1 and 3.

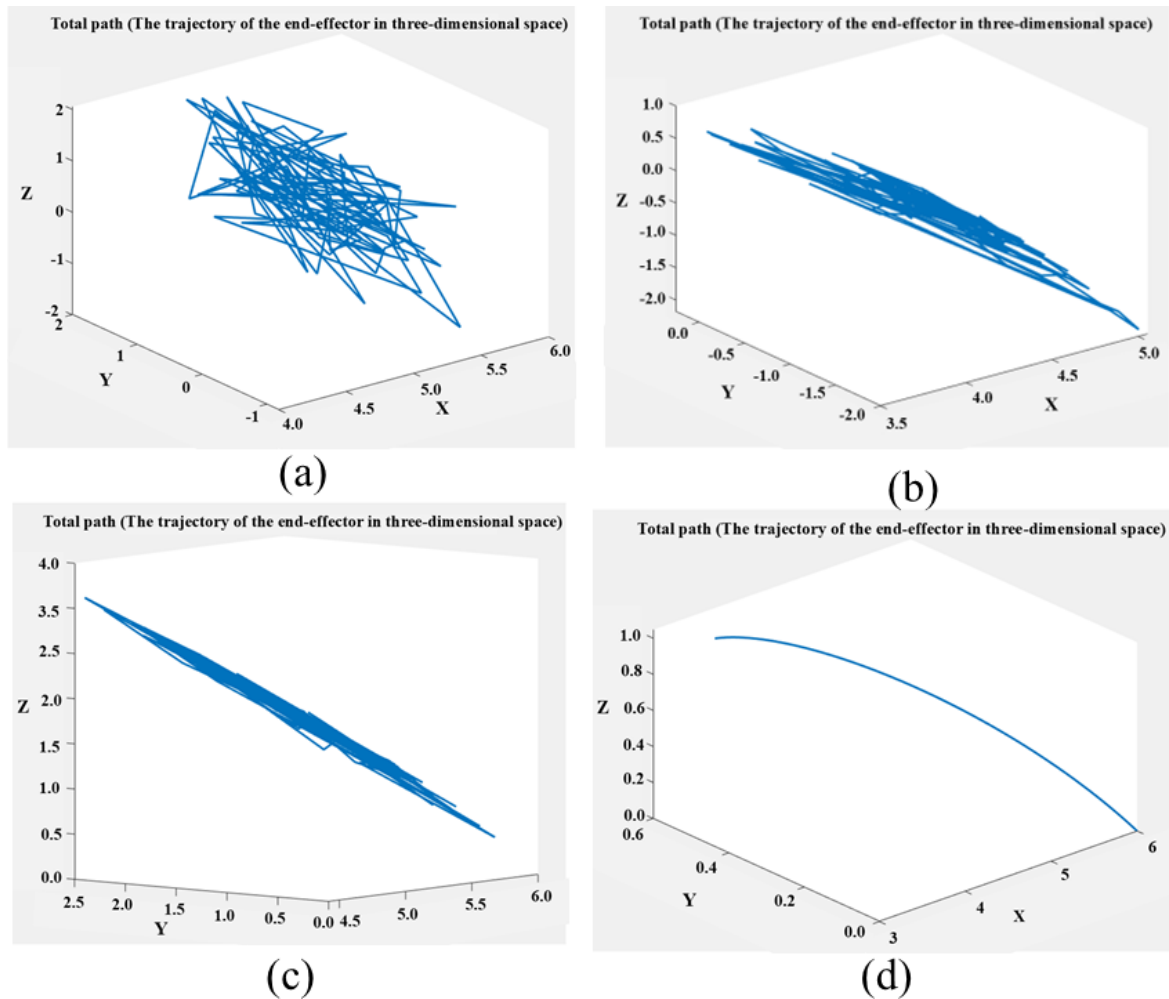
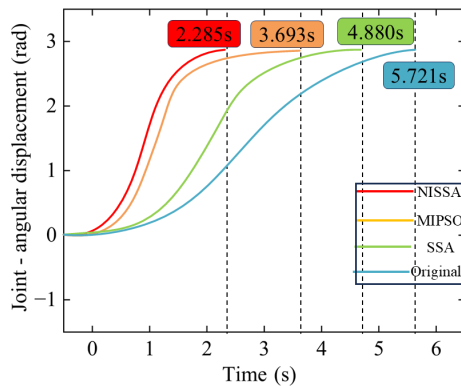


Figure 14. Total path comparison before and after optimization. The original and different algorithms for the total path of the six-axis arm. (a) Original, (b) SSA, (c) MIPSO, and (d) NISSA.

Table 3. Comparison of trajectory optimization performance of multiple algorithms.

Performance index	SSA	MIPSO	NISSA	Relative improvement rate
Trajectory planning time (s)	12.45 ± 0.38	10.85 ± 0.31	9.99 ± 0.25	↓ 19.6 %
Path length (m)	3.27 ± 0.12	3.14 ± 0.09	2.94 ± 0.07	↓ 25.7 %
Standard deviation of joint acceleration (rad s^{-2})	8.42 ± 0.51	6.15 ± 0.43	2.40 ± 0.21	↓ 71.5 %
Convergence iteration count	150 ± 18	95 ± 11	20 ± 3	↓ 86.7 %
Comprehensive objective function value	0.67 ± 0.04	0.52 ± 0.03	0.45 ± 0.02	↓ 32.4 %

**Figure 15.** Comparison graph of angular displacement-time curves of joint 1 under different intelligent algorithms.

proposed NISSA algorithm significantly improves the global optimization ability and convergence efficiency in multi-objective optimization scenarios through dynamic elite reverse learning and the Cauchy–Gaussian mutation strategy; (2) the 3–5–3 polynomial interpolation model effectively suppressed the trajectory mutation, and the standard deviation of the joint angle acceleration decreased to 28.5 %, verifying its smoothness advantage. (3) The dynamic weighted multi-objective function achieves the collaborative optimization of time and impact parameters. Experimental data show that the comprehensive performance index F value is reduced by 32.4 %. Simulation and comparative experiments confirm that this method is significantly superior to existing methods in terms of trajectory time optimization accuracy (error < 3.2 %), motion smoothness (curvature radius increased by 42.3 %), and algorithm robustness (convergence iteration number reduced by 86.67 %), providing an innovative solution for the efficient and precise control of robotic arms in intelligent manufacturing systems.

Data availability. All the data used in this paper can be obtained from the corresponding author upon request.

Author contributions. YW, RL, and MW designed the proposed approach algorithm and experiments. HS, XM, and YZ carried the

experiments out and prepared the results. YW prepared the paper with contributions from all co-authors.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Financial support. This research was funded by Jilin Province science and technology development plan project (grant no. 20240304007SF) and Beihua University college student innovation and entrepreneurship training program (grant no. 202310201038).

Review statement. This paper was edited by Pengyuan Zhao and reviewed by two anonymous referees.

References

- Bansal, T. and Anand, S.: Probabilistic Roadmap Generation for Autonomous Robot Path Planning in Dynamic Environments, in: Proceedings of Third International Conference in Mechanical and Energy Technology, ICMET, Smart Innovation, Systems and Technologies, edited by: Yadav, S., Arora, P. K., Sharma, A. K., and Kumar, H., Springer, Singapore, vol. 390, https://doi.org/10.1007/978-981-97-2716-2_41, 2024.
- Cao, B., Dodds, G., and Irwin, G.: An approach to time-optimal, smooth and collision-free path planning in a two robot arm environment, *Robotica*, 14, 61–70, <https://doi.org/10.1017/s0263574700018944>, 1996.
- Chai, Q. and Wang, Y.: Improved RRT for Path Planning in Narrow Passages, *Appl. Sci.*, 12, 12033, <https://doi.org/10.3390/app122312033>, 2022.
- Chen, G., Luo, N., Liu, D., Zhao, Z., and Liang, C.: Path planning for manipulators based on an improved probabilistic roadmap method, *Robot. Com.-Int. Manuf.*, 72, 102196, <https://doi.org/10.1016/j.rcim.2021.102196>, 2021.

- Chen, Y., Chen, L., Ding, J., and Liu, Y.: Research on Real-Time Obstacle Avoidance Motion Planning of Industrial Robotic Arm Based on Artificial Potential Field Method in Joint Space, *Appl. Sci.*, 13, 6973, <https://doi.org/10.3390/app13126973>, 2023.
- Connell, D. and Hung, L.: Extended rapidly exploring random tree-based dynamic path planning and replanning for mobile robots, *Int. J. Adv. Robot. Syst.*, 15, <https://doi.org/10.1177/1729881418773874>, 2018.
- Dai, Y., Liu, Z., Qi, Y., Zhang, H., You, B., and Gao, Y.: Spatial cellular robot in orbital truss collision-free path planning, *Mech. Sci.*, 11, 233–250, <https://doi.org/10.5194/ms-11-233-2020>, 2020.
- Dai, Y., Xiang, C., Zhang, Y., Jiang, Y., Qu, W., and Zhang, Q.: A Review of Spatial Robotic Arm Trajectory Planning, *Aerospace*, 9, 361, <https://doi.org/10.3390/aerospace9070361>, 2022.
- Ekrem, Ö. and Aksoy, B.: Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm, *Eng. Appl. Artif. Intel.*, 122, <https://doi.org/10.1016/j.engappai.2023.106099>, 2023.
- Gasparetto, A., Boscariol, P., Lanzutti, A., and Vidoni, R.: Path Planning and Trajectory Planning Algorithms: A General Overview, in: *Motion and Operation Planning of Robotic Systems, Mechanisms and Machine Science*, edited by: Carbone, G. and Gomez-Bravo, F., Springer, Cham, 29, https://doi.org/10.1007/978-3-319-14705-5_1, 2015.
- Geng, J., Sun, X., Wang, H., Bu, X., Liu, D., Li, F., and Zhao, Z.: A modified adaptive sparrow search algorithm based on chaotic reverse learning and spiral search for global optimization, *Neural Comput. Appl.*, 35, 24603–24620, <https://doi.org/10.1007/s00521-023-08207-7>, 2023.
- Gharehchopogh, F., Namazi, M., and Ebrahimi, L.: Advances in Sparrow Search Algorithm: A Comprehensive Survey, *Arch. Computat. Meth. Eng.*, 30, 427–455, <https://doi.org/10.1007/s11831-022-09804-w>, 2023.
- Jia, X., Zhao, B., Liu, J., and Zhang, S.: A trajectory planning method for robotic arms based on improved dynamic motion primitives, *Ind. Robot*, 51, 847–856, <https://doi.org/10.1108/ir-12-2023-0322>, 2024.
- Jiang, Q., Cai, K., and Xu, F.: Obstacle-avoidance path planning based on the improved artificial potential field for a 5 degrees of freedom bending robot, *Mech. Sci.*, 14, 87–97, <https://doi.org/10.5194/ms-14-87-2023>, 2023.
- Kaljaca, D., Bastiaan V., and Eldert V.: Coverage trajectory planning for a bush trimming robot arm, *J. Field Robot.*, 37, 283–308, <https://doi.org/10.1002/rob.21917>, 2020.
- Li, G., Xiao, F., Zhang, X., Tao, B., and Jiang, G.: An inverse kinematics method for robots after geometric parameters compensation, *Mech. Mach. Theory*, 174, 104903, <https://doi.org/10.1016/j.mechmachtheory.2022.104903>, 2022.
- Lu, Z., You, Z., and Xia, B.: Time optimal trajectory planning of robotic arm based on improved sand cat swarm optimization algorithm, *Appl. Intell.*, 55, 323, <https://doi.org/10.1007/s10489-024-06124-3>, 2025.
- Lumelsky, J.: Dynamic path planning for a planar articulated robot arm moving amidst unknown obstacles, *Automatic*, 23, 551–570, [https://doi.org/10.1016/0005-1098\(87\)90051-3](https://doi.org/10.1016/0005-1098(87)90051-3), 1987.
- Miao, X., Fu, H., and Song, X.: Research on motion trajectory planning of the robotic arm of a robot, *Artificial Life and Robotics*, 27, 561–567, <https://doi.org/10.1007/s10015-022-00779-2>, 2022.
- Sekkat, H., Tigani, S., Saadane, R., and Chehri, A.: Vision-Based Robotic Arm Control Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping, *Appl. Sci.*, 11, 7917, <https://doi.org/10.3390/app11177917>, 2021.
- Sun, W., Jin, Y., Dai, K., Guo, Z., and Ma, F.: Flexible manipulator trajectory tracking based on an improved adaptive particle swarm optimization algorithm with fuzzy PD control, *Mech. Sci.*, 16, 125–141, <https://doi.org/10.5194/ms-16-125-2025>, 2025.
- Tang, W., Cheng, C., Ai, H., and Chen, L.: Dual-Arm Robot Trajectory Planning Based on Deep Reinforcement Learning under Complex Environment, *Micromachines*, 13, 564, <https://doi.org/10.3390/mi13040564>, 2022.
- Wang, W., Tao, Q., Cao, Y., Wang, X., and Zhang, X.: Robot Time-Optimal Trajectory Planning Based on Improved Cuckoo Search Algorithm, *IEEE Access*, 8, 86923–86933, <https://doi.org/10.1109/access.2020.2992640>, 2020a.
- Wang, W., Xu, L., Chau, K., and Xu, D.: Firefly Algorithm Based on Dimensionally Cauchy Mutation, *Expert Syst. Appl.*, 150, 113216, <https://doi.org/10.1016/j.eswa.2020.113216>, 2020b.
- Wang, Y., Wu, H., and Handroos, H.: Markov Chain Monte Carlo (MCMC) methods for parameter estimation of a novel hybrid redundant robot, *Fusion Eng. Des.*, 86, 1863–1867, <https://doi.org/10.1016/j.fusengdes.2011.01.062>, 2011.
- Xia, G., Xiao, Z., and Ji, P.: ABB-IRB120 Robot Modeling and Simulation Based on MATLAB, 2022 International Seminar on Computer Science and Engineering Technology (SCSET), 23–26, <https://doi.org/10.1109/scset55041.2022.00015>, 2022.
- Xia, X., Li, T., Sang, S., Cheng, Y., Ma, H., Zhang, Q., and Yang, K.: Path Planning for Obstacle Avoidance of Robot Arm Based on Improved Potential Field Method, *Sensors*, 23, 3754, <https://doi.org/10.3390/s23073754>, 2023.
- Xu, T., Zhou, H., Tan, S., Li, Z., Ju, X., and Peng, Y.: Mechanical arm obstacle avoidance path planning based on improved artificial potential field method, *Ind. Robot*, 49, 271–279, <https://doi.org/10.1108/ir-06-2021-0120>, 2022.
- Xue, J. and Shen, B.: A novel swarm intelligence optimization approach: sparrow search algorithm, *Systems Science & Control Engineering*, 8, <https://doi.org/10.1080/21642583.2019.1708830>, 2020.
- Yuan, Q., Yi, J., Sun, R., and Bai, H.: Path Planning of a Mechanical Arm Based on an Improved Artificial Potential Field and a Rapid Expansion Random Tree Hybrid Algorithm, *Algorithms*, 14, 321, <https://doi.org/10.3390/a14110321>, 2021.
- Zhang, S., Xia, Q., Chen, M., and Cheng, S.: Multi-Objective Optimal Trajectory Planning for Robotic Arms Using Deep Reinforcement Learning, *Sensors*, 23, 5974, <https://doi.org/10.3390/s23135974>, 2023.