applied sciences

MDPI

Article

Multi-Objective Optimal Trajectory Planning for Woodworking Manipulator and Worktable Based on the INSGA-II Algorithm

Jiaping Yi [1], Changqing Zhang [1,*], Sihan Chen [2], Qinglong Dai [1], Hang Yu [1], Guang Yang [1] and Leyuan Yu [1]

[1] Research Institute of Wood Industry, Chinese Academy of Forestry, Beijing 100091, China; yijiaping0806@163.com (J.Y.); 16638498817@163.com (Q.D.); yuhang05@163.com (H.Y.); heyyangguang@sina.com (G.Y.); 13007119939@163.com (L.Y.)

[2] Research Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, Beijing 100091, China; han_99330@163.com

* Correspondence: zhangchangqing69@sina.com; Tel.: +86-10-62889346

Abstract: The manipulator has been widely used in the wood processing industry; the main problem currently faced is optimizing the motion trajectory to enhance the processing efficiency and operational stability of the woodworking manipulator and worktable. A 5-7-5 piecewise polynomial interpolation method is proposed to construct the spatial trajectories of each joint. An improved non-dominated sorting genetic algorithm (INSGA-II) is proposed to achieve a time–jerk multi-objective trajectory planning that can meet the dual requirements of minimal processing time and reduced motion impact. In order to address the limitations of the standard NSGA-II algorithm, which is prone to local optima and exhibits slow convergence, we propose a good point set method for multi-objective optimization population initialization and a linear ranking selection method to refine the parent selection process within the genetic algorithm. The improved NSGA-II algorithm markedly enhanced both the uniformity of the population distribution and convergence speed. In practical applications, selecting suitable weightings to construct a normalized weight function can identify the optimal solution from the Pareto frontier curve. A high-order continuous and smooth optimal trajectory without abrupt changes can be obtained. The simulation results demonstrated that the 5-7-5 piecewise polynomial interpolation curve effectively constructed a high-order smooth processing trajectory with continuous and smooth velocity, acceleration, and jerk, free from discontinuities. Moreover, the INSGA-II algorithm outperforms the original algorithm in terms of convergence and distribution, enabling the optimal time–jerk multi-objective trajectory planning that adheres to constraint conditions. Optimized by the improved NSGA-II algorithm, the optimal total running time is 4.5400 s, and the optimal jerk is 17.934 $m(rad)/s^3$. This provides a novel approach to solving the inefficiencies and operational instability prevalent in traditional woodworking equipment.

Keywords: woodworking manipulator and worktable; 5-7-5 piecewise polynomial; improved NSGA-II: time–jerk optimal trajectory planning

check for updates

Received: 21 November 2024
Revised: 26 December 2024
Accepted: 27 December 2024
Published: 31 December 2024

Citation: Yi, J.; Zhang, C.; Chen, S.; Dai, Q.; Yu, H.; Yang, G.; Yu, L. Multi-Objective Optimal Trajectory Planning for Woodworking Manipulator and Worktable Based on the INSGA-II Algorithm. *Appl. Sci.* **2025**, *15*, 310. https://doi.org/10.3390/app15010310

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The manipulators applied in wood processing and furniture intelligent manufacturing have greatly improved the degree of industrial automation, such as drilling, milling, assembling, sorting, and so on [1]. Zhang et al. [2,3] have designed a coupled manipulator and worktable apparatus for wood processing that is capable of performing drilling and milling on wooden components, thereby enhancing the efficiency of wood processing.

Trajectory planning represents a pivotal element of the manipulator's motion control system, exerting a substantial impact on joint wear, operational efficiency, and motion smoothness. By optimizing the movement of the manipulators, one can enhance production efficiency, reduce production costs, and extend the service life of the equipment. The specific optimization objectives vary depending on the manipulators being utilized. If the objective is to optimize time, trajectory optimization aims to enhance the operational efficiency of the manipulators. If the objective is to optimize energy consumption, trajectory optimization seeks to reduce energy usage in order to meet the demands of long-duration operations. If the objective is to optimize jerk, trajectory optimization endeavors to enhance the smoothness of the trajectory during the manipulators' motion, thereby minimizing jerk and safeguarding the structural integrity of the mechanical body [4].

At present, the majority of research endeavors concentrate on optimizing a single motion target, such as using time-optimal trajectory planning to enhance operational efficiency and applying jerk-optimal trajectory planning to prolong equipment lifespan. To solve the issue of poor operational stability in traditional woodworking manipulators, Chen et al. [5] proposed to use of an improved particle swarm optimization algorithm. Its approach targets time as the optimization objective in trajectory planning, thereby yielding an optimized trajectory characterized by smooth displacement and velocity, as well as acceleration devoid of abrupt changes. Liu et al. [6] put forth a time-optimal trajectory planning method for robotic arms, utilizing an adaptive elite genetic algorithm with a singularity avoidance mechanism (AEGA-SA). The simulation results indicated that the proposed method exhibited superior performance in terms of both smoothness and efficiency compared to the conventional genetic algorithm (GA). However, the single-objective optimization approach is constrained in that it can only enhance a single target. Due to its simple and easy-to-control structure, PSO has a more stable performance for high-dimensional complex functions. In solving some typical function optimization problems, PSO can achieve better optimization results than genetic algorithms. Although the GA is a mature function optimization algorithm, which often difficult to meet the desired requirements in terms of convergence speed and accuracy of optimization, PSO does not have the complex operation of selection, crossover, mutation, and other operators, which is more adaptable to solving this kind of problem, and it can reach the global optimal value faster [7–9]. In practical applications, it is often necessary to concurrently optimize multiple performance criteria.

Multiple optimization objectives can be summed up by multiplying them by a weighting factor to convert them into a single objective optimization problem. Rong et al. [10] put forward a dynamic weighting method to comprehensively address the optimization problem of time, impact, and dexterity in order to improve the operating efficiency and impact resistance of the robotic arm. At the same time, an improved particle swarm algorithm is proposed to identify the optimal solution and obtain the global optimal solution. Finally, the validity of the method is verified through simulation and prototype experiments. On the basis of five B-spline interpolation trajectories, Gasparetto et al. [11] proposed a weighting method to transform the time–jerk optimization problem into a single-objective problem and obtained the optimal solution through the Sequential Quadratic Programming (SQP) method. This method effectively mitigated the complexity inherent in algorithmic resolution, but it cannot accurately capture the nonlinear or intricate interrelations among multiple objectives. Therefore, certain performance targets cannot achieve the desired optimization outcomes. Furthermore, the fixed weights in the optimization process preclude dynamic adjustments to accommodate varying constraint conditions [12,13]. There are some disadvantages of using the weighting method to convert multi-objective optimization to single-objective optimization. First, the selection of weights depends on the subjective

judgment of the decision maker, and small adjustments of weights can lead to significant changes in the solution. In addition, the weighting method can usually find only one solution, and it is difficult to obtain multiple solutions on the Pareto frontier, especially when there are serious conflicting objectives, and it may not be able to efficiently consider all objectives. In contrast, the multi-objective optimization method can provide a set of non-dominated solutions to help the decision-maker trade-off among multiple objectives and explore more possible solutions through the diversity maintenance strategy, which deals with objective conflicts more comprehensively and reduces the influence of subjective factors. Therefore, despite the simplicity of the weighting method implementation, the multi-objective optimization approach is more advantageous in dealing with complex relationships and providing decision support [14].

In a recent study, Pu et al. [15] introduced an enhanced version of the dingo optimization algorithm based on simulated annealing (SA-DOA). This new approach effectively reduces time and jerk while enhancing the execution efficiency and motion control performance of manipulators by optimizing quintic non-uniform rational B-spline (QNURBS) trajectories. Liu et al. [16] put forth an external archives self-searching multi-objective particle swarm optimization algorithm (EASS-MOPSO), which constructs joint position curves using quartic B-spline curves. It establishes a multi-objective optimization model focusing on time, energy, and jerk, thereby achieving seamless and smooth trajectory optimization without discontinuities. To enhance the operational efficiency and mechanical performance of a six-degree-of-freedom manipulator, He et al. [17] proposed an advanced multi-objective particle swarm optimization (IMPSO) algorithm. It used the improved NSGA-II algorithm to sort the particles and optimize the weight allocation of the multi-objective function, enabling comprehensive evaluation of operational time, joint torque, and jerk. It realized a more optimal trajectory in the joint space. Liu et al. [18] put forth a multi-objective trajectory planning model for manipulators based on the improved multi-objective salp swarm algorithm (LMSSA). This model combines logistic-sine chaotic mapping, small-hole imaging learning strategy, and golden-sine development strategy with the objective of optimizing the control nodes of the seventh-order B-spline curve and completing the multi-objective trajectory planning of manipulators with time–jerk-energy. Wu et al. [19] proposed an SDE-NSGA-II algorithm to solve the problems of the traditional NSGA-II algorithm in multi-objective trajectory optimization, such as being easy to fall into the local optimum, slow convergence speed, etc. The SDE-NSGA-II algorithm adopts the initialization of the population of the good point set as well as the improvement of the evolution strategy with the DC operator. The results show that the improved algorithm solves the Pareto solution set with a more uniform distribution and obtains a more continuous and smooth trajectory. The discrete firefly algorithm has good global search capability and is easy to implement by simulating the glow-in-the-dark behavior of fireflies, but it can easily fall into local optimality in complex problems and is more sensitive to parameter settings. The multi-objective particle swarm optimization (MOPSO) algorithm can deal with multiple objectives simultaneously, maintain the Pareto front, and converge faster, but it is more difficult to maintain the diversity of solutions in high-dimensional objective space, and the parameter setting is more complicated. The multi-objective bottlenose dolphin swarm algorithm has strong global search capability and adaptability by simulating the group behavior of bottlenose dolphins and is suitable for dealing with complex multi-objective problems, but it is relatively complex to implement and may require more computational resources [20–22].

In order to solve the insensitivity problem to the shape of the Pareto front inherent in traditional multi-objective optimization algorithms [23], we propose a multi-objective trajectory optimization method based on an improved non-dominated sorting genetic

algorithm II (NSGA-II). This method uses a 5-7-5 piecewise polynomial to construct motion trajectories for manipulators and worktables, thereby ensuring continuous smoothness in displacement, velocity, acceleration, and jerk. The primary optimization objectives are to minimize motion time and reduce impact, with the goal of achieving an optimal balance between time and jerk. This equilibrium allows the manipulators to complete machining tasks in the shortest possible time while minimizing the impact and vibration generated during joint movements. The multi-objective trajectory optimization method proposed is very important for enhancing production efficiency, extending equipment lifespan, and ensuring machining quality. The time–jerk optimal trajectory planning ensures that the manipulator completes machining tasks in the shortest possible time while avoiding excessive accelerations that could cause jerk, reducing equipment wear and damage to wooden components and significantly enhancing production efficiency. Furthermore, multi-objective trajectory optimization contributes to the reduction in wear on joints and drive systems, thus extending the maintenance cycle and service life of equipment that operates under high-intensity and prolonged conditions. Furthermore, multi-objective trajectory planning guarantees that the manipulator operates at a constant velocity, preventing defects caused by vibration and impact. This ensures the smoothness and precision of wooden component surfaces and improves product quality [24,25].

2. Description and Design of the Woodworking Manipulator and Worktable Under Study

2.1. Physical Model of Woodworking Manipulator and Worktable

Figure 1 illustrates the overall configuration of the woodworking manipulator and worktable, along with the orthogonal relationship between the manipulator and worktable axes of motion (X, Y, Z), with the X-axis serving as the common axis. Accordingly, the worktable can be conceptualized as a manipulator with a right-angle coordinate system, with two moving joints and one rotating joint, analogous to a PPR-type robot arm. This configuration allows the end to move translationally along the X- and Y-axes and rotate about the A'-axis [26]. The manipulator, on the other hand, can be considered a robot arm with two moving joints and two rotating joints, which is analogous to a PPRR-type robot arm. This configuration allows the end to move translationally along the X- and Z-axes and rotate about the A- and B-axes. To simplify the kinematic model of the manipulator and the worktable, the moving components of the manipulator and the worktable are conceptualized as retractable rigid rods, and the moving joints are represented as prismatic and revolute joints, as shown in Figure 2.

2.2. Kinematic Analysis and Modeling

After simplifying the structure of the woodworking manipulator and worktable, the forward kinematics modeling is carried out by the D-H parametric method [27]. Due to the advantages of few parameters, a simple form, and easy mathematical modeling, the D-H parametric method is widely used in the kinematic analysis of manipulators, which is generally divided into the standard D-H parametric method (SDH) and the modified D-H parametric method (MDH), and the main difference between the two is the way of placing the coordinate system, and its definition of a transformation matrix is different. The standard D-H method focuses on the common plumb line between the axes, which is suitable for most manipulator models with simple joint arrangements, but it can lead to cumbersome modeling of complex mechanisms. The improved D-H method simplifies the coordinate system arrangement, reduces redundant calculations, and is suitable for modeling complex robotic systems. In this paper, the modified D-H parameter method is used to model the manipulator and worktable more easily. After homogeneous

transformation by using the modified D-H parameter method, the transformation matrix of the adjacent linkage coordinate system can be obtained. Tables 1 and 2 show the D-H parameter tables of the manipulator and worktable.

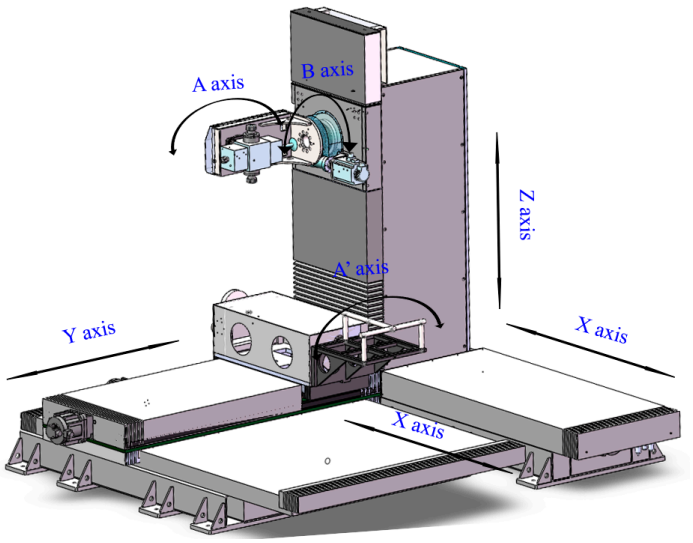


Figure 1. Overall structure of woodworking manipulator and worktable.

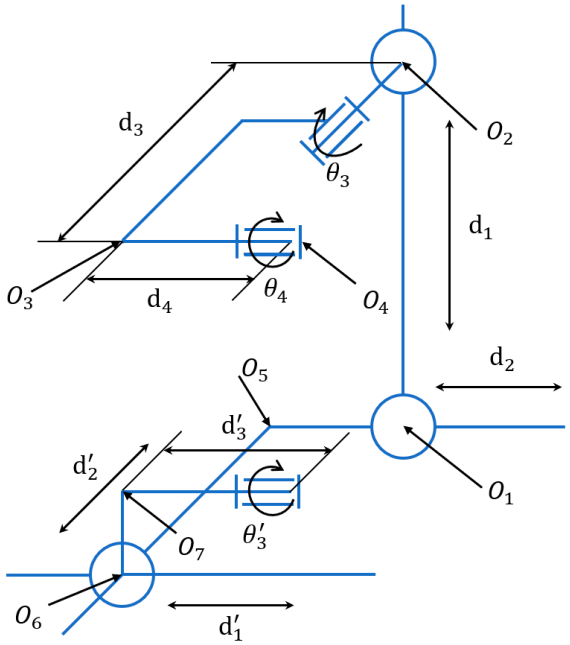


Figure 2. Simplified woodworking manipulator and worktable.

Table 1. Modified D-H parameters of woodworking manipulator.

i	θ_i/(rad)	d_i/(mm)	α_{i-1}/(°)	a_{i-1}/(mm)	Joint Range
1	$-\pi/2$	d_1	0	0	0–1500 mm
2	$\pi/2$	d_2	$-\pi/2$	0	0–1500 mm
3	θ_3	d_3	$-\pi/2$	0	$-\pi$-π
4	θ_4	d_4	$\pi/2$	0	$-\pi$-π

where θ_3 and θ_4 represent the joint angle variables corresponding to revolute joints 3 and 4; the domain of the θ_3 and θ_4 variables spans the interval [$-180°$, $180°$]; d_1 and d_2 represent the joint displacement variables corresponding to joints 1 and 2; the domain of the d_1 and d_2 variables span the interval [0, 1500]; d_3 = 672 mm; and d_4 = 286.5 mm.

Table 2. Modified D-H parameters of worktable.

i	θ_i/(rad)	d_i/(mm)	α_{i-1}/(°)	a_{i-1}/(mm)	Joint Range
1	0	d'_1	$-\pi/2$	0	0–1500 mm
2	0	d'_2	$\pi/2$	0	0–1500 mm
3	θ'_3	d'_3	0	0	$-\pi$-π

where θ'_3 represents the joint angle variables corresponding to revolute joint; the domain of the θ'_3 variable spans the interval $[-\pi, \pi]$; d'_1 and d'_2 represent the joint displacement variables corresponding to joints I and II; the domain of the d'_1 and d'_2 variables spans the interval [0, 1500]; and d'_3 = 500 mm.

The parameters in the above table are inserted into the coordinate system transformation matrices of the joints of the manipulator and worktable to realize the transformation from the base coordinate system to the tool coordinate system, and then the chi-square transformation matrices T_1 and T_2 of the corresponding end-effector are obtained. In the Equations (1)–(4), ${}^{i-1}A_i$ is the transformation matrix of the neighboring linkage coordinate system of the i-th joint, $C\theta$ denotes cosθ, $S\theta$ denotes sinθ, T_1 and T_2 are the tool coordinate system transformation matrices of the manipulator and worktable, respectively; T_{01} and T_{02} are the base coordinate system transformation matrices of the manipulator and worktable, respectively; and $Tool_1$ and $Tool_2$ are the tool coordinate system transformation matrices, respectively.

$$\begin{aligned} T_1 &= T_{01} \cdot {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot Tool_1 \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -C\theta_4 S\theta_3 & S\theta_3 S\theta_4 & C\theta_3 & d_2 + d_4 C\theta_3 + yS\theta_3 S\theta_4 \\ S\theta_3 & C\theta_4 & 0 & d_3 + yC\theta_4 \\ -C\theta_3 C\theta_4 & C\theta_3 S\theta_4 & -S\theta_3 & d_1 + z - d_4 S\theta_3 + yC\theta_3 S\theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

where y and z are the parameters in $Tool_1$, r_{ij} represents the pose information of the end-effector of the manipulator, and P_x, P_y, and P_z represent the position information of the end-effector of the manipulator.

$$\begin{aligned} T_2 &= T_{02} \cdot {}^0A'_1 \cdot {}^1A'_2 \cdot {}^2A'_3 \cdot Tool_2 \\ &= \begin{bmatrix} r\prime_{11} & r\prime_{12} & r\prime_{13} & P\prime_x \\ r\prime_{21} & r\prime_{22} & r\prime_{23} & P\prime_y \\ r\prime_{31} & r\prime_{32} & r\prime_{33} & P\prime_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 & d'_2 + d'_3 \\ S\theta'_3 & C\theta'_3 & 0 & d'_1 + y_2 + y_3 C\theta'_3 \\ -C\theta'_3 & S\theta'_3 & 0 & z_2 + y_3 S\theta'_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2}$$

where y_i and z_i are the parameters in $Tool_2$, r'_{ij} represents the pose information of the end-effector of the worktable, and P'_x, P'_y, and P'_z represent the position information of the end-effector of the worktable.

Since this paper focuses on trajectory optimization based on the position and orientation of the end-effector, an inverse kinematics solution is required to determine the position information for each joint.

$$\begin{cases} \theta_3 = \arctan2(-r_{33}, r_{13}) \\ \theta_4 = \arctan2(r_{12}, r_{11}) \\ d_1 = P_z - z_1 - y_1 C\theta_3 S\theta_4 + d_4 S\theta_3 \\ d_2 = P_x - y_1 S\theta_3 S\theta_4 - d_4 S\theta_3 \end{cases} \tag{3}$$

$$\begin{cases} \theta'_3 = \arctan2(r'_{11}, -r'_{21}) \\ d'_1 = P'_y - z_2 - y_3 S\theta'_3 \\ d'_2 = P'_z - d'_3 + y_2 \end{cases} \tag{4}$$

3. Solving the Time–Jerk Optimal Problem

In industrial applications, the actual operating trajectory of a manipulator is typically divided into two categories: point-to-point discrete trajectory and continuous trajectory. The continuous working trajectory needs to be segmented and realized by trajectory interpolation. In this paper, the trajectory planning of the manipulator and worktable is conducted in the joint space. The joint position q_{ij} of each joint i at each path point j is obtained through the inverse kinematics solution, where $i = 1, 2, ..., 7, j = 1, 2, 3$, and t_j is the movement time of the joints from the path point $j-1$ to path point j.

This paper presents a multi-objective optimization model with the objectives of time and smoothness. The specific optimization objective function is as follows:

$$S_1 = T = \sum_{j=1}^{3} t_j \tag{5}$$

$$S_2 = \sum_{i=1}^{7} \sqrt{\frac{1}{T}\int_0^T [j_i(t)]^2 dt} \tag{6}$$

where S_1 represents the total running time of the trajectory, and S_2 represents the sum of jerk.

In order to guarantee the safety of the manipulator and worktable in practical applications, it is essential to impose constraints on the movements of their respective moving parts which are as follows:

$$\begin{cases} |v(t)| \le v_m \\ |a(t)| \le a_m \\ |j(t)| \le j_m \end{cases} \tag{7}$$

where $v(t)$, $a(t)$, and $j(t)$ represent the joint velocity, acceleration, and jerk, respectively. Similarly, v_m, a_m, and j_m denote the limit values of velocity, acceleration, and jerk, respectively.

4. Joint Trajectory Equations Based on High Degree Polynomials

The piecewise polynomial interpolation method is a common approach to trajectory planning. Its fundamental principle is the use of different numbers of polynomials to generate trajectories in different time periods, with the aim of ensuring the smoothness and continuity of the resulting trajectories. A single high-order polynomial is susceptible to fluctuations and oscillations during the fitting process, and the parameters are challenging to adjust when satisfying complex boundary conditions, which increases the computational complexity [28–30]. Additionally, low-order polynomial curves are unable to guarantee the

higher-order continuity of the joint trajectories of the robotic arm [31,32]. In contrast, the piecewise polynomial is better suited to adapt to local characteristics, enhance fitting accuracy, and reduce computational complexity. In this paper, to circumvent the discontinuities in acceleration that are inherent to 3-5-3 piecewise polynomial trajectory planning, a 5-7-5 piecewise polynomial is used for trajectory planning. Firstly, the time interval of the entire trajectory is divided into three sections, and the trajectory simulation is conducted using quintic, septic, and quintic polynomials in succession, as shown in Figure 3.

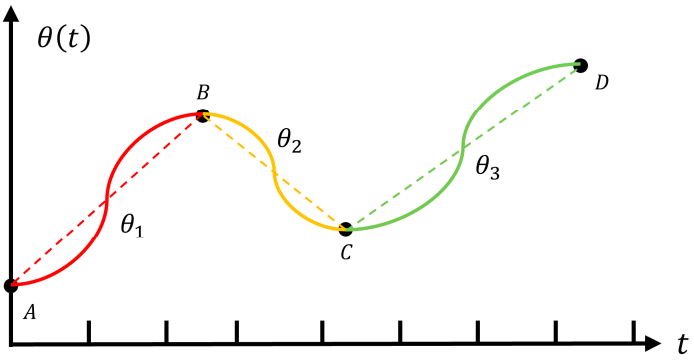


Figure 3. 5-7-5 piecewise polynomial interpolation.

A and D represent the initial and final points of the trajectory, respectively; B and C represent intermediate points along the trajectory. The fifth-degree polynomials are used for interpolation in the A-B and C-D segments, while in the B-C segment, a seventh-degree polynomial interpolation is used. Equation (8) denotes the interpolating function of a 5-7-5 segmented polynomial trajectory of the joint variables with respect to time t, and Equations (9)–(11) denote (angle)velocity, (angle)acceleration, and (angle)jerk, respectively.

$$\begin{cases} \theta_1(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 + a_{14}t^4 + a_{15}t^5 \\ \theta_2(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 + a_{24}t^4 + a_{25}t^5 + a_{26}t^6 + a_{27}t^7 \\ \theta_3(t) = a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3 + a_{34}t^4 + a_{35}t^5 \end{cases} \tag{8}$$

$$\omega_i(t) = d\theta_i(t)/dt \tag{9}$$

$$\alpha_i(t) = d^2\theta_i(t)/dt^2 \tag{10}$$

$$j_i(t) = d^3\theta_i(t)/dt^3 \tag{11}$$

In order to guarantee the continuity and smoothness of the operational trajectories of the manipulator and worktable segments, it is essential to ensure that the following constraints and constraint boundaries are satisfied for the joint variables, namely the angle of velocity, the angle of acceleration, and the angle of jerk:

$$\begin{cases} \theta_1(0) = \theta_A \\ \theta_2(0) = \theta_1(1) = \theta_B \\ \theta_3(0) = \theta_2(1) = \theta_C \\ \theta_3(0) = \theta_D \end{cases} \tag{12}$$

$$\begin{cases} \omega_1(0) = \omega_A = 0 \\ \omega_2(0) = \omega_1(1) = \omega_B \\ \omega_3(0) = \omega_2(1) = \omega_C \\ \omega_3(0) = \omega_D = 0 \end{cases} \tag{13}$$

$$\begin{cases} \alpha_1(0) = \alpha_A = 0 \\ \alpha_2(0) = \alpha_1(1) = \alpha_B \\ \alpha_3(0) = \alpha_2(1) = \alpha_C \\ \alpha_3(0) = \alpha_D = 0 \end{cases} \tag{14}$$

$$\begin{cases} j_1(0) = j_A = 0 \\ j_2(0) = j_1(1) = j_B \\ j_3(0) = j_2(1) = j_C \\ j_3(0) = j_D = 0 \end{cases} \tag{15}$$

In accordance with Equations (12)–(15), the matrix A can be listed, and t_1, t_2, and t_3 in matrices B, D, and F denote the interpolation times of the three-piece polynomials, respectively, and matrix b denotes the (angular) displacement matrices of the joints, and the coefficients α of each of the terms in Equation (8) can be solved according to Equation (25) to obtain the interpolation function of the 5-7-5 piecewise polynomial trajectory.

$$A = \begin{bmatrix} B & C & 0 \\ 0 & D & E \\ 0 & 0 & F \\ G & H & I \end{bmatrix} \tag{16}$$

$$B = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \\ 0 & 0 & 0 & 6 & 24t_1 & 60t_1^2 \end{bmatrix} \tag{17}$$

$$C = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

$$D = \begin{bmatrix} 1 & t_2 & t_2^2 & t_2^3 & t_2^4 & t_2^5 & t_2^6 & t_2^7 \\ 0 & 1 & 2t_2 & 3t_2^2 & 4t_2^3 & 5t_2^4 & 6t_2^5 & 7t_2^6 \\ 0 & 0 & 2 & 6t_2 & 12t_2^2 & 20t_2^3 & 30t_2^4 & 42t_2^5 \\ 0 & 0 & 0 & 6 & 24t_2 & 60t_2^2 & 120t_2^3 & 210t_2^4 \end{bmatrix} \tag{19}$$

$$E = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

$$F = \begin{bmatrix} 1 & t_3 & t_3^2 & t_3^3 & t_3^4 & t_3^5 \\ 0 & 1 & 2t_3 & 3t_3^2 & 4t_3^3 & 5t_3^4 \\ 0 & 0 & 2 & 6t_3 & 12t_3^2 & 20t_3^3 \\ 0 & 0 & 0 & 6 & 24t_3 & 60t_3^2 \end{bmatrix} \tag{21}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{23}$$

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

$$a = A^{-1}b^T = \begin{bmatrix} A_1 & A_2 & A_3 \end{bmatrix}^T \tag{25}$$

$$A_1 = \begin{bmatrix} a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \end{bmatrix} \tag{26}$$

$$A_2 = \begin{bmatrix} a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} \end{bmatrix} \tag{27}$$

$$A_3 = \begin{bmatrix} a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \tag{28}$$

$$b = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \beta_4\ 0\ 0\ 0\ \beta_1\ 0\ 0\ \beta_2\ \beta_3\ 0\ 0] \tag{29}$$

Figure 4 shows the 5-7-5 polynomial trajectory planning outcomes, which are examined in terms of displacement, velocity, acceleration, and jerk. The displacement curve displays a smooth waveform, indicating that the joint motion remains continuous and smooth throughout the process. This can reduce vibration and wear in the system. The smooth change of the velocity curve ensures the controllability and stability of the motion. The smoothness of the acceleration curve reflects that the trajectory planning method ensures efficient and precise operation by accurately controlling each motion parameter, which is of significant application value in automation systems that require high precision and stability. The system exhibits no sudden change in acceleration and deceleration, thereby reducing the impact on the mechanical structure. Furthermore, the uninterrupted abrupt change of the jerk curve indicates that the system effectively avoids violent shocks

and vibrations in the dynamic response, thus improving the smoothness and accuracy of the motion.

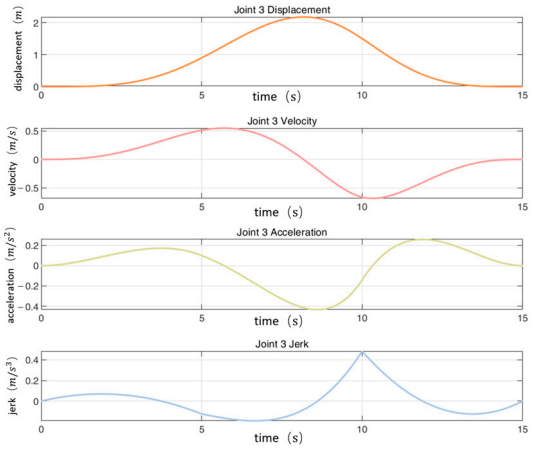


Figure 4. Joint3 trajectory planning results by 5-7-5 polynomial interpolation.

5. Improved NSGA-II Algorithm

The NSGA-II (non-dominated sorting genetic algorithm) is an improved genetic algorithm that is primarily utilized to make optimal decisions under the conditions of weighing between two or more conflicting objectives [33]. Additionally, it is used to address the trade-off optimization problem between multiple conflicting and conflicting objectives.

5.1. Good Point Set Population Initialization

The random initialization of populations in NSGA-II may result in several shortcomings, as outlined in reference [34]. Firstly, randomly generated populations may not adequately cover the search space, particularly in high-dimensional optimization problems. This may result in the algorithm prematurely converging to a locally optimal solution. Furthermore, an unequal distribution of the initial population may impact the efficacy of the non-dominated ordering and congestion distance calculations, which in turn influences the diversity and distribution of the Pareto front. Furthermore, randomly generated individuals may not satisfy the constraints of the problem, necessitating additional correction steps that increase the computational complexity and time cost. These factors may interact to affect the overall performance and efficiency of NSGA-II. Therefore, the use of the good point set for population initialization can significantly ameliorate these problems, enhance population diversity by uniformly distributing individuals, improve the convergence speed of the algorithm, and reduce the complexity of constraint processing, thereby improving the overall effectiveness of the multi-objective optimization algorithm [35,36].

The good point set [37] was first proposed by mathematician Hua Luogeng. Let G_s be the unit cube of s-dimensional Euclidean space, then $r \in G_s$,$P_n(i) = (r_1 i_1, r_2 i_2, r_3 i_3, \ldots r_n i_n)$, $i = 1, 2, 3, \ldots n$. n denotes the number of samples, $P_n(i)$ denotes the set of good points, and r denotes the good points, which is generally taken as $r = \left\{2\cos\left(\frac{2\pi j}{k} i\right), 1 \le i \le n; 1 \le i \le s\right\}$, or $r = \{e^j i\}$, where k is the smallest prime number that satisfies $(k-3)/2 \ge s$.

Figure 5 demonstrates that the good point set initialization exhibits a more uniform data distribution within the 0~1 interval, which is less random than the traditional random initialization method. A comparison of scatterplots demonstrates that the good point set markedly enhances the uniformity of the distribution in multidimensional space while partially compromising the randomness inherent to the process. This approach effectively circumvents the data aggregation issue commonly observed in traditional random sam-

pling, thereby ensuring comprehensive coverage of the search space. Consequently, the good point set initialization not only facilitates a more comprehensive global search for the NSGA-II algorithm but also markedly reduces the probability of falling into a local optimum.

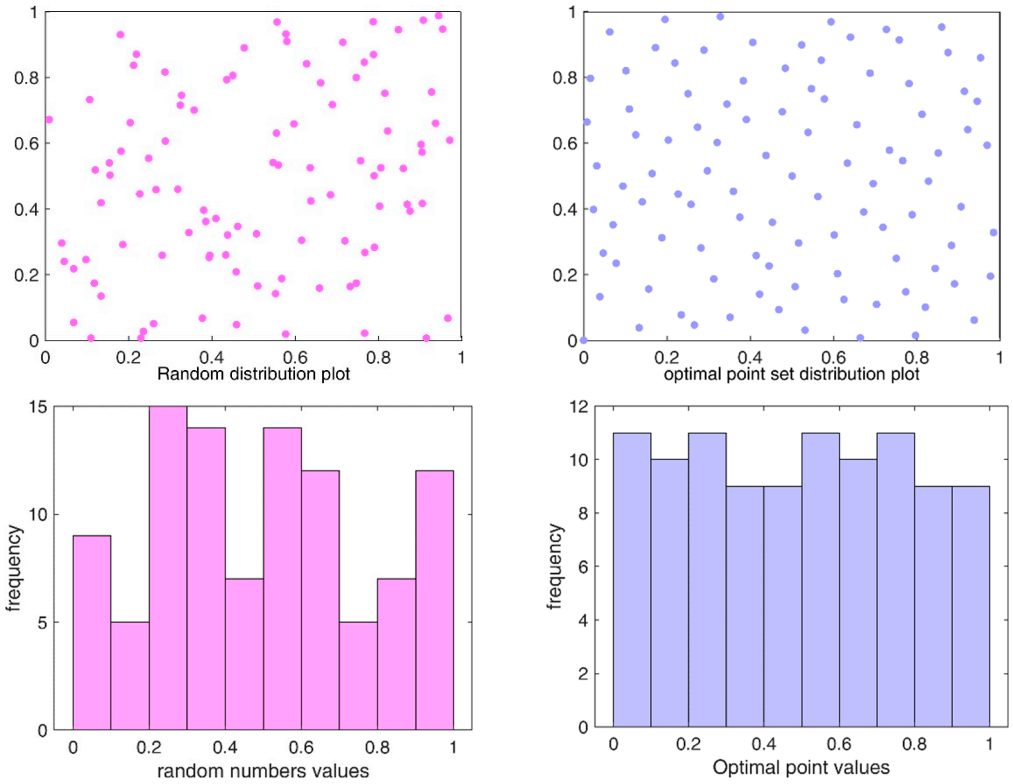


Figure 5. Comparison of two population initialization methods.

5.2. Parental Selection Based on Linear Ranking

The conventional NSGA-II algorithm typically uses tournament selection to identify superior adapted parent individuals from the population and incorporate them into the offspring population. This process is repeated until the new population size aligns with the initial population size [38]. As a consequence of the fact that individuals are released back into the original population each time they participate in the race, the random sampling of put-back trials results in the repeated selection of individuals from the Pareto frontier. This has the effect of reducing population diversity and raising the risk of premature convergence of the algorithm [39]. The implementation of a linear ranking-based parent selection method to ascertain the selection probability through the ranking of individuals [40] markedly diminishes the probability of the NSGA-II algorithm entering into local optimization rather than relying directly on fitness and enables efficient selection even in the presence of considerable differences in fitness. This approach mitigates the risk of premature convergence due to significant discrepancies in fitness while maintaining population diversity.

In linear ranking selection, the population of individuals is sorted from smallest to largest fitness. When applied to the NSGA-II algorithm, this is followed by a descending order of Pareto rank and followed by an ascending order of congestion distance. In the event that the population size is N and x_1 represents the highest ranked individual and x_N is the lowest ranked individual, the probability that individual x_i is as follows:

$$P(i) = \frac{1}{N}\left(n^- + \left(n^+ - n^-\right)\frac{i-1}{N-1}\right) \tag{30}$$

where $\frac{n^-}{N}$ represents the probability that the least optimal individual is selected, whereas $\frac{n^+}{N}$ denotes the probability that the most optimal individual is selected.

As illustrated in Figure 6, there is a notable discrepancy between the tournament selection and the linear ranking selection with respect to the distribution of individual selections. The density curve of the linear ranking selection is more concentrated with higher peaks, indicating that its selection pressure is higher and that individuals are more inclined to gather in high-fitness regions. This may facilitate rapid convergence of the algorithm. In contrast, the density profile of the tournament selection is relatively smooth, with a wider distribution and lower selection pressure, which helps to maintain population diversity. This discrepancy may influence the algorithm's global search capability and rate of convergence. The linear ranking selection is more inclined to achieve rapid convergence, whereas the tournament selection is more conducive to comprehensive exploration of the solution space, increasing the probability of identifying a globally optimal solution.

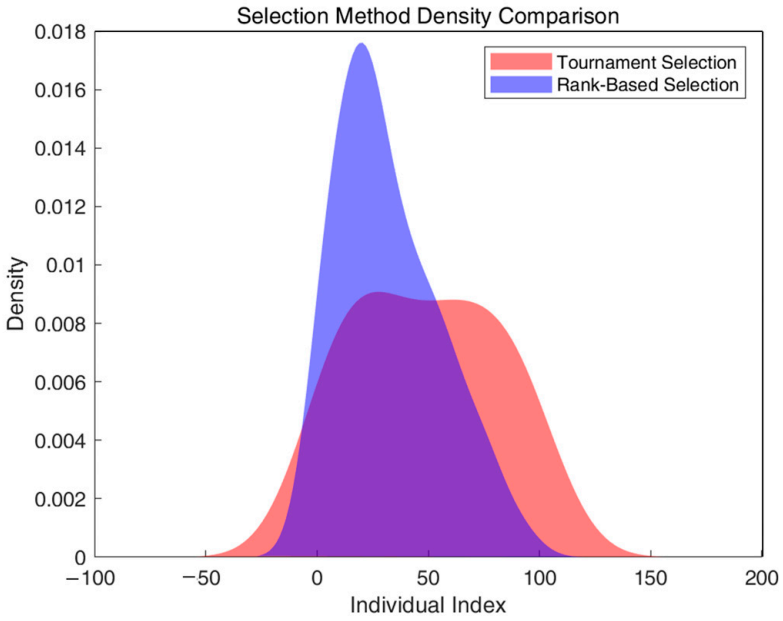


Figure 6. Selection method density comparison.

5.3. Crossover and Mutation

The crossover method chosen is a simulated binary crossover, and the mutation method selected is a polynomial mutation.

The simulated binary crossover operator (SBX) [41] is a crossover operator commonly used in real-coded genetic algorithms to address continuous variables. Its operation is based on the following fundamental principles:

(a) The mean of the values of the two parents and the two children before and after the crossover are equal; i.e.,

$$\frac{p_1 + p_2}{2} = \frac{c_1 + c_2}{2} \tag{31}$$

$$\int \ln\left(x^2 + 1\right) dx \tag{32}$$

(b) The quotient of the interpolated values of the two parents and the interpolated values of the two children before and after the crossover is approximately equal to 1; i.e.,

$$\beta = \left| \frac{c_1 - c_2}{p_1 - p_2} \right| \approx 1 \tag{33}$$

where c_1 and c_2 represent the two offspring individuals, p_1 and p_2 represent the two parent individuals, and the SBX operator is used to invert the two offspring individuals:

$$\begin{cases} c_1 = 0.5[(1+\beta)p_1 + (1-\beta)p_2] \\ c_2 = 0.5[(1+\beta)p_2 + (1-\beta)p_1] \end{cases} \tag{34}$$

where β is a random variable determined dynamically and randomly by the distribution factor η in the following way:

$$\beta = \begin{cases} (2\mu)^{1/(\eta+1)}, \mu \leq 0.5 \\ \left(\frac{1}{2-2\mu}\right)^{1/(\eta+1)}, \mu > 0.5 \end{cases} \tag{35}$$

where μ is a uniformly distributed random number on (0, 1).

Polynomial mutation (PM) [42] is a variation operation for sequential optimization. Its purpose is twofold: firstly, to introduce a small range of stochastic variations in order to ensure the population diversity of the solution space; secondly, to improve the global search capability of the algorithm. The following is a description of the method in question:

(a) The initial parameters must be defined, the probability of variation p_m must be determined, and the distribution index η_m must be set.

(b) Select the variation points of variation and, for each variable, generate a random number r. If $r < p_m$, the variable is varied; calculate the variation δ:

$$\delta = \begin{cases} (2u)^{\frac{1}{\eta m+1}} - 1, \ u < 0.5 \\ 1 - (2-2u)^{\frac{1}{\eta m+1}}, \ u \geq 0.5 \end{cases} \tag{36}$$

where $0 < u \leq 1$.

(c) For the selected variation x_i, the update is $x_i' = x_i + \delta \cdot (x_{max} - x_{min})$, where x_{max} and x_{min} are the upper and lower bounds of the variable.

5.4. Fast Undominated Sorting

As shown in Figure 7, in the context of multiple optimization objective functions, a set of optimization solutions can be obtained through the application of the NSGA-II algorithm. It is important to note that there is no absolute superiority or inferiority between the solutions obtained. Furthermore, it is not possible to improve one objective at the expense of another. These solutions are therefore classified as Pareto-optimal or non-dominated solutions. The Pareto frontier class is used to classify the superiority or inferiority of the solutions in multi-objective optimization. The set of solutions is divided into multiple classes through the application of the Pareto dominance relationship. The initial level comprises all Pareto-optimal solutions, that is, solutions that are not dominated by other solutions. Following the removal of the aforementioned solutions, a novel Pareto optimal solution is identified amongst the remaining solutions, thus establishing the second Pareto level. This process is repeated until all solutions have been assigned ranks [43].

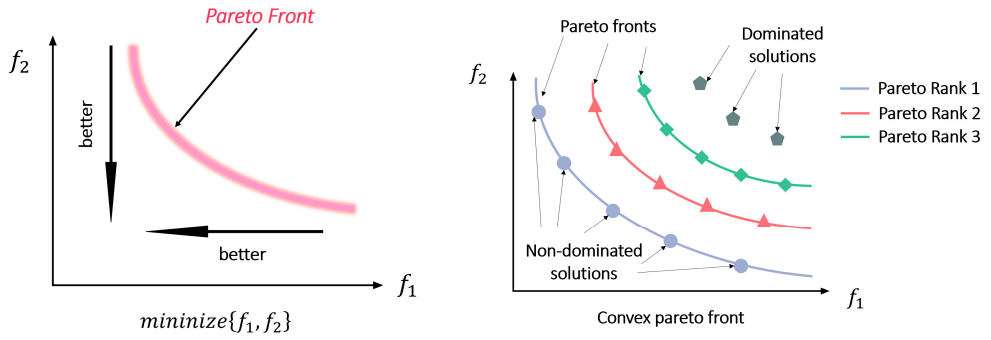


Figure 7. Pareto curve.

5.5. Calculation of Crowding Distance

In the NSGA-II algorithm, the confirmation of crowding distance (CD) represents a crucial stage in the maintenance of population diversity [44]. The crowding distance is initially set to 0 for each solution, and the population is subsequently ranked in accordance with the specified optimization objectives. For the boundary solutions of each objective, the crowding distance is set to infinity, thereby ensuring that these boundary solutions are retained. The crowding distance value is then updated by calculating and normalizing the neighboring differences of each solution on that objective. With regard to both optimization objectives, the crowding distance is defined as the perimeter of the rectangle formed by the adjacent $i-1$ and $i+1$, as illustrated in Figure 8.

$$CD_{im} = \frac{f_m(x_{i+1}) - f_m(x_{i-1})}{f_m(x_{max}) - f_m(x_{min})}, i = 1, 2, \cdots\cdots, (l-1) \tag{37}$$

$$CD_i = \sum_{m=1}^{M} CD_{im} \tag{38}$$

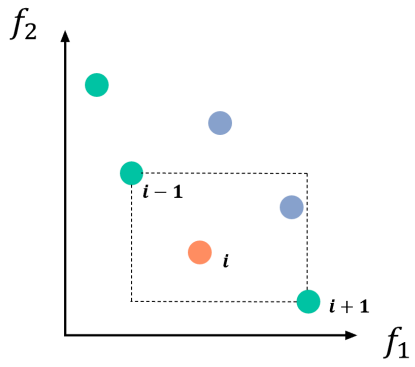


Figure 8. Calculate crowding distance.

Figure 8 illustrates that CD_i is diminished in size and more densely populated in the vicinity of the individual in question. In order to maintain the diversity of the population, it is necessary to compare the crowding degree operators in order to guarantee that the algorithm converges on a uniform Pareto front. Consequently, as a consequence of the non-dominated ordering and crowding calculation, each individual (i) in the population is characterized by two attributes: The Pareto rank and the crowding degree (CD) are two key attributes. In the event that two individuals possess disparate non-dominated rankings, the individual with the lower Pareto rank is selected. Conversely, if two individuals are in the same rank, the individual with the higher crowding degree is chosen, thereby promoting diversity within the population and averting the potential issue of premature convergence.

Figure 9 shows the algorithm flow of Improved Non-dominated Genetic Algorithm (INSGA-II) based on elite strategy with the following steps:

(a) Determine the time interval range $t_i \in [D_i, G_i]$ for each path point based on the maximum velocity v_m of each joint, where D_i and G_i represent the upper and lower time limits. In this paper, we take 1 s as the lower limit and 3 s as the upper limit. The initial parent population, designated as H_0, is generated using the set of good points in the time interval range. The population size is set to N to ensure that the initial solution set is uniformly distributed.

(b) The adaptation value of each population target is calculated according to Equations (5) and (6). Following the non-dominated sorting process, the initial generation of the offspring population, designated as I_1, is generated through the application of three fundamental genetic algorithm operations: selection, crossover, and mutation.

(c) From the second generation onwards, a parent selection method based on linear ranking was used to identify parent individuals for fusion with the offspring population, thereby generating a population P_i with a population size of 2N. The new population adaptation value was then calculated.

(d) The elite selection strategy entails non-dominated sorting of 2N individuals in the population P_i according to fitness, with the crowding degree of individuals in the same dominance layer calculated according to Equation (38). New individuals are then selected based on dominance relationships and crowding degree values, forming the parent population H_{i+1} with a population number N.

(e) Repeat steps 2–3 until the termination condition is satisfied, and output the Pareto optimal solution set.

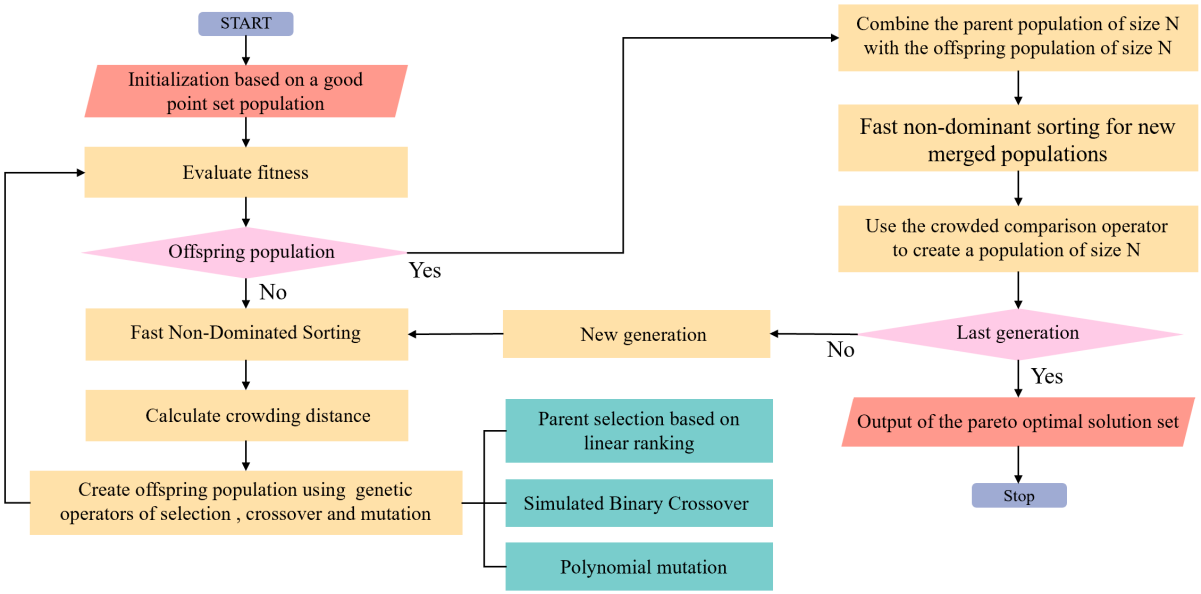


Figure 9. Flowchart of the improved NSGA-II algorithm.

5.6. Optimization Results and Analysis

In order to verify the performance of the improved NSGA-II algorithm, this experiment employs classical test functions, such as the ZDT series, to evaluate the algorithm's performance prior to and following the implementation of enhancements. The algorithm parameters were set as follows: the population size is 200; the mating pool size is 100; the crossover rate is 0.9; the mutation rate is 0.1; the crossover and mutation distribution indexes are both 20; and the number of iterations is 500. As illustrated in Figure 10, the improved NSGA-II demonstrates superior performance compared to the conventional NSGA-II in terms of the Pareto front. The improved NSGA-II exhibits enhanced convergence and a solution that is closer to the theoretically optimal solution. Additionally, the

distribution of the solution is more uniform and covers a wider range, indicating that the improved NSGA-II has accelerated convergence and higher solution accuracy.

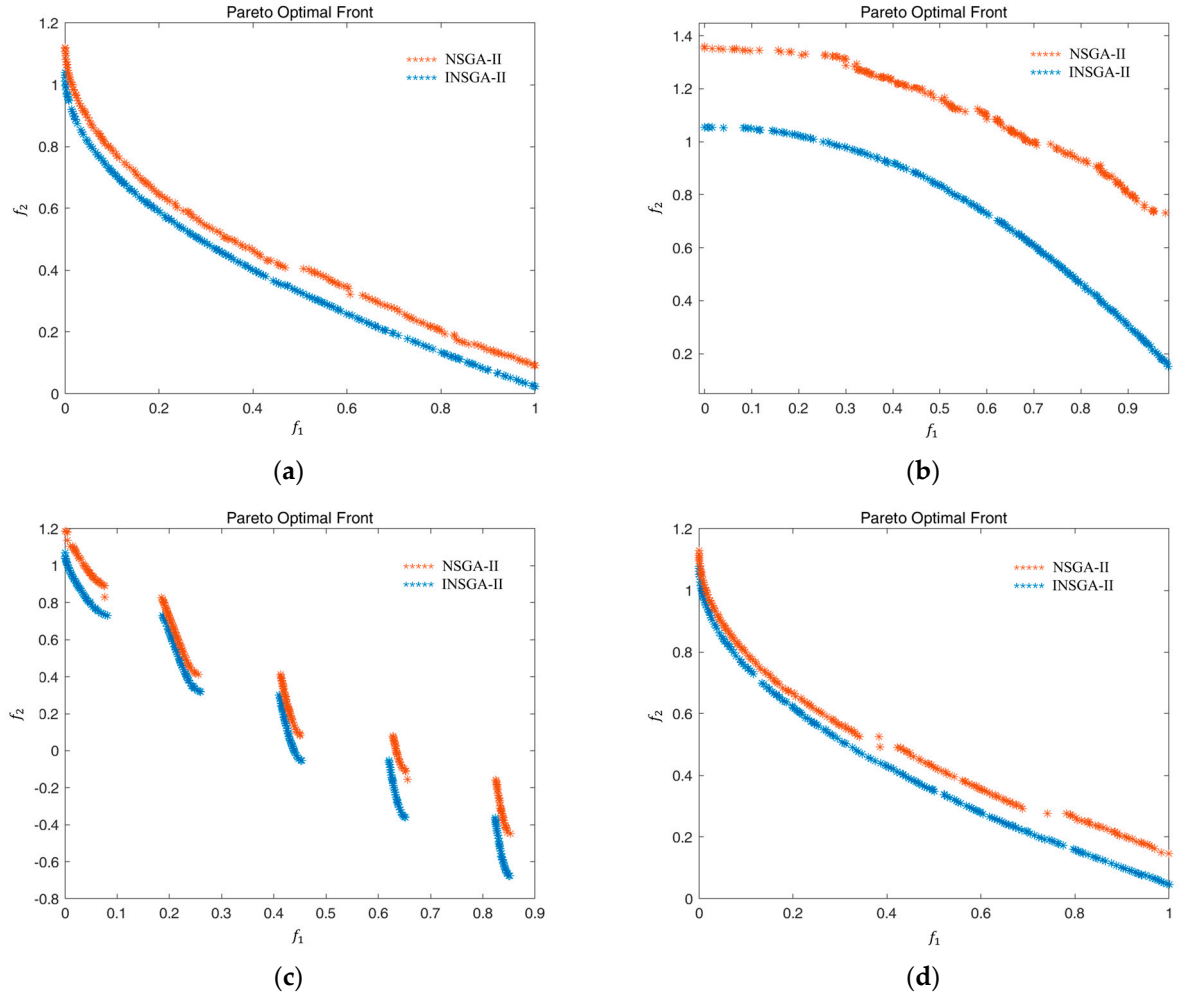


Figure 10. Test function comparison: (**a**) ZDT1, (**b**) ZDT2, (**c**) ZDT3, and (**d**) ZDT6.

The comparison of the Pareto frontiers of the objective function optimized by the NSGA-II and INSGA-II algorithms is presented in Figure 11. The improved NSGA-II algorithm (INSGA-II) exhibits notable advantages in multi-objective optimization through the initialization of the good point set population and the selection of parents based on linear ranking. The good point set approach ensures diversity and coverage of the solution by rationally distributing the initial population, thereby generating a more uniform Pareto front and accelerating the convergence of the algorithm. The selection of parents based on linear ranking increases the selection pressure, thereby increasing the likelihood of superior individuals being selected. This approach balances global search and local exploitation and improves the convergence speed and stability. These enhancements facilitate the convergence of the INSGA-II algorithm to the Pareto front in a more expeditious and uniform manner, reduce the number of iterations, circumvent premature convergence, and enhance the overall optimization performance of the algorithm.

Figure 12 shows the evolution of the average objective value of the multi-optimization objective over the course of the multi-generation evolution of the INSGA-II algorithm. The vertical axis represents the average objective value, while the horizontal axis represents the number of population iterations per generation. The objective value of the objective function S_1 remains relatively stable and has reached the global optimum near 10 generations. The objective function S_2 exhibits a rapid decline before 50 generations, followed by a gradual

stabilization, indicating that S_2 demonstrates superior optimization efficacy at the outset and is in proximity to the global optimum. In consideration of the convergence of the multi-objective function, the rate of convergence, and the optimal solution, this paper proposes 50 generations as the termination condition for the INSG-II algorithm's population iteration.

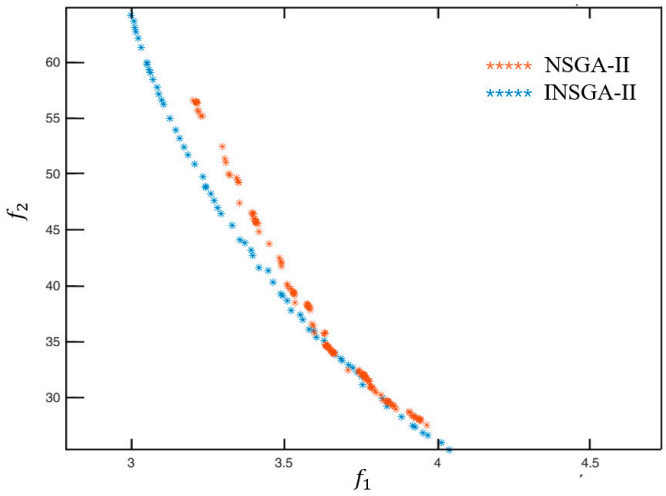


Figure 11. Objective function comparison.

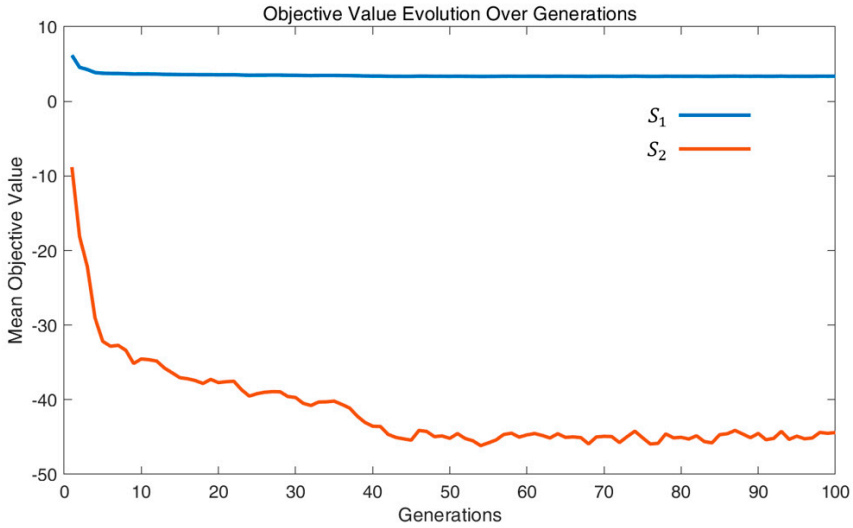


Figure 12. Effect of multi-objective iterative optimization.

5.7. Solving Pareto Optimal Solution Set

In this study, the woodworking manipulator and worktable are taken as the research object. Four path points, designated A(a), B, C(c), and D, are set in the space. The actuator ends of the woodworking manipulator and worktable overlap at the two points B and D, which are used to simulate the machining operation at these two points. Consequently, the manipulator and the worktable can be regarded as woodworking equipment with seven joints. The values of the joint variables of the joints are shown in Table 3.

Table 3. Seven joint positions and angle of path points.

Path Points	1	2	3	4	5	6	7
A(a)	0.800 m	1.200 m	3.141 rad	0.000 rad	0.400 m	0.500 m	0.000 rad
B	0.578 m	0.868 m	1.897 rad	1.047 rad	0.236 m	0.359 m	−1.402 rad
C(c)	0.668 m	0.567 m	1.468 rad	1.711 rad	0.344 m	0.407 m	−0.785 rad
D	0.459 m	0.964 m	2.355 rad	0.754 rad	0.325 m	0.310 m	−0.098 rad

The enhanced NSGA-II algorithm is used to identify the optimal Pareto solution set within the context of multi-objective optimization. The INSGA-II algorithm has been simulated, and the algorithm's parameters have been configured as follows: The population size is 200, the upper bound is 3, the lower bound is 1, the crossover rate is 0.9, the variance rate is 0.1, the indexes of hybridization and variance distributions are both 20, and the number of iterations is 50. The optimization objective is to minimize both the shortest time and the smallest jerk. The optimization objective function is S_1 and S_2, and the decision variables are three trajectory times (t_1, t_2, and t_3). The velocity (v), acceleration (α), and jerk (j) constraints are imposed on the individual joints to obtain the Pareto frontier curves after trajectory optimization through the NSGA-II algorithm. These results are shown in Figure 13.

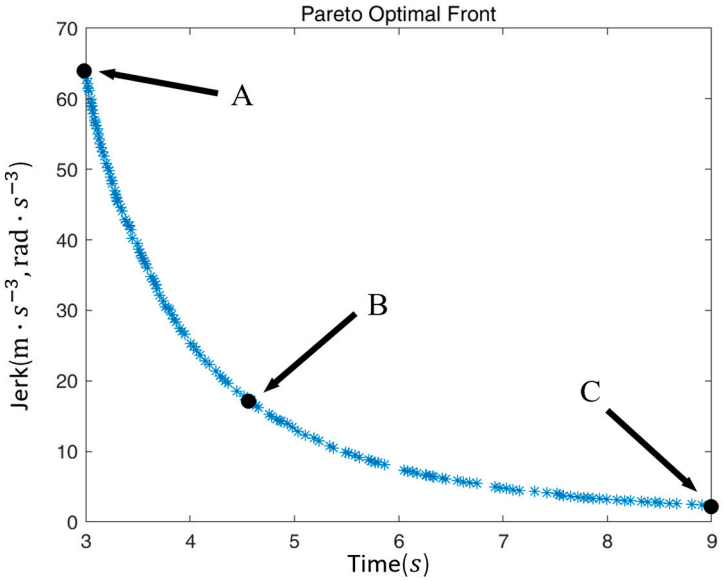


Figure 13. Time–jerk Pareto optimal solution set based on INSGA-II algorithm.

As illustrated in Figure 12, point A represents the time-first optimal solution, point C denotes the jerk-first optimal solution, and point B signifies the balanced optimization good point. As the point of reference is approached, the time required is reduced, and the joint jerk increases. Conversely, as the point of reference is approached, the time required is increased, and the joint jerk decreases. The time and jerk are two coupled and contradictory motion indexes that restrict each other's optimal performance.

In order to eliminate the influence of different features on parameter selection and to facilitate the processing and comparison of data, the trajectory running time and jerk are normalized and scaled to a target range of [0, 1]. This normalization allows for a fair comparison of the advantages and disadvantages of different solutions. The normalization transformation formula is as follows:

$$S^* = \frac{S - \min(S)}{\max(S) - \min(S)} \tag{39}$$

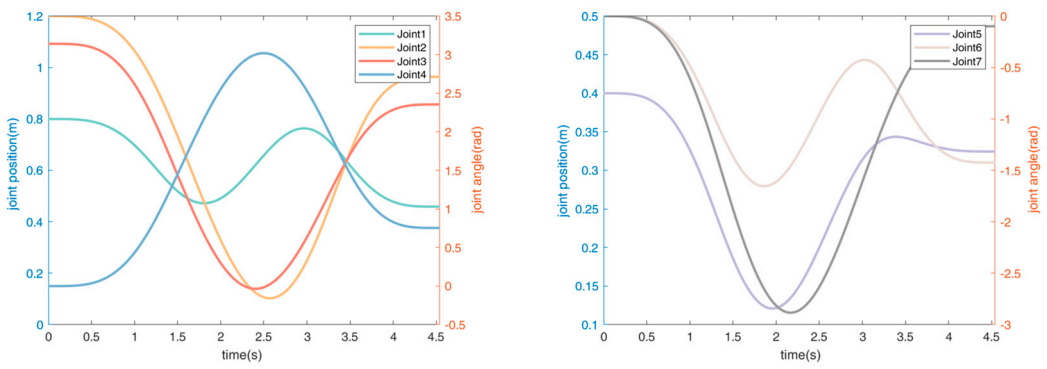
where S^* is the normalized [0, 1] range function value, and S is the original function value.

In order to select a solution from the Pareto curve that is applicable to actual working conditions, it is necessary to introduce weight parameters ω_1 and ω_2 to define a weight function as follows:

$$S_c = \omega_1 S_1^* + \omega_2 S_2^* \tag{40}$$

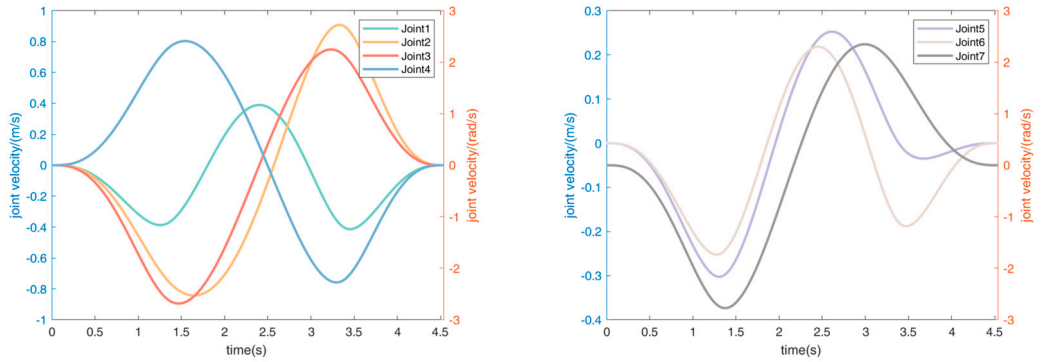
where S_c is the normalized weight function, S_1^* is the normalized running time, and S_2^* is the normalized total jerk.

In this paper, we take $\omega_1 = \omega_2 = 0.5$, and in Figure 12, point B indicates this optimal solution, the optimal total running time is 4.5400 s, the optimal jerk is 17.934 $\mathrm{m(rad)/s^3}$, and the optimal time series is [1.327, 2.054, 1.159], which is carried into Equation (8), and can be obtained as a 5-7-5 polynomial piecewise interpolation trajectory, and the joint variables, velocity, acceleration, and jerk of each joint are shown in Figures 14–17.



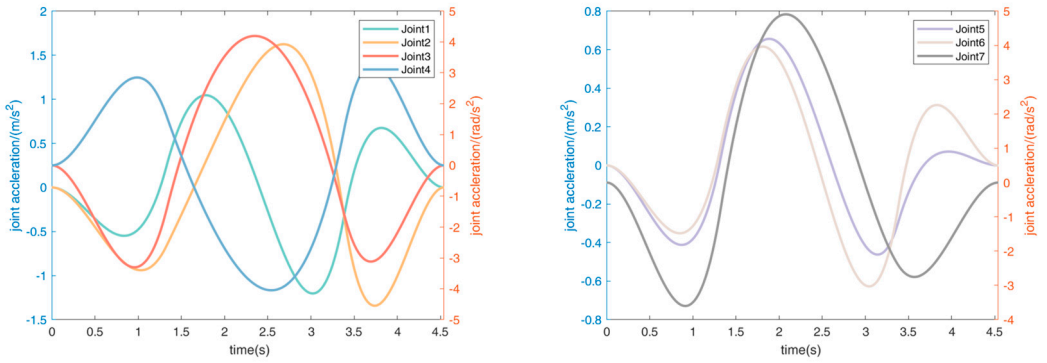
(**a**) (**b**)

Figure 14. Displacement curve: (**a**) woodworking manipulator displacement; (**b**) woodworking worktable displacement.



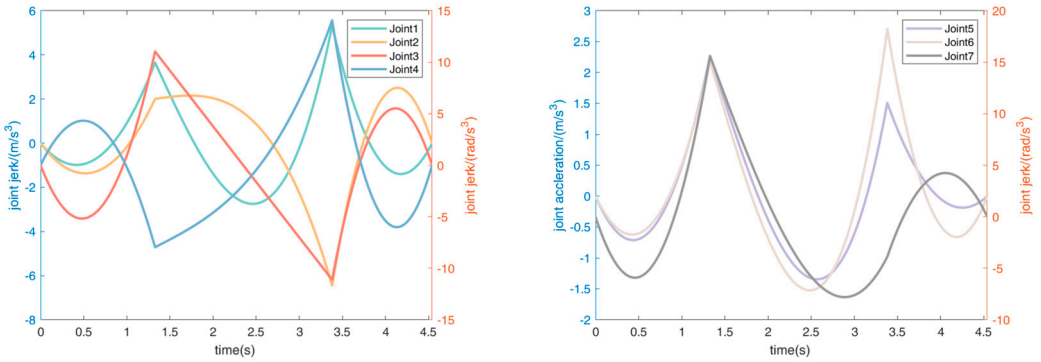
(**a**) (**b**)

Figure 15. Velocity curve: (**a**) woodworking manipulator velocity; (**b**) woodworking worktable velocity.



(**a**) (**b**)

Figure 16. Acceleration curve: (**a**) woodworking manipulator acceleration; (**b**) woodworking worktable acceleration.



(**a**) (**b**)

Figure 17. Jerk curve: (**a**) woodworking manipulator jerk; (**b**) woodworking worktable jerk.

Figures 14–17 illustrate the displacement, velocity, acceleration, and jerk curves of the manipulator and worktable. The displacement, velocity, and acceleration curves of each joint are characterized by smooth, continuous, and uninterrupted behavior, and the jerk is also continuous and uninterrupted. In examining the joint displacements, it becomes evident that the manipulator (Joint1–Joint4) and the worktable (Joint5–Joint7) exhibit discernible periodic fluctuations. However, both components reach the desired positions at all four interpolation points, indicating that the system is functioning as intended. The joint velocity curves and acceleration curves illustrate the dynamic response characteristics of the system. Provided that the system constraints are satisfied, the velocity and acceleration changes of the manipulator and the worktable are relatively smooth and approach the extreme values, indicating that the joint operation efficiency has been enhanced. In the context of manipulator systems, the discontinuity of the jerk can have a significant impact on the stability of the operation. To ensure the continuity of the jerk, it is essential to increase the order of the piecewise polynomials, thereby preventing the occurrence of jumps and interruptions that may arise under a 3-5-3 trajectory planning approach.

The optimized trajectory curves of the manipulator and worktable are imported into the MATLAB Robotic Toolbox, where they are treated as telescopic and rotating connecting rods. The respective end-effector trajectories of the manipulator and the worktable are then determined, as illustrated in Figure 18. The manipulator passes through the four points A, B, C, and D, and the worktable passes through the four points A, B, C, and D. The simulation results demonstrate that the proposed trajectories in this paper are highly similar to the trajectory curves of the manipulator and the worktable. The simulation results demonstrate that the INSGA-II algorithm proposed in this paper is capable of achieving stable operational trajectory planning and cooperative control between the manipulator and the worktable, while satisfying the time–jerk optimization.

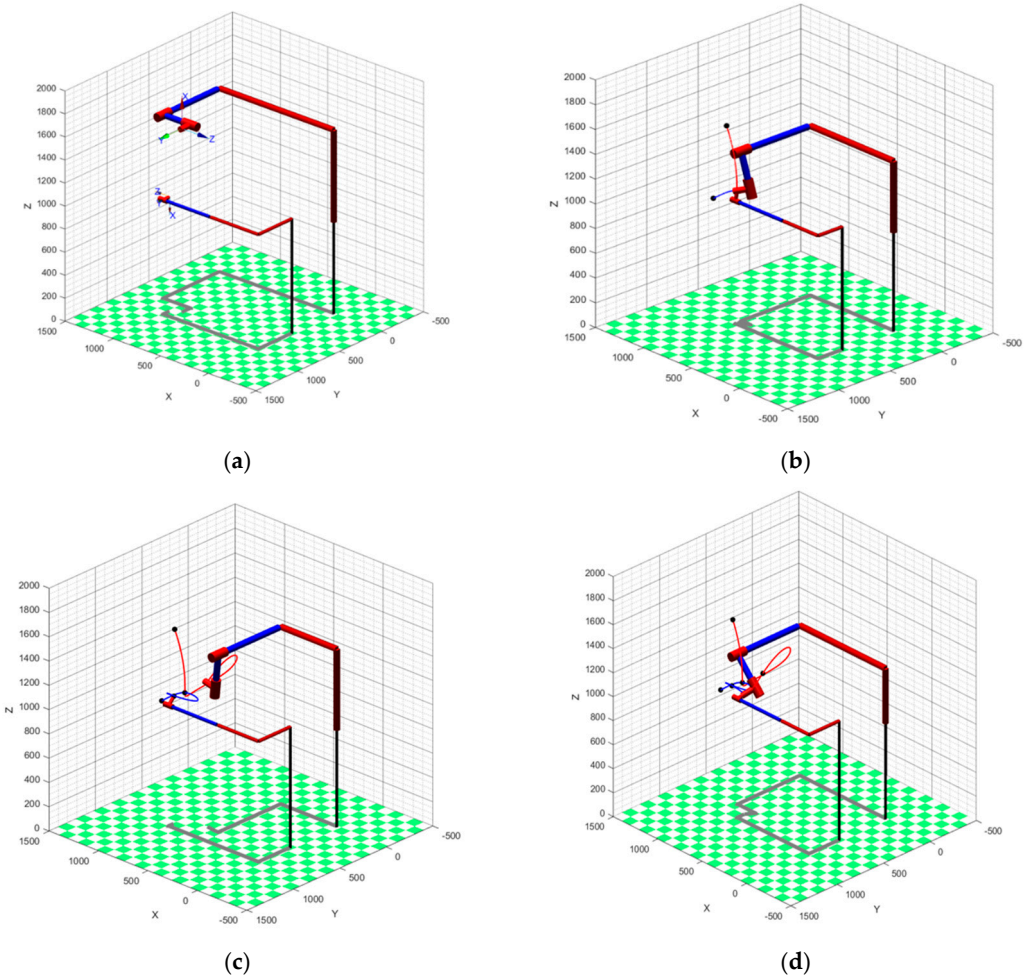


Figure 18. Spatial trajectory of the manipulator and worktable. (**a**) The woodworking manipulator and worktable end-effector at start point A and a respectively; (**b**) the woodworking manipulator and worktable end-effector at point B; (**c**) the woodworking manipulator and worktable end-effector at point C and c respectively; (**d**) the woodworking manipulator and worktable end-effector at end point D.

6. Conclusions

This paper studied the optimization trajectory for a woodworking manipulator and worktable, with the objective of minimizing both time and jerk. The improved D-H parametric method is utilized for both forward and inverse kinematics analysis, and a trajectory optimization method is proposed that integrates the 5-7-5 piecewise polynomial interpolation method and the improved NSGA-II algorithm, which incorporates multiple strategies. To ensure the smoothness and continuity of the trajectories, we increase the order of the polynomials from the usual 3-5-3 to 5-7-5. This adjustment increases the computational complexity while improving the continuity and smoothness of the trajectories. For high-degree of freedom robotic arms, this complexity can limit the real-time and scalability of the algorithm. Therefore, a balance between performance and computational complexity must be found to ensure the effectiveness of the algorithm in practical applications. In order to solve the NSGA-II algorithm susceptibility of local optimization and slow convergence speed, a novel approach is proposed. By substituting the random population initialization in the original algorithm with a good point set, the non-uniform distribution of the Pareto frontier is effectively avoided. The parent selection mechanism based on sex ranking is proposed to replace the tournament selection method in the original algorithm, which accelerated the convergence process of the algorithm. This resulted in a notable acceleration of

the algorithmic convergence process. The objective of this paper is to optimize the time–jerk of the manipulator and worktable. The results showed that the 5-7-5 piecewise polynomial interpolation method can achieve a high-order, smooth, and continuous running trajectory for each joint. Moreover, this approach can be constrained in terms of both start–stop speed and acceleration, thereby preventing any abrupt and uninterrupted changes in impact. The trajectory obtained by improving the INSGA-II algorithm is smooth, and the joint motion is more stable, thus reducing mechanical damage. The simulation results demonstrated that the improved NSGA-II algorithm was more effective than the original algorithm in completing the task of multi-objective trajectory optimization of the woodworking manipulator and worktable.

Author Contributions: Conceptualization, J.Y. and C.Z.; funding acquisition, C.Z.; methodology, J.Y.; resources, C.Z.; software, J.Y., S.C. and Q.D.; supervision, C.Z.; validation, J.Y., S.C. and L.Y.; visualization, J.Y., S.C. and Q.D.; writing—original draft, J.Y.; writing—review and editing, J.Y., H.Y. and G.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2024YFD2200700.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: First of all, I would like to express my sincere gratitude to all the people who helped me in the process of writing this paper. In particular, my heartfelt thanks go to Yanxia Du, whose advice and encouragement have helped me to gain a deeper understanding of these studies. It is my great honor and pleasure to study under his guidance and supervision. Secondly, I am also very grateful to all my friends and classmates who have given me a lot of help and companionship in preparing this thesis. Finally, I would like to express my heartfelt gratitude to all those who have spent a lot of time reading this thesis and have given me a lot of advice that will benefit me in my future studies.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ni, H.; Xiong, X. A digital technology infrastructure of packaging section in mass custom furniture. *Chin. J. Wood Sci. Technol.* **2022**, *36*, 31–36.
2. Zhang, C.; Guo, Q.; Yu, H.; Yang, G.; Bai, C. Kinematics analysis of operating arm for high-speed multi-spindle. *For. Grassl. Mach.* **2019**, *30*, 1–3.
3. Zhang, C.; Yu, H.; Yang, G.; Bai, S.; Li, G. Motion control system and structure design on the operating arm of high-speed multi-spindle with multiple-head. *For. Grassl. Mach.* **2018**, *29*, 1–3.
4. Long, Z.; Li, X.; Shuai, T.; Wen, F.J.; Feng, W.R.; Lang, C.P. Review of research state of trajectory planning for industrial robots. *Mech. Sci. Technol. Aerosp. Eng.* **2021**, *40*, 853–862.
5. Chen, S.; Zhang, C.; Yi, J. Time-Optimal Trajectory Planning for Woodworking Manipulators Using an Improved PSO Algorithm. *Appl. Sci.* **2023**, *13*, 10482. [CrossRef]
6. Liu, Y.; Guo, C.; Weng, Y. Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance. *IEEE Access* **2019**, *7*, 146301–146308. [CrossRef]
7. Kachitvichyanukul, V. Comparison of three evolutionary algorithms: GA, PSO, and DE. *Ind. Eng. Manag. Syst.* **2012**, *11*, 215–223. [CrossRef]
8. Khan, K.; Ashok, S. A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. *Int. J. Intell. Syst. Appl.* **2012**, *4*, 23–29. [CrossRef]
9. Hassan, R.; Cohanim, B.; De Weck, O.; Venter, G. A comparison of particle swarm optimization and the genetic algorithm. In Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structural Dynamics and Materials Conference, Austin, TX, USA, 18–21 April 2005.

10. Rong, Y.; Chen, G.; Dou, T. A Multi Index Comprehensive Optimal Anti Impact Trajectory Planning Method. *China Mech. Eng.* **2024**, *35*, 305–316.
11. Gasparetto, A.; Zanotto, V. A new method for smooth trajectory planning of robot manipulators. *Mech. Mach. Theory* **2007**, *42*, 455–471. [CrossRef]
12. Rahimi, I.; Gandomi, A.; Chen, F.; Mezura-Montes, E. A review on constraint handling techniques for population-based algorithms: From single-objective to multi-objective optimization. *Arch. Comput. Methods Eng.* **2023**, *30*, 2181–2209. [CrossRef]
13. Wang, X.; Feng, Y. Time-Jerk-Optimal trajectory planning for robot manipulator. *Commun. Inf. Technol.* **2024**, 33–38.
14. Zhang, H.; Zhao, J.; Luo, H.; Xie, X. Objective weight computation based on personal preference for multi-objective optimization problem. *Control Dec.* **2014**, *29*, 1471–1476.
15. Pu, Q.; Xu, X.; Li, Q.; Zhang, H. Robotic arm time–jerk optimal trajectory based on improved dingo optimization. *J. Braz. Soc. Mech. Sci. Eng.* **2024**, *46*, 198. [CrossRef]
16. Liu, Y.; Zhang, X. Trajectory optimization for manipulators based on external archives self-searching multi-objective particle swarm optimization. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2022**, *236*, 1188–1201. [CrossRef]
17. He, Z.; Tang, X.; Shen, Q.; Duan, C.; Jia, C. Optimization of a Six-degree-of-Freedom robot trajectory based on Improved Multi-objective PSO Algorithm. *Int. J. Robot. Autom.* **2023**, *38*, 218–230.
18. Liu, J.; Huang, H.; Fan, Q.; Ma, C.; Zhang, L. Multi-objective trajectory planning of robotic arms based on improved salp swarm algorithm. *China Mech. Eng.* **2024**, *12–28*, 1–13.
19. Wu, F.; Yuan, L.; Shen, D. Multi-target Trajectory Optimization of Manipulator with Improved NSGA-II Algorithm. *Mech. Sci. Technol. Aerosp. Eng.* **2024**, *12–13*, 1–12.
20. Ma, H.; Zhang, Y.; Sun, S.; Liu, T.; Shan, Y. A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artif. Intell. Rev.* **2023**, *56*, 15217–15270. [CrossRef]
21. Hojjati, A.; Monadi, M.; Faridhosseini, A.; Mohammadi, M. Application and comparison of NSGA-II and MOPSO in multi-objective optimization of water resources systems. *J. Hydrol. Hydromech.* **2018**, *66*, 323–329. [CrossRef]
22. Wu, X.; Liu, D. Feature selection based on hybrid optimization of salp swarm algorithm and particle swarm optimization. *J. Chongqing Univ. Posts Telecommun. (Nat. Sci. Ed.)* **2021**, *33*, 844.
23. Lu, J.; Ping, X. Time-jerk-optimal Trajectory Planning Algorithm for Manipulators. *Mech. Sci. Technol. Aerosp. Eng.* **2019**, *38*, 1548–1554.
24. Liang, H.; Huang, H.; Zhou, M. The optimizing of seven-axis manipulator minimum energy linear trajectory algorithm. *Cyber Secur. Data Gov.* **2015**, *34*, 1–4.
25. Li, N.; Zhao, X. Motion Planning of Mobile Manipulator Based on Genetic Algorithms. *Dev. Innov. Mach. Electr. Prod.* **2007**, *20*, 24–26.
26. Hu, X.; Zhang, C.; Yu, H.; Yang, G. Kinematics simulation analysis of multi-output manipulator coupling worktable system based on Adams. *Eng. J. Wuhan Univ.* **2024**, *57*, 979–987.
27. Singh, A.; Singla, A.; Soni, S. Extension of DH parameter method to hybrid manipulators used in robot-assisted surgery. *Proc. Inst. Mech. Eng. Part H J. Eng. Med.* **2015**, *229*, 703–712. [CrossRef]
28. Li, W.; Deng, Z.; Ge, J.; Liu, W.; Li, C. Research progress of joint space trajectory planning for robots. *Mach. Des. Manuf. Eng.* **2022**, *51*, 15–23.
29. Springer, K.; Gattringer, H.; Staufer, P. On time-optimal trajectory planning for a flexible link robot. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2013**, *227*, 752–763. [CrossRef]
30. Cao, B.; Doods, G.I.; Irwin, G.W. Time-Optimal and Smooth Constrained Path Planning for Robot Manipulators. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; IEEE: New York, NY, USA, 1994; pp. 1853–1858.
31. Ma, R.; Hu, X.; Yin, G. The Shortest Time Optimal Planning of 6-DOF Industrial Robot's Trajectory. *Mach. Des. Manuf.* **2014**, *4*, 30–32+35.
32. Sun, Y.; Cheng, W.; Li, G. The Algorithm and Research Status of Trajectory Planning of Joint Robot. *J. Yangtze Univ. (Nat. Sci. Ed.)* **2016**, *13*, 32–38+5.
33. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
34. Zhang, X.; Ma, L. An improved NSGA-II algorithm for multi-objective nonlinear optimization. *Microelectron. Comput.* **2020**, *37*, 45–53.
35. Li, S.; Chen, Y.; Yang, Y.; Yu, M.; Zhu, Y. Point cloud registration method based on artificial fish swarm algorithm using good point set. *J. Hefei Univ. Technol. (Nat. Sci.)* **2023**, *46*, 1203–1209.
36. Chen, L.; Cao, K.; Zhang, S.; Bai, H.; Han, Y. Recent Progress of Swarm Intelligent Optimization Algorithms. *Comput. Eng. Appl.* **2024**, *60*, 46–67. [CrossRef]
37. Hua, L.; Wang, Y. *Applications of Number Theory to Numerical Analysis*; Science Press: Beijing, China, 1978; pp. 83–99.

38. Liu, Y.; Gong, D.; Sun, J.; Jin, Y. A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. Cybern.* **2017**, *47*, 2689–2702. [CrossRef]
39. Chen, X. Research on multi-objective optimization model of drilling parameters based on intelligent algorithm. *Chongqing Univ. Sci. Technol.* **2023**. [CrossRef]
40. Back, T. Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 27–29 June 1994; IEEE: New York, NY, USA, 1994; pp. 57–62.
41. Deb, K.; Agrawal, R.B. Simulated binary crossover for continuous search space. *Complex Syst.* **1995**, *9*, 115–148.
42. Deb, K.; Goyal, M.A. Combined genetic adaptive search (GeneAS) for engineering design. *Comput. Sci. Inform.* **1996**, *26*, 30–45.
43. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference, Paris, France, 18–20 September 2000; Proceedings 6*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 849–858.
44. Fortin, F.A.; Parizeau, M. Revisiting the NSGA-II Crowding-Distance Computation. In Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, Amsterdam, The Netherlands, 6–10 July 2013; pp. 623–630.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.