Article

Time-Impact Optimal Trajectory Planning for Wafer Handling Robotic Arms Based on Improved Snake Optimization Algorithm

Yujie Ji * and Jiale Yu

Posted Date: 12 February 2025

doi: 10.20944/preprints202502.0880.v1

Keywords: ISO algorithm; S-shaped speed curve; trajectory planning; time-impact optimal doi:10.20944/preprints202502.0880.v1

Article

Time-Impact Optimal Trajectory Planning for Wafer Handling Robotic Arms Based on Improved Snake Optimization Algorithm

Yujie Ji * and Jiale Yu

School of Mechanical Engineering, Shenyang Ligong University, Shenyang, Liaoning 110159, China

* Correspondence: jyj19700521@163.com

Abstract: To enhance the working efficiency of a wafer transport robotic arm and simultaneously alleviate the impact and vibration during the motion process, a trajectory planning approach based on an improved snake optimization (ISO) algorithm is proposed. The following improvements have been made to the snake optimization (SO) algorithm: the introduction of a Chaotic Tent Map for initializing the swarm, the use of randomly perturbed dynamic learning factors to replace fixed values, the application of a cosine annealing learning rate for self-adaptively updating individual positions, and the incorporation of Bayesian optimization for parameterization and fine-tuning of the system's selection process. Furthermore, the ISO is introduced into Cartesian space of the robotic arm to effectively address the trajectory planning issue of the single-segment start-stop S-shaped speed curve with arc transitions. The simulation results indicate that the improved S-shaped speed curve has increased by 24.1% compared with the original plan, and the mean and variance rankings of ISO algorithm have respectively improved by 60.8% and 63.4% compared with the SO algorithm. Meanwhile, this study has successfully achieved the Pareto optimal solution with time and impact as the targets based on the established MATLAB experimental simulation platform.

Keywords: ISO algorithm; S-shaped speed curve; trajectory planning; time-impact optimal

1. Introduction

With the rapid advancement of the semiconductor industry, wafer-handling manipu-lator arms have been confronted with elevated demands in terms of precision, flexibility, and stability [1–4]. Consequently, reliable tracing of the anticipated trajectory during operation has become a key concern in the current research. When deliberating the trajectory planning of the manipulator arm, a choice must be made between the planning schemes in Cartesian space and Joint space, which are typically determined by specific task requirements, characteristics of the manipulator arm, and tracking accuracy, among other factors. For instance, trajectory planning in Cartesian space enables direct control of the position and orientation of the manipulator end effector, which is relatively more intuitive than that in Joint space, and can more effectively plan smooth motion trajectories [5–7]. From the perspective of advanced computational methods, Xiao *et al*. [8] successfully mitigated the shock and vibration experienced by a robot during its movement from the starting position to the target position in Cartesian space, thereby ensuring the accuracy and safety of point-to-point movements. In contrast, Zhang *et al*. [9] successfully planned two sets of trajectories using polynomial interpolation in Joint space and verified the model by using SimMechanics. However, owing to the insufficient robustness of the PID controller against external environmental factors, the simplification of the model significantly affected the accuracy of the experiment. Additionally, the improved quantum particle swarm optimization algorithm proposed by Luo *et al*. [10] performed outstandingly in the trajectory planning of the robot joint space, effectively enhancing the efficiency and accuracy of trajectory planning. Regarding the S-shaped speed curve, Wu *et al*. [11] investigated how to plan trajectories in Joint space of industrial robots that satisfy kinematic constraints and have continuous jerk. Nevertheless, trajectory planning in Joint space requires complex inverse kinematic solutions. Owing to the presence of joint coupling, the trajectory converted to Cartesian space might be distorted or deviate from the expected path. This constitutes a potential issue for manipulator arms employed in wafer-handling scenarios.

Trajectory planning curves are vital for motion control strategies. The common types include linear interpolation, B-spline curves, polynomial curves, T-shaped speed curves, and S-shaped speed curves. Polynomial interpolation is frequently employed for this purpose. By selecting the appropriate polynomial order, its adaptability can be flexibly adjusted to match different numbers of data points and smooth the curve, thus effectively reducing the vibration and shock of the robotic arm during movement [12–14]. Although high-order polynomial interpolation can provide a smoother curve fit, it may lead to the Runge phenomenon with an excessive number of data points. In addition, they are highly sensitive to minor data point variations, potentially leading to numerical instability. Furthermore, Cong *et al.* [15] proposed the use of fifth-order Non-uniform B-spline curves in the trajectory planning and crafted a multi-objective trajectory optimization model employing the Non-dominated Sorting Genetic Algorithm-II (NSGA2). Regarding the T-shaped speed curve, Ayazbay *et al*. [16] applied a mini-mum time-impact planning algorithm to optimize the curve, targeting an enhanced dynamic performance and reduced mechanical wear. However, this curve lacked smoothness at the endpoints, necessitating additional processing to ensure the overall continuity and smoothness of the trajectory. In response to these limitations, researchers have proposed S-shaped speed curves [17,18] as trajectory planning curves, which are distinguished by smooth transitions at the velocity inflection points. This results in less impact force and noise compared to the T-shaped acceleration and deceleration phases. S-shaped speed curves are commonly used in applications requiring precise control and minimization of mechanical vibration.

Heuristic algorithms are increasingly employed in trajectory planning and typically address complex optimization problems when it is difficult to find an exact solution or when the cost of computing an exact solution is too high. Intelligent algorithms, such as genetic algorithms (GA), particle swarm optimization (PSO) algorithms, annealing algorithms (SA), whale optimization algorithms (WOA), and grey wolf optimization (GWO) algorithms provide novel solutions for addressing intricate trajectory planning problems. They have the characteristics of flexibility, robustness, and diversity, which can help users find the global optimal solutions. Among them, the PSO algorithm [19], which simulates the social behavior of bird flocks to achieve optimization, is widely used to optimize the use of robotic arms owing to its simplicity, ease of implementation, and strong ability for multi-modal optimization. However, the PSO algorithm cannot avoid problems of parameter sensitivity and convergence acceleration. Therefore, research teams have improved to varying degrees to overcome these shortcomings [20–23]. Lu and Wang [24] built an optimal trajectory planning method based on a B-spline and a WOA to improve the work efficiency of a robotic arm. However, they did not fully compare the performance differences between the WOA and other existing algorithms. By combining the back propagation neural network (BPNN) with the GA, the overall performance of the upper limb rehabilitation robotic arm trajectory planning was improved to some extent. However, the parameter adjustment after the combination was very complex and required a large number of experiments and professional knowledge for correction [25]. The SO algorithm excels outstanding in the realm of optimization owing to its exceptional flexibility and robustness. It is not only easy to integrate chaos theory and learning mechanisms to bolster search efficiency but also to maintain efficient exploration ability under poor initial conditions. In addition, it can rapidly adapt to the changes in dynamic environments and optimize search strategies in real time, thereby safeguarding the continuity and effectiveness of the optimization process. Li *et al*. used a SO algorithm to find optimal paths in dynamic environments. Although its performance has been improved compared with other intelligent algorithms, the improved algorithm still requires careful parameter tuning [26]. To address the shortcomings of the SO algorithm, recent literatures [27,28] begun to focus on solving its core mechanism to raise the overall performance of optimization problems. The ISO has also been gradually applied to path planning and to solve general constrained optimization problems.

Therefore, considering the shortcomings of the afore-mentioned studies and the main factors that should be prioritized during the actual operation of the wafer manipulator, this study proposes an improved control strategy for a manipulator that combines the single-segment start-stop S-shaped speed of the circular arc transition, an ISO algorithm, and a time-impact optimization method of trajectory planning. It is worth noting that it has more advantages in terms of the stability of the control effect or the convergence ability of the algorithm compared with other types of methods. And it can adapt to parameter optimization in different scenarios. In summary, the contributions of this study are as follows.

(1) In the area of intelligent optimization methods, the following improvements have been made to the SO algorithm.

- Tent Chaotic Map initialize populations, instead of randomly generating populations with ' *rand* ' functions.
- Dynamic learning factors, integrated with random perturbations, replace static learning factors.
- A cosine annealing learning rate scheme is employed for the adaptive updating of individual positions.
- The Bayesian method is utilized to optimize thresholds, surpassing the reliance on empirical thresholds.

(2) In the area of trajectory planning, the following improvements have been made to the S-shaped speed curve.

- The interpolated arc transition supplants the abrupt right-angle transition
- The continuous single-segment start-stop mechanism supersedes the segmented start-stop mechanism.

2. Materials and Methods

2.1. The Structure Parameters of the Wafer-Handling Robotic Arms

Assuming that the task requirements can be realized in a plane, the system model is simplified when the robotic arm transports the wafer, as shown in Figure 1. Before planning the trajectory, it is necessary to establish the link coordinate system correctly. In general, the points along the Z_i axis are aligned with the i-th joint axis. Furthermore, the points along the X_i axis are aligned with a common vertical line. In addition, the Y_i axis is oriented according to the right-hand rule. In this study, the parameters of the linkage rods are expressed using the improved Denavit-Hartenberg (DH) method. The parameters of each joint and link are obtained using the link coordinate system as a reference, as shown in Table 1.

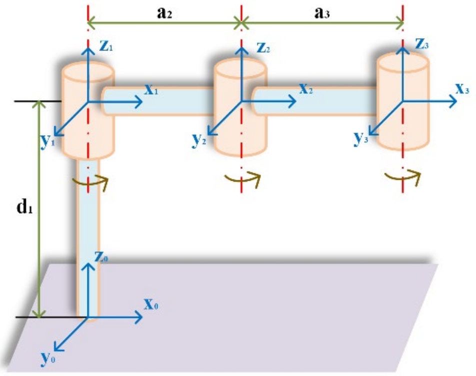


Figure 1. Simplified model of the wafer handling robotic arm.

Table 1. Linkage parameters of the 3R robotic arm.

Link$_i$	a_i /mm	α_i /(°)	d_i / mm	θ_i (/°)	Range of θ_i / (°)
1	0	0	400	0	-120 - 120
2	0	238.4	0	0	-120 - 120
3	0	238.4	0	0	-120 - 120

To facilitate the visual analysis of the subsequent trajectory planning and verify the correctness of the simplified model, this study utilizes the MATLAB robot toolbox developed by Peter Corke. It is a powerful robotic simulation and analysis tool that provides a range of capabilities for modeling, simulating, controlling, and visualizing robots. A robotic arm consistent with the simplified model is successfully constructed on the interface by following the building instructions in the MATLAB robot toolbox, as shown in Figure 2.

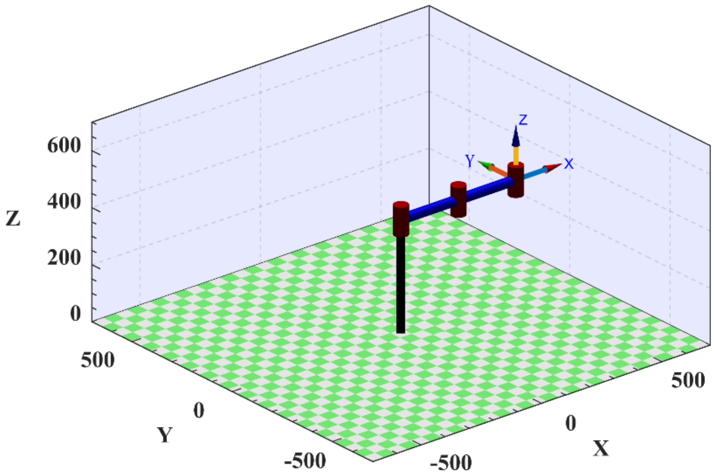


Figure 2. Wafer handling robotic arm in MATLAB.

2.2. *The Mathematical Model of Multiple Objective Function*

In recent years, the optimization of time, energy, impact, and other indicators has been shown to improve the operational efficiency and motion performance of robotic arms in the field of robot trajectory planning [29–32]. Considering the production background and optimization effect, a multi-segment point-to-point trajectory in Cartesian space is adopted to achieve multi-objective optimization of time and impact. First, according to the expected trajectory of the robotic arm, it must be discretized into several interpolation points in Cartesian space. Subsequently, several discrete points are connected using the trajectory function. The joint angle corresponding to the entire trajectory in the joint space can be calculated using the inverse kinematics equation of the robot. Finally, only the selected joint angle conforming to the motion trajectory is substituted into the forward kinematics of the robot to solve the cartesian coordinate value of the expected trajectory. It is assumed that the time at which the manipulator reaches each interpolation point is t_i. The time interval between adjacent interpolation points is $h_i = t_{i+1} - t_i$, where i is the index of the interpolation points ($i = 0,1 \ldots$n). At the same time, through the preliminary analysis, it is clearly observed that there is an inverse correlation between time and impact. When the time of the robotic arm increases, the impact decreases. Therefore, it is impossible to simultaneously obtain the minimum values. Thus, considering the constraints of the geometric joints of the robotic arm, velocity, acceleration, and jerk, the primary goal of this study is to achieve an optimal balance between time and impact. Therefore, a Pareto optimal method is adopted in this study to achieve the tradeoff between optimization objectives. The time function is given by Equation (1). Because the impact quantity can not be directly measured, it is quantified by the acceleration, as shown in Equation (2). The constraints are expressed in Equations (3). The multi-objective optimization function is shown in Equation (4).

$$\min F_1 = \sum_{i=0}^{n} h_i \tag{1}$$

$$\min F_2 = \sum_{i=0}^{n} \sqrt{\frac{\int_0^{t_i} j_i^2 dt}{t_i}} \tag{2}$$

$$\begin{cases} |\theta_i| \le \theta_{\max} \\ |v_i| \le v_{\max} \\ |a_i| \le a_{\max} \\ |j_i| \le j_{\max} \end{cases} \tag{3}$$

$$\min F = w_1 \cdot |F_1 - F_{1\min}| + w_2 \cdot |F_2 - F_{2\min}| \tag{4}$$

where F_1, F_{1min}, and w_1 are the objective function, minimum value, and weight value of time, respectively. F_2, F_{2min}, and w_2 are the objective function, minimum value, and weight of impact, respectively. h_i is the time interval between the i-th and i+1-th interpolation point. t_i and j_i denote the time node and jerk value of the i-th interpolation point. θ_{max}, v_{max}, a_{max}, and j_{max} are the maximum rotational angle, speed, acceleration, and jerk of the joint of the manipulator arm, respectively.

2.3. The Construction of Trajectory Planning

2.3.1. The Explanation of S-Shaped Speed Curve

The T-shaped speed curve is a widely used method for robotic arm control. Because of its simplicity and ease of control, it is extremely common in industrial applications. As shown in Figure 3, this curve may produce large impacts and vibrations at points A, B, C, and D, potentially causing the wafer to break. By contrast, the S-shaped speed curve uses nonlinear adjustments to increase and decrease the velocity, making the motion process smoother and effectively solving the discontinuity problem of the acceleration of the T-shaped speed curve. Therefore, the S-shaped speed curve is more suitable in scenarios requiring higher operational stability.

The implementation complexity of the S-shaped speed curve is higher than the T-shaped speed curve. However, it effectively reduces impact and improves the quality of motion. The S-shaped speed curve divides the entire motion process into seven stages: accelerating acceleration phase, uniform acceleration phase, decelerating acceleration phase, uniform speed phase, accelerating deceleration phase, uniform deceleration phase, and decelerating deceleration phase.

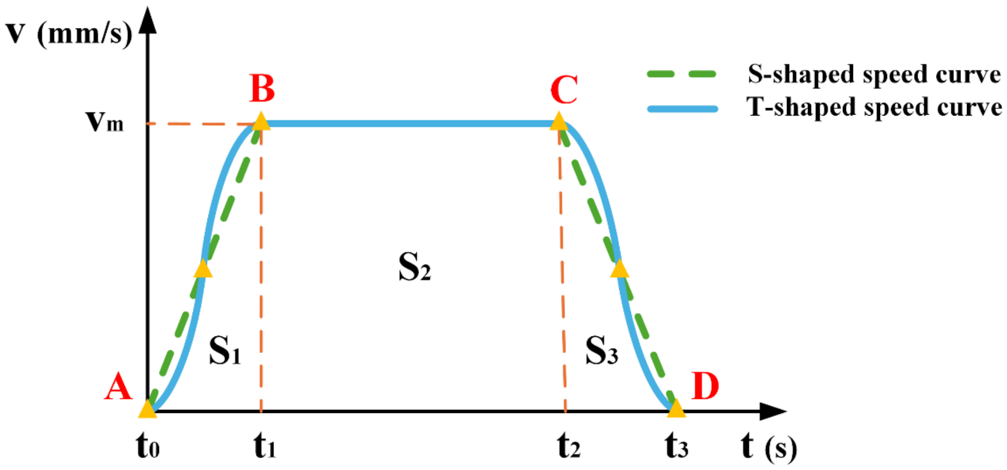


Figure 3. Correlation chart between T-shaped and S-shaped speed curve.

In this study, the kinematic characteristics of an S-shaped speed curve at different stages are described in detail. Specifically, the position, velocity, acceleration, and jerk functions of each stage are shown by Equations (5), (6), (7), and (8), respectively.

$$s(t)=\begin{cases} v_s\tau_1+\frac{1}{6}j_{\max}\tau_1^3 & 0\le t\le t_1 \\ s_1+v_1\tau_2+\frac{1}{2}a_{\max}\tau_2^2 & t_1\le t\le t_2 \\ s_2+v_2\tau_3+\frac{1}{2}a_{\max}\tau_3^2-\frac{1}{6}j_{\max}\tau_3^3 & t_2\le t\le t_3 \\ s_3+v_3\tau_4 & t_3\le t\le t_4 \\ s_4+v_4\tau_5-\frac{1}{6}j_{\max}\tau_5^3 & t_4\le t\le t_5 \\ s_5+v_5\tau_6-\frac{1}{2}a_{\max}\tau_6^2 & t_5\le t\le t_6 \\ s_6+v_6\tau_7-\frac{1}{2}a_{\max}\tau_7^2+\frac{1}{6}j_{\max}\tau_7^3 & t_6\le t\le t_7 \end{cases} \tag{5}$$

$$v(t)=\begin{cases} v_s+\frac{1}{2}j_{\max}\tau_1^2 & 0\le t\le t_1 \\ v_1+a_{\max}\tau_2 & t_1\le t\le t_2 \\ v_2+a_{\max}\tau_3-\frac{1}{2}j_{\max}\tau_3^2 & t_2\le t\le t_3 \\ v_3 & t_3\le t\le t_4 \\ v_4-\frac{1}{2}j_{\max}\tau_5^2 & t_4\le t\le t_5 \\ v_5-a_{\max}\tau_6 & t_5\le t\le t_6 \\ v_6-a_{\max}\tau_7+\frac{1}{2}j_{\max}\tau_7^2 & t_6\le t\le t_7 \end{cases} \tag{6}$$

$$a(t)=\begin{cases} j_{\max}\tau_1 & 0\le t\le t_1 \\ a_{\max} & t_1\le t\le t_2 \\ a_{\max}-j_{\max}\tau_3 & t_2\le t\le t_3 \\ 0 & t_3\le t\le t_4 \\ -j_{\max}\tau_5 & t_4\le t\le t_5 \\ -a_{\max} & t_5\le t\le t_6 \\ -a_{\max}+j_{\max}\tau_7 & t_6\le t\le t_7 \end{cases} \tag{7}$$

$$j(t)=\begin{cases} j_{\max} & 0\le t\le t_1 \\ 0 & t_1\le t\le t_2 \\ -j_{\max} & t_2\le t\le t_3 \\ 0 & t_3\le t\le t_4 \\ -j_{\max} & t_4\le t\le t_5 \\ 0 & t_5\le t\le t_6 \\ j_{\max} & t_6\le t\le t_7 \end{cases} \tag{8}$$

where t_k (k = 0, 1...7) represents the transition point time of part k, which indicates the specific point in time at which each stage begins or ends. τ_k (k = 0, 1...7) represents the local time period of part k, which refers to the difference from the starting point of the current part, called time zero, to the time point of the current part. The equation used $\tau_k = t_k - t_{k-1}$ (k = 1, 2...7). s_k (k = 1, 2...6) represents the last shift of part k. v_k (k = 1, 2...6) represents the final speed of part k and v_s represents the initial speed. v_{max} is maximum speed. a_{max} is the maximum acceleration. j_{max} is the maximum acceleration.

2.3.2. The Improvement of S-Shaped Speed Curve

Trajectory planning is a crucial step in the robotic motion control. When the wafer-handling robotic arm is in operation, it frequently encounters turning points where the direction of movement requires alteration. These turning points often serve as a source of abrupt jerk changes. To alleviate this problem, an S-shaped speed curve adopts a circular arc transition strategy to guarantee a smooth trajectory transition. This strategy not only mitigates the impact and vibration induced by the robotic arm but also contributes to shortening the execution time of the motion trajectory. Prior to constructing the S-shaped speed curve, it is imperative to determine the key interpolation points of the trajectory in Cartesian space, as indicated in Table 2. Simultaneously, the principal motion parameters must be set as presented in Table 3. $v\_\ p_0$, $v\_\ p_1$, and $v\_\ p_2$ are the velocities of the initial point, the circular arc transition section, and the terminal point, respectively. v_{max}, a_{max}, and j_{max} denote the maximum velocity, the maximum acceleration, and the maximum rate of change in the acceleration of an end effector in Cartesian space, respectively. r is the radius of the circular arc transition at the inflection point.

Table 2. Segmentation points in Cartesian space.

Starting point	Intermediate point	Ending point
(400, 0, 400)	(200, 0, 400)	(200, 200, 400)

Table 3. The main motion parameters of S-shaped speed curve.

$v\_\ p_0$ (mm/s)	$v\_\ p_1$ (mm/s)	$v\_\ p_2$ (mm/s)	v_{max} (mm/s)	a_{max} (mm/s^2)	j_{max} (mm/s^3)	r (mm)
0	100	0	100	100	100	30

The trajectory curve shown in Figure 4. can be generated by applying the parameters listed in Table 3. to the S-shaped speed curve model. In this figure, the black lines represent the direct path without a circular transition, whereas the blue lines show a smooth path with a circular transition. It is noteworthy that the radius of the circular transition can be adjusted according to specific work requirements. Overall, the circular transition not only improves the smoothness and efficiency of the robotic arm's motion but also helps to protect the robotic arm and its payload, while enhancing the overall operational performance. This transition method allows the robotic arm to perform complex tasks more accurately and reliably.

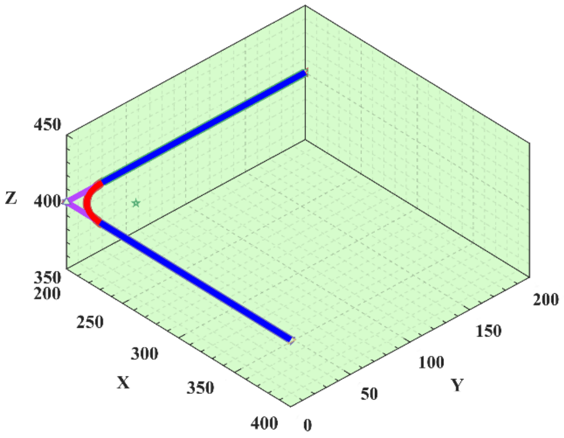


Figure 4. Contrast diagram of arc transition and right-angle path.

By analyzing the data in Table 2, it is clear that this trajectory planning task involves three key nodes that the robotic arm must reach in sequence. S-shaped speed curve planning is a standard point-to-point start-stop motion scheme. Specifically, this task is represented by two movements from the starting point to the middle point and then from the middle point to the end point, with zero starting and ending speeds for each movement. Based on the data in Table 3, the velocity image of the scheme is shown in Figure 5. However, this scheme causes the manipulator to frequently start and stop, which not only increases the vibration during the motion process but also leads to time wastage in the acceleration and deceleration stages. To solve this problem, this study has proposed an improved scheme for the single-segment start-stop mechanism, optimizing the starting and stopping parts of the intermediate process into a smooth motion mode with constant speeds at both ends, as shown in Figure 6. When the velocity of the first segment of motion attains the maximum value of the set parameters, instead of reducing the velocity back to zero as in the standard scheme, the second segment of motion proceeds in a uniform transition manner. The specific parameters still refer to Table 3. By using this improved scheme, the problems existing in the conventional scheme can be effectively reduced. Simultaneously, the movement efficiency can be improved and the vibration can be reduced.

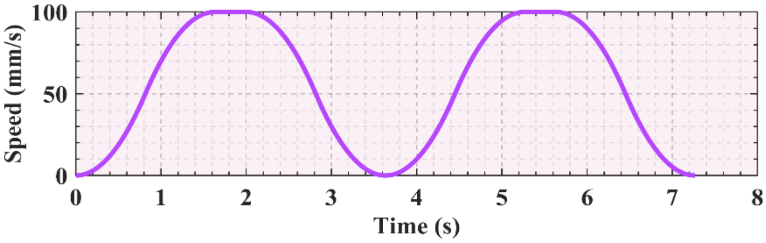


Figure 5. Continuous start-stop S-shaped speed curve.

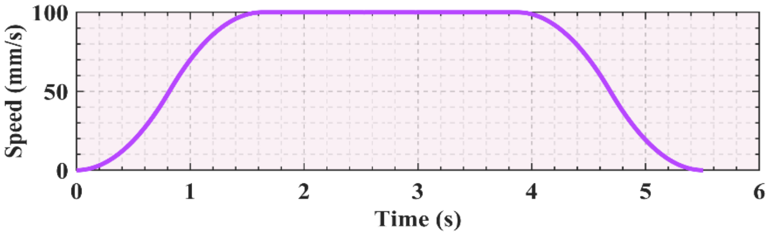


Figure 6. Improved start-stop S-shaped speed curve.

Comparing Figure 5. and 6, it can be seen that the improved motion scheme results in a smoother speed curve. Concurrently, it effectively reduces the vibration generated by the motor during starting and stopping. With the basic parameters unchanged, the unoptimized time schedule which does not include arc adjustment and smooth transition took 7.26 s, whereas the optimized time schedule, which includes arc adjustment and smooth transition, is shortened to 5.51 s. After the two parts are optimized, the total time is saved by 1.75 s, a 24.1% reduction compared to the original scheme.

2.4. Improved Snake Optimization Algorithm

2.4.1. Initialization Population of Tent Chaotic Map

The snake optimization (SO) algorithm is an intelligent optimization technique that simulates the specific mating behavior of snakes. It achieves an effective balance between global and local searches by using a unique search mechanism. In the initialization phase of the algorithm, individuals in the population are equally divided into male and female individuals. The composition of this initial population can be represented by Equation (9), where the numbers of male and female individuals are represented by Equations (10) and (11), respectively.

$$X_i = lb + rand \times (ub - lb) \tag{9}$$

$$N_m = \frac{N}{2} \tag{10}$$

$$N_f = N - N_m \tag{11}$$

where X_i denotes the position of the i-th individual. The random function ‘ *rand* ’ generates a real random number between 0 and 1. The upper and lower bounds of X_i are *ub* and *lb*, respectively. *N* is the total population size. N_m and N_f are the numbers of individual males and females, respectively.

However, the use of the function ‘ *rand* ’ in Equation (9) of the SO algorithm may result in an uneven distribution of random numbers. Moreover, because this function is based on a deterministic algorithm, the same random sequence is generated when the same seed is given. This situation may lead to the premature convergence of the heuristic algorithm to the local optimal solution, making it challenging to find the global optimal solution. To solve this problem in population initialization, this study introduces a Tent Chaotic Map. It has excellent chaotic characteristics, which can effectively improve the uniformity of population distribution. In addition, compared with other Chaotic Maps, its implementation is more intuitive and simpler. The Tent Chaos Map can be expressed as Equation (12). And ‘ x_i ’ replaces ‘ *rand* ’ as shown in Equation (13).

$$x_{i+1} = \begin{cases} \frac{x_i}{a}, & 0 \le x_i < a \\ \frac{1-x_i}{1-a}, & a \le x_i \le 1 \end{cases} \tag{12}$$

$$X_i = lb + x_i \times (ub - lb) \tag{13}$$

where x_i and x_{i+1} stand for the i-th and i+1-th values of Tent Chaotic Map, respectively. X_i stands for the i-th individual position. Moreover, parameter *a* is the variable value, which is 0.5 in this study [33].

2.4.2. Dynamic Learning Factor with Random Noise

In heuristic algorithms, learning factors play a key role in regulating the search behavior, including the balance between exploration and exploitation. Exploration refers to the ability to search for new solutions randomly, whereas development refers to the ability to search locally for the best solution. The learning factors of the SO algorithm include the constant value C_1 of the food calculation equation, constant value C_2 of the food exploration stage, and constant value C_3 of the food development stage. The initial values of these constants are typically set to C_1 = 0.5, C_2 = 0.05, and C_3 = 2. The equations containing C_1, C_2, and C_3 are Equations (14), (15), (16), (17), and (18). The subscript ‘m’ and ‘f’ refers to the male and female snakes, respectively. The special mention of female snake expressions will be omitted in the following text.

$$Q = C_1 \times (\frac{t-T}{T}) \tag{14}$$

$$X_{i,m\_1}(t+1) = X_{rand,m\_1}(t) \pm C_2 \times A_m \times \left[(ub-lb) \times rand + lb\right] \tag{15}$$

$$X_{i,m\_2}(t+1) = X_{\text{food}}(t) \pm C_3 \times \text{Temp} \times rand \times \left[X_{\text{food}}(t) - X_{i,m}(t)\right] \tag{16}$$

$$X_{i,m\_3}(t+1) = X_{i,m\_3}(t) \pm C_3 \times F_m \times rand \times \left[Q \times X_{best,f} - X_{i,m}(t)\right] \tag{17}$$

$$X_{i,m\_4}(t+1) = X_{i,m\_4}(t) \pm C_3 \times M_m \times rand \times \left[Q \times X_{i,f}(t) - X_{i,m}(t)\right] \tag{18}$$

where *Q* is the amount of food consumed. *t* denotes the current number of iterations. *T* denotes the maximum number of iterations. *Temp* is the iteration temperature. X_{i,m_1}, X_{i,m_2}, X_{i,m_3}, and X_{i,m_4} are the locations of food in the exploration phase of male snakes, food in the development phase of male snakes, combat mode in the development phase of male snakes, and mating mode in the development phase of male snakes, respectively. X_{rand,m_1} is the randomly selected male population.

X_{best_f} is the best position for the females in this population. A_m, F_m, and M_m are measured for the male's ability to find food, the male's ability to fight, and the male's ability to mate.

Because the learning factors C_1, C_2, and C_3 remain unchanged during the iteration, the algorithm may not adapt to the dynamic changes in the solution space. Hence, the delta value is introduced for the normal distribution of random tubs. This perturbation helps the SO algorithm eliminate the trap of local optimal solutions, thus enhancing its global search abilities. By dynamically adjusting these fixed learning factors C_1, C_2, and C_3, the SO algorithm can achieve an effective dynamic balance between global and local searches, thereby improving the performance and efficiency of the algorithm. The concrete expression of the random disturbance is shown in Equations (19) and (20). In Equation (19), *alpha* is used to store the parameter values for learning factors C_1, C_2, and C_3.

$$delta = sigma \times randn \tag{19}$$

$$alpha(t+1) = alpha(t) + delta \tag{20}$$

where *sigma* represents the amplitude of the perturbation. It is 0.1 in this paper [34]. *alpha* is limited to a range between 0.01 and 1. Random function ' *randn* ' represents a random value with a standard normal distribution.

2.4.3. Cosine Annealing Learning Rate

In Equations (15), (16), (17), and (18), the strategy for updating the individual positions is relatively simple. Without considering adjusting the strategy as an iteration progress, the SO algorithm mainly relies on the current individual's position and velocity vector to determine the next individual's position. This method is more likely to cause the algorithm to become stuck in a locally optimal solution. To improve the update strategy of the original SO algorithm, this study introduced a cosine learning annealing rate. This strategy simulates the dynamic adjustment of the learning rate based on the cosine function, which can dynamically adjust the speed of the position updating based on the number of iterations. In the early stages of the iteration, the learning rate can be reduced rapidly to achieve fast convergence. In the later stages of the iteration, fine adjustments are made to optimize the precision of the solution by reducing the learning rate, as shown in Figure 7. The specific expression of the cosine learning annealing rate is shown in Equation (21).

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \times (1 + \cos(\frac{T_{cur}}{T_{\max}}\pi)) \tag{21}$$

where η_t is the current learning rate. η_{min} and η_{max} are the minimum and maximum values of the learning rate, which are 0.001 and 0.01 in this paper. T_{cur} and T_{max} are the current and total iterations, respectively. T_{max} is usually set to half the number of iterations of the main algorithm, which is 20 in this article [35].

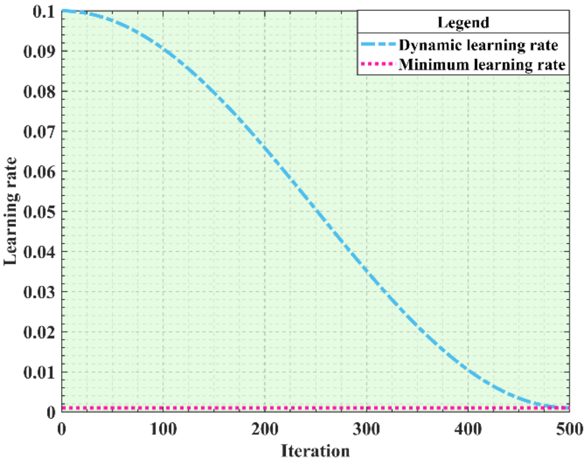


Figure 7. Cosine annealing learning rate iteration graph.

The cosine annealing learning rate exhibits periodic change, which enables the SO algorithm to adopt diverse search strategies in different stages, thus enhancing the dynamic adjustment ability of the algorithm. This flexibility helps the algorithm to avoid premature convergence during the search process, thereby increasing the possibility of finding a global optimal solution. Therefore, by introducing the cosine annealing learning rate based on Equations (15), (16), (17), and (18), the improved Equations (22), (23), (24), and (25) are obtained.

$$X_{i,\,m\_1}(t+1)=\eta_t\times X_{rand,\,m\_1}(t)\pm\eta_t\times C_2\times A_m\times\left[(ub-lb)\times rand+lb\right] \tag{22}$$

$$X_{i,\,m\_2}(t+1)=\eta_t\times X_{\text{food}}(t)\pm\eta_t\times C_3\times \text{Temp}\times rand\times\left[X_{\text{food}}(t)-X_{i,\,m}(t)\right] \tag{23}$$

$$X_{i,\,m\_3}(t+1)=\eta_t\times X_{i,\,m\_3}(t)\pm\eta_t\times C_3\times F_m\times rand\times\left[Q\times X_{best,f}-X_{i,\,m}(t)\right] \tag{24}$$

$$X_{i,\,m\_3}(t+1)=\eta_t\times X_{i,\,m\_3}(t)\pm\eta_t\times C_3\times F_m\times rand\times\left[Q\times X_{best,f}-X_{i,\,m}(t)\right] \tag{25}$$

2.4.4. Cosine Annealing Learning Rate

In the process of determining threshold values of the SO algorithm, parameter setting often depends on the experience and intuition of algorithm developers or needs to be determined by practical users through practice. This approach lacks universality and requires considerable time for experimentation and adjustments. If the parameters are not selected properly, the adaptability of the algorithm may be affected. To solve the problem in which the SO algorithm has difficulty adapting parameters in the iterative process, this study introduces a hyperparameter tuning method commonly used in machine learning, which is known as Bayesian optimization. The Gaussian process mainly includes Gaussian process regression and an acquisition function.

First, it establishes prior knowledge of the global behavior of the objective function through a Gaussian process. The mean vector and covariance matrix involved in the prior distribution are shown in Equations (26) and (27). Subsequently, using the observed objective function to obtain values at different input points, this prior knowledge is updated to form a posterior distribution. The posterior mean and variance are given by Equations (28) and (29). Then, the improved collection function is expected to consider the current best observations and the posterior distribution of the Gaussian process to determine the next evaluation point. The collection function used in this study is expected to improve equation shown in (30). In short, this method of guiding the search based on existing knowledge can avoid many useless attempts.

$$\mu_0=\left[\mu_0(x_1),\ldots,\mu_0(x_k)\right]^T \tag{26}$$

$$\sum\nolimits_0=\begin{bmatrix}\sum_0(x_1,x_1) & \cdots & \sum_0(x_1,x_k)\\ \vdots & \ddots & \vdots\\ \sum_0(x_k,x_1) & \cdots & \sum_0(x_k,x_k)\end{bmatrix} \tag{27}$$

We assume that the above expression follows a multi-variate normal distribution, where μ0(x) represents the mean function. $\sum_0$(x,x') represents the kernel function, also known as the covariance function, which describes the similarity between points x and x' in the input space.

$$\mu_{\text{n}}(x)=\mu_0(x)+\sum\nolimits_0(x,x_1:n)\times(K_{x_1:n,x_1:n}+\sigma^2 I)^{-1}\times(f(x_1:n)-\mu_0(x_1:n)) \tag{28}$$

$$\sigma^2_{\text{n}}(x)=\sum\nolimits_0(x,x)-\sum\nolimits_0(x,x_1:n)\times(K_{x_1:n,x_1:n}+\sigma^2 I)^{-1}\times\sum\nolimits_0(x_1,n:x) \tag{29}$$

$$EI(x)=\mathbb{E}\left[\max\left(f(x)-f^*,0\right)\right] \tag{30}$$

where $K_{x1:n,x1:n}$ are the kernel matrix. σ^2 is the noise variance. f(x_1:n) is the observed vector. f *= max$_{i = 1, \ldots n}$ is the currently observed best target value. f(x_i) is the function value.

2.4.5. ISO Algorithm Steps

The ISO algorithm is outlined in Algorithm 1, as shown in Table 4.

Table 4. ISO pseudo-code.

Algorithm 1: Improved Snake Optimization Algorithm

```
Initialize parameters
Initialize the population using the Tent Chaotic Map as Eq. (13)
Divide population into $N_m$ and $N_f$ using Eqs. (10) and (11)
while ($T < T_{max}$) do
    Evaluate the fitness and identify the best female and male
    Calculate the iteration temperature and food quantity Eq. (14)
    if (Food < $threshold_1$) then
        Cosine annealing exploration using Eq. (15)
    else if (Food > $threshold_2$) then
        Cosine annealing exploration using Eq. (16)
    else
        if (rand < $threshold_3$) then
            Enter Fight Mode using Eq. (17)
        else
            Enter Mating Mode using Eq. (18)
            Replace the worst-performing female and male
        end if
    end if
end while
Return the optimal solution
```

The flow chart of the ISO algorithm, which illustrates the step-by-step process, is shown in Figure 8.

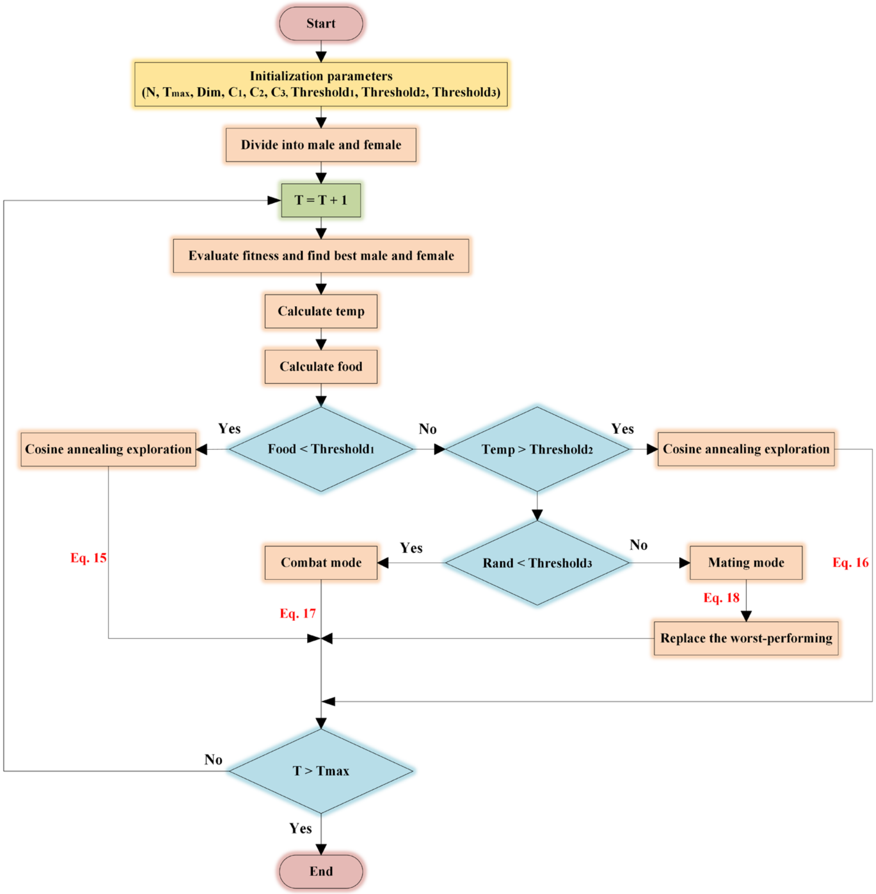

Start
Initialization parameters
(N, T_{max}, Dim, C_1, C_2, C_3, $Threshold_1$, $Threshold_2$, $Threshold_3$)
Divide into male and female
T = T + 1
Evaluate fitness and find best male and female
Calculate temp
Calculate food
Yes
Cosine annealing exploration
Food < $Threshold_1$
No
Temp > $Threshold_2$
Yes
Cosine annealing exploration
Eq. 15
Combat mode
Yes
Rand < $Threshold_3$
No
Mating mode
Eq. 16
Eq. 17
Eq. 18
Replace the worst-performing
No
T > Tmax
Yes
End

Figure 8. The flow chart of the improved snake optimization algorithm.

3. Results

3.1. Ability Test and Analysis for ISO

3.1.1. Parameter Settings and Simulation Results

In this study, all algorithm experiments and simulations are performed using MATLAB 2022B software platform. To verify the convergence performance and accuracy of the ISO algorithm, five heuristic algorithms—SO [36], SA [37], PSO [19], GWO [38], and WOA [39]—are added for comparison. In addition, to ensure the rigor and accuracy of the experimental results, the CEC2005 benchmark test set is selected as the objective function, as listed in Table 5. Five unimodal functions (F1-F5) and three multi-modal functions (F6-F8) are listed in Table 5. Through an in-depth analysis of these benchmark functions, the performance of the ISO algorithm is evaluated to a certain extent.

Before simulating the benchmark test functions for the six heuristic algorithms, a brief introduction to their main parameter settings is necessary to ensure that researchers can obtain similar experimental results when conducting simulations. To ensure the fairness of experiments, the population size (*N*) of all algorithms is set to 50, the maximum iteration number (T_{max}) is set to 100, the dimension (*Dim*) is set to 30, and each algorithm is run 20 times independently. For the other key parameters within the algorithms, recommended values based on experience are used for setting, and the specific parameter values are listed in Table 6. After successfully configuring these parameters, a MATLAB script is used to start the six algorithms simultaneously. The experimental results are summarized in Table 7, and the iteration curves of the benchmark test functions are displayed in Figure 9.

Table 5. Test functions.

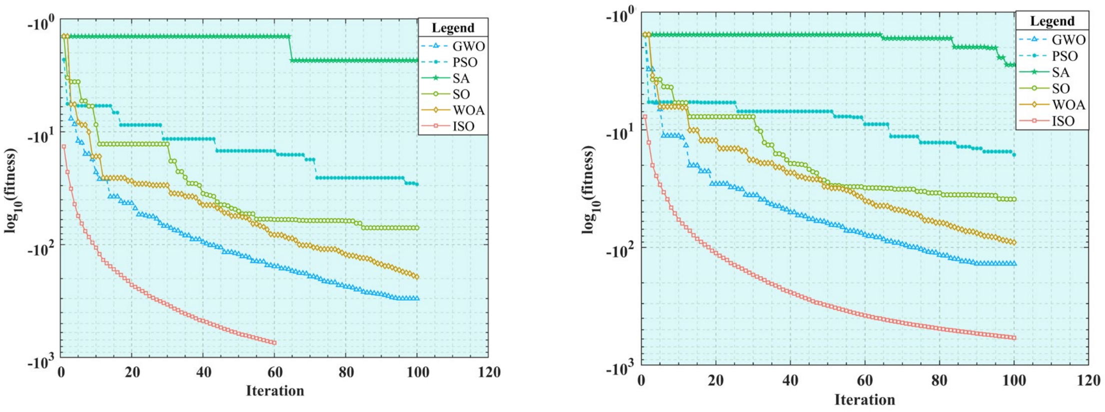
Function	Range	Min
$F_1(x)=\sum_{i=1}^{n}x_i^2$	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$	[-100,100]	0
$F_4(x)=\max_i\left\{\lvert x_i\rvert,1\le i\le n\right\}$	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n}ix_i^4+randm[0,1)$	[-128,128]	0
$F_6(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	[-5.12,5.12]	0
$F_7(x)=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	0
$F_8(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}$ $+\sum_{i=1}^{n}u(x_i,5,100,4),$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & a>x_i>-a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$	[-50,50]	0

Table 6. Parameter settings for the algorithms.

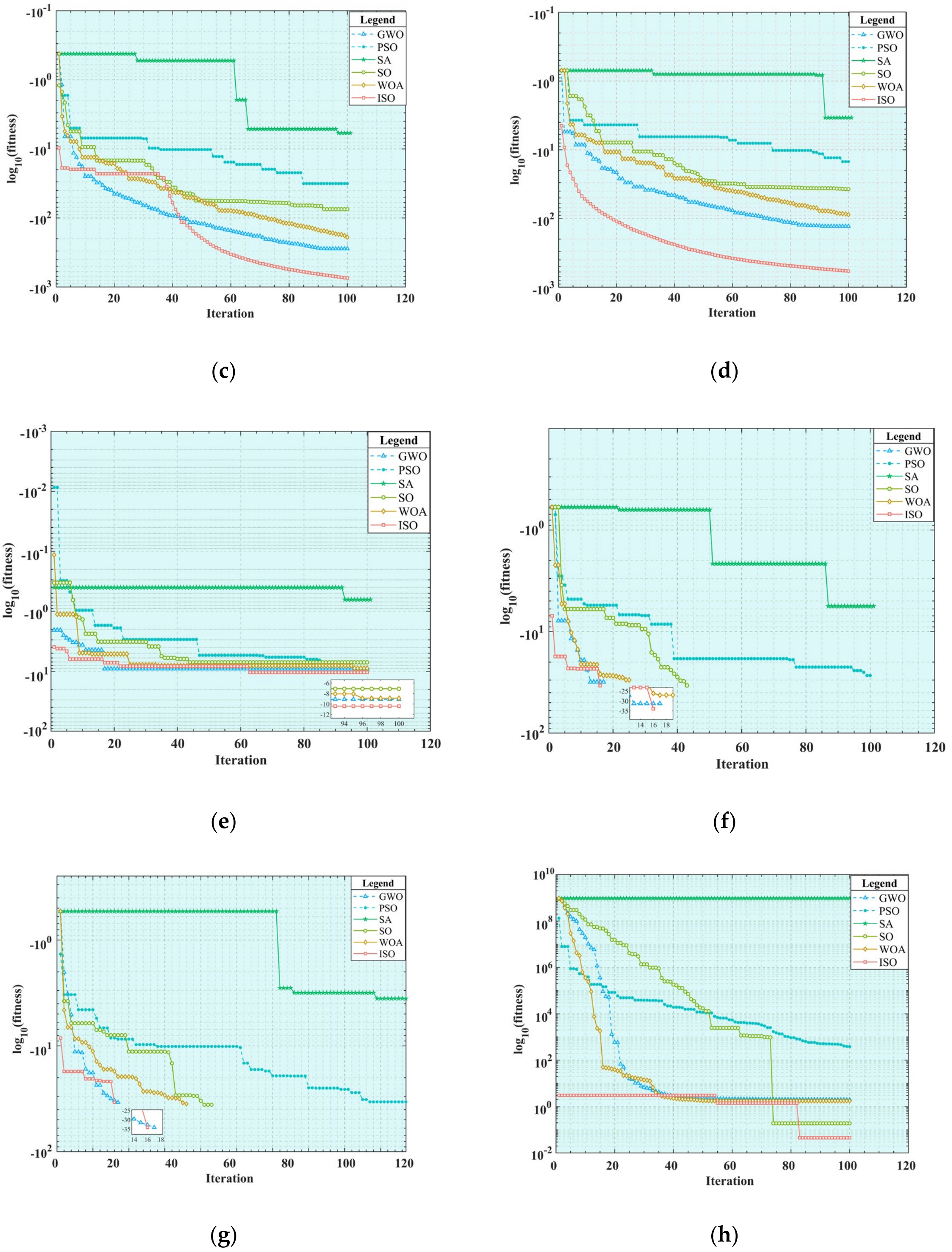
Algorithm	Parameter settings
SO	Theshold$_1$ = 0.25, Theshold$_2$ = 0.6, Theshold$_3$ = 0.6, C$_1$ = 0.5, C$_2$ = 0.05, C$_3$ = 2
SA	Temp = 100, Temp$_{min}$ = 0.01, alpha = 0.95
PSO	w = 0.9, c$_1$ = 2, c$_2$ = 2, V$_{max}$ = 0.3 * ub, V$_{min}$ = 0.3 * lb
GWO	r$_1$ = r$_2$ = rand
WOA	r$_1$ = r$_2$= rand, p = rand, b = 1
ISO	sigma = 0.1, alpha$_{min}$ = 0.01, alpha$_{max}$ = 1.0, eta$_{max}$ = 0.01, eta$_{min}$ = 0.001

Table 7. Results obtained by different algorithms (D = 30).

Function		SO	SA	PSO	GWO	WOA	ISO
F$_1$	MEAN	8.27E-40	7.30E-02	3.04E-12	6.24E-105	8.40E-79	**0**
	STD	2.51E-39	2.27E-01	6.51E-12	2.79E-104	2.10E-78	**0**
F$_2$	MEAN	3.95E-21	4.26E-02	1.38E-07	8.55E-57	2.95E-40	**9.14E-214**
	STD	5.26E-21	2.87E-02	2.22E-07	3.13E-56	7.95E-40	**0**
F$_3$	MEAN	2.16E-39	3.85E-02	6.47E-12	6.66E-107	2.18E-79	**0**
	STD	7.04E-39	8.37E-02	2.16E-11	2.98E-106	6.74E-79	**0**
F$_4$	MEAN	2.01E-20	1.34E-01	1.38E-06	1.27E-53	8.56E-40	**2.27E-213**
	STD	3.54E-20	9.71E-02	2.36E-06	3.93E-53	2.64E-39	**0**
F$_5$	MEAN	7.01E-04	6.64E-01	1.77E-03	5.21E-04	7.33E-04	**1.75E-04**
	STD	7.23E-04	8.91E-01	1.26E-03	5.08E-04	7.05E-04	**2.72E-04**
F$_6$	MEAN	**0**	1.79E-02	2.39E-12	**0**	**0**	**0**
	STD	**0**	3.42E-02	3.35E-12	**0**	**0**	**0**
F$_7$	MEAN	**0**	1.67E-02	6.15E-12	**0**	**0**	**0**
	STD	**0**	3.42E-02	3.35E-12	**0**	**0**	**0**
F$_8$	MEAN	1.74E+01	1.10E+09	4.41E+01	1.24E+00	8.13E-01	**2.06E-01**
	STD	3.32E+01	1.35E+08	2.59E+01	4.26E-01	3.08E-01	**2.03E-01**



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



(**c**)

(**d**)

(**e**)

(**f**)

(**g**)

(**h**)

Figure 9. Convergence curve of the reference function: (**a**) F_1; (**b**) F_2; (**c**) F_3; (**d**) F_4; (**e**) F_5; (**f**) F_6; (**g**) F_7; (**h**) F_8.

3.1.2. Simulation Result Analysis

The results of the eight benchmark functions listed in Table 7 are comprehensively analyzed and compared. As shown in Figure 9(a)-(e), it is clearly observed that when the dimension (*Dim*) is 30, the ISO algorithm achieves an optimal fitness value of 0 for the F_1 and F_3 problems in the unimodal test function. Its convergence curve can rapidly jump out and continue to decline after falling into the local optimum for a short time. This indicates that the ISO algorithm can effectively escape from a local optimal solution and successfully obtain a global optimal solution. By contrast, to a certain extent, the other five algorithms fall into the local optimum. For the F_2 and F_4 problems, although the average value of the ISO algorithm is not zero, the variance is zero, indicating that the algorithm can obtain a global optimal solution to some extent. Its average precision is better than that of the second ranked GWO algorithm. In addition, the convergence speed of the ISO algorithm is faster than those of the other algorithms. For the F_5 problem, the performances of the ISO algorithm and the other five algorithms are similar, but its overall optimization precision is slightly higher, and the convergence curve is better, without falling into the local optimum too early. Overall, the optimization effect of the ISO algorithm on unimodal functions is significantly better than that of the other five algorithms. Further analyzing Figure 9(f)-(h), when the dimension (*Dim*) is 30 in the multi-modal test functions, the SO, GWO, WOA, and ISO algorithms can achieve the global optimal value in the F_6 and F_7 problems with good overall performance, but the ISO algorithm has a faster iteration speed. However, the performance of the six algorithms is not outstanding in the F_8 problem, but the average value and variance of the ISO algorithm are the best, with an optimization precision approximately twice that of the other algorithms. In summary, the ISO algorithm demonstrates higher optimization precision in both unimodal and multi-modal functions than the other algorithms. It also improves the avoidance of falling into the local optimum compared to the SO algorithm, showing the expected optimization effect.

3.1.3. Friedman Test

Further analysis of the results from 20 experimental groups using the Friedman test method reveal significant differences in the overall performance of the six algorithms across eight benchmark functions. It is a flexible and effective tool that does not depend on the specific distribution of data, making it particularly suitable for comparing the algorithm performance. When dimension (*Dim*) is set to 30, the average ranking of the mean and variance of each algorithm is shown in Table 8. A lower average ranking indicates that the algorithm performed better, and the results are more stable. According to the data in Table 8, the ISO algorithm ranks first in terms of the mean value and variance. Compared with the GWO algorithm, which ranked second, the average ranking score of the ISO algorithm's mean and variance are 38.9% higher. Combined with the SO algorithm, the ISO algorithm shows a significant improvement in the average ranking score of the mean and variance, reaching 60.8% and 63.4%, respectively. Considering the overall performance of the algorithm, the ISO algorithm exhibits better performance in terms of stability and convergence accuracy.

Table 8. Friedman test table.

Algorithm	SO	SA	PSO	GWO	WOA	ISO
Mean rank	3.5	6	5	2.25	2.875	1.375
STD rank	3.75	6	4.875	2.25	2.75	1.375

3.2. Time-Jerk Optimal Trajectory Planning Based on ISO

After elaborating on the ISO algorithm and the modified S-shaped speed curve in detail, a combination of the two is performed. The objective function is obtained in this study, as presented in Equation (4). Firstly, Equation (4) adopts a compromise algorithm by adjusting the values of w_1 and w_2 to determine the equilibrium point in the multi-objective optimization. Because the weights of time and impact are considered consistent in this study, both w_1 and w_2 are set to 0.5. Secondly, the time function F_1 and the impact function F_2 belong to different categories of physical quantities. When the weighted sum of their calculation results is taken directly, it leads to an imbalance between the time and impact. To address this issue, a minimum-maximum normalization method is employed in this study. It is typically used to eliminate the influence of dimensions among different features, making the data comparable on the same scale, as shown in Equations (31) and (32).

$$F_{1\_norm} = \frac{F_1 - \min(F_1)}{\max(F_1) - \min(F_1)} \tag{31}$$

$$F_{2\_norm} = \frac{F_2 - \min(F_2)}{\max(F_2) - \min(F_2)} \tag{32}$$

After the above operations, the domains of the values for F_1 and F_2 functions are restricted within the same range. Based on this, a set of excellent solutions is selected by solving a multi-objective function. Then, the optimal solution is identified using the Pareto optimization method. The Pareto judgment standard is as follows. If a new solution improves at least one objective without causing a performance decline in other objectives, then the new solution is considered to be superior to the original solution. When these Pareto optimal solutions are smoothly connected, a Pareto frontier is formed. Ultimately, this study aims to determine a balance point on this curve to achieve the best overall effect.

The specific parameter configuration of the simulation experiment can be found in Table 9 for the application of the ISO algorithm to the S-shaped speed curve optimization problem, and the kinematic parameters are set according to Industrial Robot ISO 9283:1998. In the simulation experiments, a_{max} and j_{max} are set as independent variables and their upper bound (*ub*) and lower bound (*lb*) are between 80 and 120. In the SO algorithm, the empirical values of threshold parameters threshold$_1$, threshold$_2$, and threshold$_3$ are 0.25, 0.6, and 0.6, respectively. To determine more reasonable parameter combinations, the ranges of these thresholds are unified and adjusted. The range of threshold$_1$ is limited to 0.05 to 0.45, while the ranges of threshold$_2$ and threshold$_3$ are adjusted to 0.4 to 0.8. Correspondingly, the search range of the threshold parameter (Bayesian threshold) in the Bayesian optimization process is also set to 0.05 to 0.8 to facilitate determining the optimal parameter configuration. The other parameters have the same meanings as those described earlier. The Bayesian optimization threshold simulation results are shown in Table 10, which includes the iteration, the actual observed objective value of the current iteration, the best observed value among all observations (BestSoFar_observed), the model-predicted best value (BestSoFar_estimated), and the best thresholds combination obtained in this generation. However, because there are no enough data to support the construction of the first-generation model of Bayesian optimization, Objective, BestSoFar_estimated, and BestSoFar_observed are not included in the calculation. In addition, a simulation diagram of the minimum objective value and function calculation times corresponding to Bayesian optimization is shown in Figure 10, where the initial value of the first generation is replaced by number 1. For the dual-objective optimization problem considered in this study, the distribution of the Pareto frontier is generally in the form of a curve, which is intuitively shown in Figure 11.

Table 9. The main parameter setting.

v_ p_0 (mm/s)	v_ p_1 (mm/s)	v_ p_2 (mm/s)	v_{max} (mm/s)	r_1 (mm)	Bayesian evaluations	Bayesian threshold	T_{max}	Dim	N	lb	ub
0	100	0	100	30	10	[0.05,0.8]	100	2	20	[80,120]	[80,120]

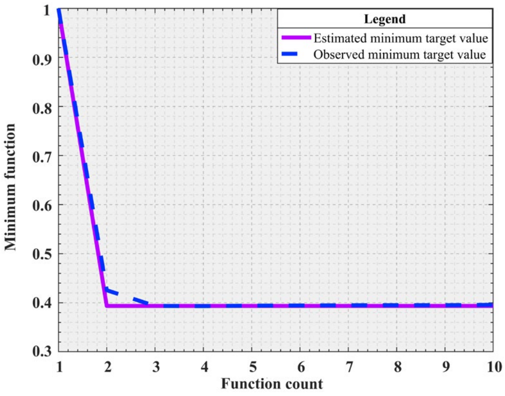


Figure 10. Function count.

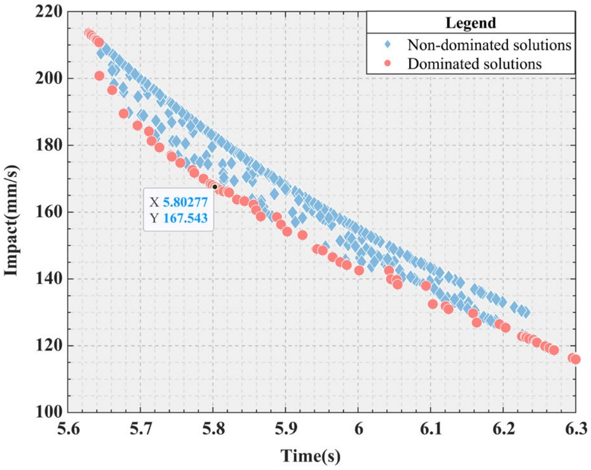


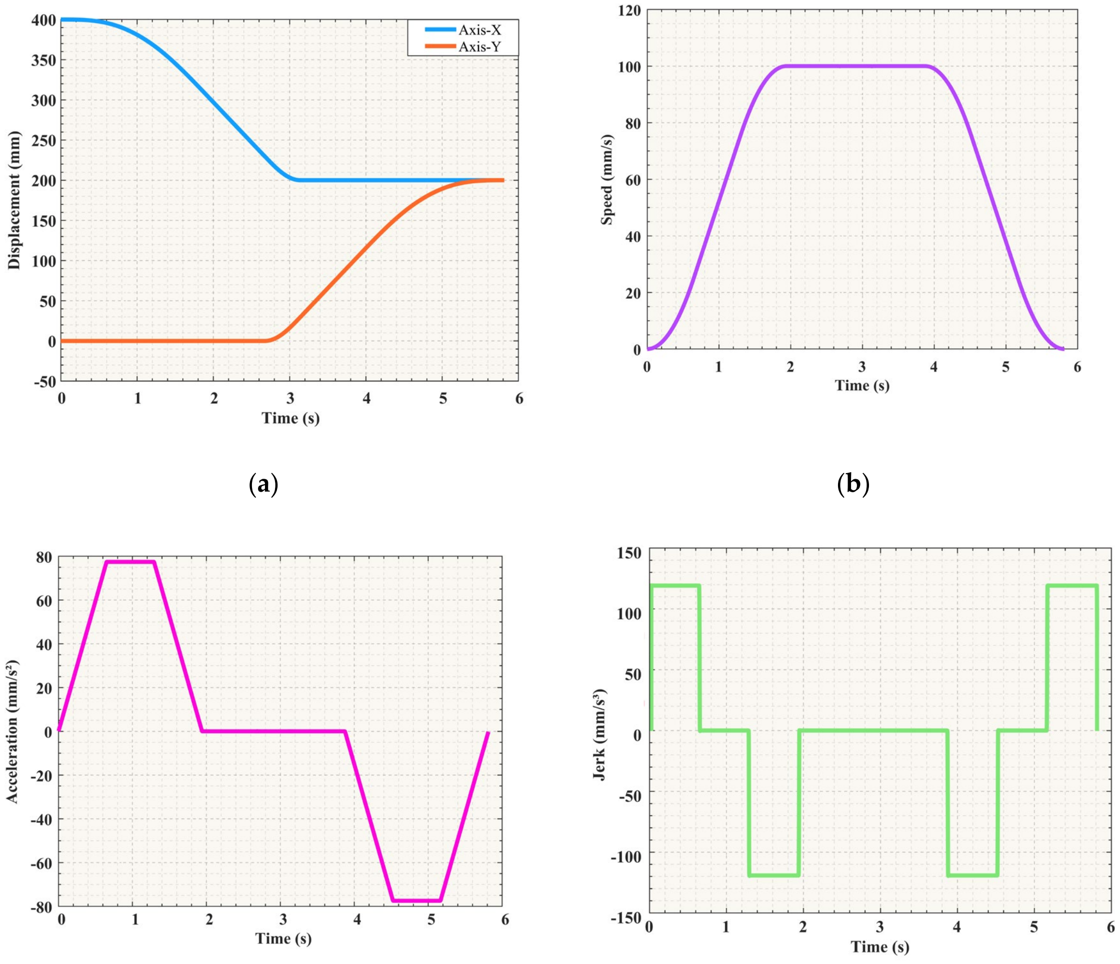
Figure 11. Pareto frontier.

According to the data provided in Table 10, Bayesian optimization is used to select the second-generation simulation results with the minimum objective function value. This selection is based on minimization of the target value during the iteration. Figure 10. clearly shows the dynamic change in the target value with the iteration, making the change process clear. Further analysis of Figure 11, the minimum values of F_1 and F_2 respectively reach 5.6294 s and 115.937 mm/s. And the two minimum values are respectively located at the two ends of the Pareto frontier composed of the red dots. To optimize both the time and impact, an optimal advantage must be found on the Pareto frontier. A specific point is identified in Figure 11, which achieves an optimal balance between the time and impact. When a_{max} and j_{max} are 78.1476 mm/s^2 and 119.8756 mm/s^3, the time and impact of the Pareto optimal value are 5.8028 s and 167.5427 mm/s, respectively.

Using the computed a_{max} and j_{max}, these data are reintegrated into the trajectory planning of the S-shaped speed curve. Through this process, it is possible to accurately determine the partial displacement of the trajectory planning in the directions of the X-axis and Y-axis, as well as the associated combined velocities, combined accelerations, and combined accelerations, as shown in Figure 12. These figures clearly show that the planned S-shaped speed curve exhibited smooth characteristics during movement, effectively avoiding sudden changes and shocks. This is extremely important for precision operations such as wafer handling, as it helps reduce the damage that can be caused by impact.

Table 10. The iterative graph of Bayesian thresholds.

Iteration	Objective	BestSoFar(observed)	BestSoFar(estimated)	threshold$_1$	threshold$_2$	threshold$_3$
1	—	—	—	0.4227	0.7616	0.5225
2	0.3936	0.3936	0.4255	0.0989	0.7764	0.5705
3	0.3952	0.3936	0.3936	0.1721	0.7210	0.7933
4	0.3969	0.3936	0.3932	0.0752	0.4968	0.5437
5	0.3953	0.3936	0.3942	0.1403	0.6057	0.7633
6	0.3991	0.3936	0.3946	0.0500	0.5764	0.5441
7	0.3957	0.3936	0.3949	0.1485	0.5278	0.4003
8	0.3969	0.3936	0.3951	0.1113	0.7425	0.4637
9	0.3955	0.3936	0.3951	0.1592	0.7106	0.7313
10	0.3979	0.3936	0.3958	0.1587	0.6833	0.6341



(a) (b)

(c) (d)

Figure 12. Kinematic image: (**a**) Displacement curve in Cartesian space; (**b**) Speed curve in Cartesian space; (**c**) Acceleration curve in Cartesian space; (**d**) Jerk curve in Cartesian space**.**

In addition, this study meticulously records the key parameters, particularly the joint angles, during the S-shaped speed curve movement. Two sets of joint angles consisting of several interpolation points are substituted into the forward kinematics formula, as shown in Equation (33). X and Y represent the coordinate values in the Cartesian coordinate system, respectively.

$$\begin{cases} X = link_1 \times \cos\theta_1 + link_2 \times \left(\cos\theta_1 + \cos\theta_2\right) \\ Y = link_1 \times \sin\theta_1 + link_2 \times \left(\sin\theta_1 + \sin\theta_2\right) \end{cases} \quad (33)$$

The actual motion trajectory depicted in the planar space is shown in Figure 13. By comparing Figure 4 and 13, the high consistency of the two trajectories can be clearly verified. This comparison clearly validates the accuracy and reliability of the ISO. This planning method not only elevates the accuracy of trajectory planning, but also ensures the safety and reliability of the motion.

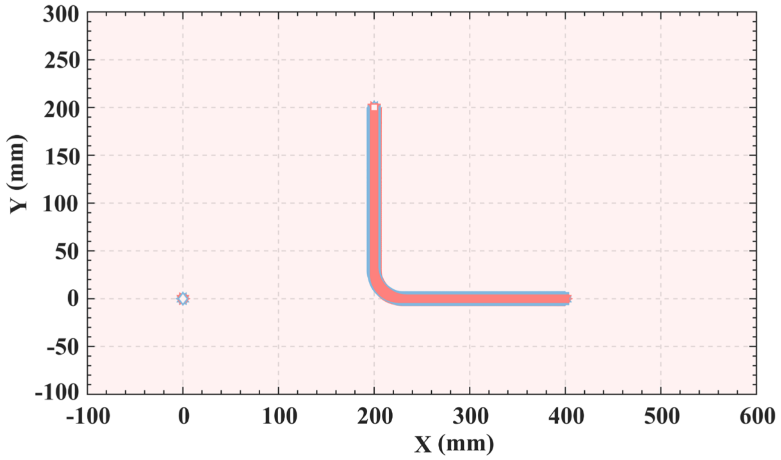


Figure 13. Motion trajectories in a two-dimensional plane.

4. Conclusions

This study proposes an improved snake optimization (ISO) algorithm. Meanwhile, a single-segment start-stop S-shaped speed curve with arc transition is constructed based on ISO, which is a method specifically designed for the rapid and stable operation requirements of the wafer-handling robotic arm. This method has conducted in-depth research and optimization on the key issues of time and impact within the context of robotic arm trajectory planning. Through a comparative analysis with the original S-shaped speed curve scheme under the same parameter settings, the new method improves the time efficiency by 24.1%, thereby significantly reducing the time required for the robotic arm to complete the task. Furthermore, the performance of the ISO is compared with that of SO, SA, PSO, GWO, and WOA on a series of benchmark test functions. The results indicate that, in terms of the two key performance indicators, —average value and variance, —the ISO algorithm outperforms the second ranked GWO algorithm by a margin of 38.9%. These improved simulations fully demonstrate the effectiveness and superiority of the proposed method in addressing the trajectory planning problem for wafer-handling robotic arms.

In this study, the shortcomings of the SO algorithm have been extensively studied and improved. Although some improvements have been made in searching for the optimal solution, there is still a risk of falling into local optima when dealing with some unimodal and multi-modal problems. Additionally, although Bayesian optimization improves the search efficiency, it consumes more computer resources. Therefore, the follow-up work of this study will focus on further adjusting the search mechanism of the SO algorithm to enhance its convergence abilities. Simultaneously, we plan to explore the combination of Bayesian optimization and bandit-based methods, aiming to reduce computational costs while improving the optimization efficiency.

Author Contributions: Conceptualization, YJ.J. and JL.Y.; Methodology, JL.Y.; Software, JL.Y.; Validation, JL.Y.; Formal analysis, JL.Y.; Investigation, YJ.J. and JL.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Basic Scientific Research Project of Higher Education Institutions of Liaoning Province (grant number JYTQN2023064).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Lin, Y.C.; Qian, C.; Awtar, S.; Okwudire, C.E. A Preliminary Investigation of Input Shaping to Reduce the Residual Vibration of a Wafer-Handling Robot. In Proceedings of the 2024 19th International Manufacturing Science and Engineering Conference (MSEC), Knoxville, Tennessee, USA, 17-21 June 2024. [CrossRef]
2. Chou, C.H.; Qian, C.; Lin, Y.C.; Awtar, S.; Okwudire, C.E. Feedforward Compensation of the Pose-Dependent Vibration of a Silicon Wafer Handling Robot. *CIRP Ann-Manuf Technol.* **2024**, 73, 321-324. [CrossRef]
3. He, Y.B.; Chen, J.J.; Gao, J.; Cui, C.Q.; Yang, Z.J.; Chen, X.; Chen, Y.; Zhang, K.; Tang, H.; Research on Motion Simulation of Wafer Handling Robot Based on SCARA. In Proceedings of the 2018 19th International Conference on Electronic Packaging Technology (ICEPT), Shanghai, China, 08-11 August 2018; pp. 734-739. [CrossRef]
4. Han, B.Y.; Zhao, B.; Sun, R.H. Research on Motion Control and Wafer-Centering Algorithm of Wafer-Handling Robot in Semiconductor Manufacturing. *Sensors* **2023**, 23, 8025. [CrossRef]
5. Shrivastava, A.; Dalla, V.K. Strategy of Smooth Motion Planning of Multi-Axes Space Manipulator Avoiding Dynamic Singularity in Cartesian Space. *J. Braz. Soc. Mech. Sci. Eng*. **2022**, 44, 278. [CrossRef]
6. Tagliavini, A.; Bianco, C.G. A Smooth Orientation Planner for Trajectories in the Cartesian Space. *IEEE Rob. Autom. Lett.* **2023**, 8, 2606-2613. [CrossRef]
7. Lin, J.J.; Rickert, M.; Knoll, A. Parameterizable and Jerk-Limited Trajectories with Blending for Robot Motion Planning and Spherical Cartesian Waypoints. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xian, China, 30 May-5 June 2021; pp. 13982-13988. [CrossRef]
8. Xiao, P.F.; Ju, H.H.; Li, Q.D. Point-to-point Trajectory Planning for Space Robots Based on Jerk Constraints. *Rev. Sci. Instrum*. **2021**, 92, 09450. [CrossRef]
9. Zhang, C.; Zhang, Z. Research on Joint Space Trajectory Planning of SCARA Robot Based on SimMechanics. In Proceedings of the 2019 Electronic and Automation Control Conference (ITNEC), Chongqing, China, 15-17 March 2019; pp. 1446-1450. [CrossRef]
10. Luo, L.; Guo, T.B.; Cui, K.K.; Zhang, Q. Trajectory Planning in Robot Joint Space Based on Improved Quantum Particle Swarm Optimization Algorithm. *Appl.* **2023**, 13, 7031. [CrossRef]
11. Wu, G.L.; Zhang, N. Kinematically Constrained Jerk-Continuous S-Curve Trajectory Planning in Joint Space for Industrial Robots. *Electronics* **2023**, 12, 1135. [CrossRef]
12. Hu, Y.H.; Zhang, S.L.; Chen, Y.H. Trajectory Planning Method of 6-DOF Modular Manipulator Based on Polynomial Interpolation. *J. Comput. Methods Sci*. **2023**, 23, 1589-1600. [CrossRef]
13. Chen, G.R.; Ma, M.; Guo, S.; Hou, B.W.; Wang, J.Z. An Easy-implemented Optimization Method of Trajectory Planning Based on Polynomial Interpolation. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26-28 July 2021; pp. 2802-2807. [CrossRef]
14. Shrivastava, A.; Dalla, V.K. Multi-Segment Trajectory Tracking of the Redundant Space Robot for Smooth Motion Planning Based on Interpolation of Linear Polynomials with Parabolic Blend. *P. I. Mech. Eng. C-J Mec*. **2022**, 236, 9255-9269. [CrossRef]
15. Cong, Y.Z.; Jiang, C.R.; Liu, H.; Du, H.B.; Gan, Y.H.; Jiang, C.H. Research on Trajectory Planning Method of Dual-Arm Robot Based on ROS. In Proceedings of the 2020 Chinese Automation Congress (CAC), Xian, China, 6-8 November 2020; pp. 2616-2621. [CrossRef]
16. Ayazbay, A.A.; Gani, B.; Sandugash, O.; Konrad, G.; Algazy, Z. Trajectory Planning, Kinematics, and Experimental Validation of a 3D-Printed Delta Robot Manipulator. *Mech. Eng. Robot. Res*. **2024**, 13. [CrossRef]
17. Wang, G.R.; Xu, F.; Zhou, Z.K.; Pang, Z.H. S-Velocity Profile of Industrial Robot Based on NURBS Curve and Slerp Interpolation. *Processes* **2022**, 10, 2195. [CrossRef]

18. Patle, B.K.; Chen, S.L.; Singh, A.; Kashyap, S.K. Optimal Trajectory Planning of the Industrial Robot Using Hybrid S-Curve-PSO Approach. *Rob. Intell. Autom.* **2023**, 43, 153-174. [CrossRef]
19. Wu, N.K.; Jia, D.Y.; Li, Z.Q.; He, Z.H. Trajectory Planning of Robotic Arm Based on Particle Swarm Optimization Algorithm. *Appl. Sci*. **2024**, 14, 8234. [CrossRef]
20. Du, Y.X.; Chen, Y.H. Time Optimal Trajectory Planning Algorithm for Robotic Manipulator Based on Locally Chaotic Particle Swarm Optimization. *Chin. J. Electron*. **2022**, 31, 906-914. [CrossRef]
21. Luo, L.; Guo, T.B.; Zhang, Q. Optimal Trajectory Planning for Robotic Arm Based on Improved Dynamic Multi-Population Particle Swarm Optimization Algorithm. *Int. J. Adv. Comput. Sci. Appl* **2024**, 15, 710-718. [CrossRef]
22. Zhong, F.F.; Liu, G.P.; Lu, Z.Y.; Hu, L.Y.; Han, Y.Y.; Xiao, Y.S.; Zhang, X.R. Dynamic Parameter Identification Based on Improved Particle Swarm Optimization and Comprehensive Excitation Trajectory for 6R Robotic Arm. *Ind. Robot* **2024**, 51, 148-166.
23. Liang, B.Q.; Lin, X.Z.; Liu, G.H.; Lei, J. Trajectory Analysis and Optimization of Sea Buckthorn Fruit Vibration Separation Manipulator Based on IPSO Algorithm. *Sci. Rep*. **2023**, 13, 20124.
24. Lu, Z.G.; Wang, X. Mechanical Arm Trajectory Planning Based on B-Spline and Whale Optimization Algorithm. *J. Northeast. Univ.* **2024**, 45, 683-689. [CrossRef]
25. Qie, X.H.; Kang, C.F.; Zong, S.C.; Chen, S.J. Trajectory Planning and Simulation Study of Redundant Robotic Arm for Upper Limb Rehabilitation Based on Back Propagation Neural Network and Genetic Algorithm. *Sensors* **2022**, 22, 4071.
26. Li, W.; Zhao, J.B.; Wang, J.Z. An Improved Snake Optimization Algorithm and Its Application in Manipulator Trajectory Planning. In Proceedings of the 2023 35th Chinese Control and Decision Conference. (CCDC), Yichang, China, 20-22 May 2023; pp. 4306-4313. [CrossRef]
27. Lu, S.Y.; He, Q.; Wang, Y.M.; Li, Y.B. Enhanced Snake Optimization Algorithm for Mobile Robot Path Planning. Trans. Microsyst. *Technol*. **2023**, 42, 127-133. [CrossRef]
28. Liang, X.M.; Shi, L.Y.; Long, W. Improved Snake Optimization Algorithm for Solving Constrained Optimization Problems. *Comput. Eng. Appl*. **2024**, 60, 76-87. [CrossRef]
29. Li, Q.L.; Ma, Q.H.; Weng, X. Dynamic Path Planning for Mobile Robots Based on Artificial Potential Field Enhanced Improved Multiobjective Snake Optimization. *J. Field Rob*. **2024**, 41, 1843-1863.
30. Jin, R.Y.; Rocco, P.; Geng, Y.H. Cartesian Trajectory Planning of Space Robots Using a Multi-Objective Optimization. *Aerosp. Sci. Technol*. **2021**, 108, 106360. [CrossRef]
31. Pu, Q.C.; Xu, X.R.; Li, Q.Q.; Zhang, H. Robotic Arm Time-Jerk Optimal Trajectory Based on Improved Dingo Optimization. *J. Braz. Soc. Mech. Sci. Eng*. **2024**, 46, 198. [CrossRef]
32. Zhang, X.C.; Chao, Y.S.; Li, C.Y.; Zhou, J.L. Multi-Objective Trajectory Planning for Industrial Robots Based on Machine Learning. *Mod. Manuf. Eng*. **2024**, 28–39. [CrossRef]
33. Liu, J.; Deng, Y.H.; Liu, Y.L.; Chen, L.L.; Hu, Z.Z.; Wei, P.Y.; Li, Z.B. A Logistic-tent Chaotic Mapping Levenberg Marquardt Algorithm for Improving Positioning Accuracy of Grinding Robot. *Sci. Rep*. **2024**, 14, 9649. [CrossRef]
34. Huang, Y.; Liang, F.C.; Fan, C.L.; Song, Z.F. Improved Particle Swarm Optimization Algorithm with Random Mutation and Perception. *J. Northwest. Polytech. Univ*. **2023**, 41, 428-438. [CrossRef]
35. Zhang, C.; Shao, Y.C.; Sun, H.J.; Xing, L.; Zhao, Q.; Song, Z.F. The WuC-Adam Algorithm Based on Joint Improvement of Warmup and Cosine Annealing Algorithms. *Math Biosci. Eng*. **2024**, 21, 1270-1285. [CrossRef]
36. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A Novel Meta-Heuristic Optimization Algorithm. *Knowl. Based Syst*. **2022**, 242, 108320.

37. Shi, W.J.; He, Z.; Tang, W.; Liu, W.F.; Ma, Z.Y. Path Planning of Multi-Robot Systems With Boolean Specifications Based on Simulated Annealing. *IEEE Rob. Autom. Lett.* **2022**, 7, 6091-6098. [CrossRef]
38. Zhao, Y.T.; Gan, L.; Li, W.G.; Liu, A. Path Planning of Spot Welding Robot Based on Multi-Objective Grey Wolf Algorithm. *J. Intell. Fuzzy Syst* **2021**, 41, 6181-6189. [CrossRef]
39. Gharehchopogh, F.S.; Gholizadeh, H.A. A Comprehensive Survey: Whale Optimization Algorithm and Its Applications. *Swarm Evol. Comput.* **2019**, 48,1-24. [CrossRef]

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.