

基于改进麻雀算法的机械臂轨迹优化

周雅豪

上海理工大学机械工程学院, 上海

收稿日期: 2025年3月28日; 录用日期: 2025年4月21日; 发布日期: 2025年4月29日

摘要

机械臂轨迹优化是机器人运动控制领域的核心研究问题, 旨在生成满足运动学与动力学约束的最优轨迹, 以提高运动效率、稳定性及能耗控制能力。本文针对标准麻雀搜索算法在优化精度及局部搜索能力方面的不足, 提出了一种基于改进麻雀搜索算法的机械臂轨迹优化方法。该方法结合3-5-3分段多项式插值, 以时间最优为优化目标, 在满足关节速度、加速度及冲击量等约束的前提下, 通过引入动态权重调整策略和蝴蝶算法, 增强搜索能力并提高解的质量。实验基于UR5机械臂进行仿真分析, 并与标准麻雀算法及粒子群优化算法进行对比。结果表明, 改进麻雀搜索算法在收敛速度、全局搜索能力及优化精度方面均优于其他方法, 使轨迹规划总耗时较PSO减少15.67%, 较SSA减少9.94%。该方法有效提升了机械臂运动的平稳性与执行效率, 为机械臂运行复杂任务提供了一种高效的轨迹优化方案。

关键词

机械臂, 轨迹优化, 改进麻雀搜索算法, 3-5-3分段多项式插值

Trajectory Optimization of Manipulator Based on Improved Sparrow Algorithm

Yahao Zhou

School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 28th, 2025; accepted: Apr. 21st, 2025; published: Apr. 29th, 2025

Abstract

The trajectory optimization of robot arm is the core research problem in the field of robot motion control, aiming at generating the optimal trajectory satisfying the kinematic and dynamic constraints to improve the motion efficiency, stability and energy consumption control ability. Aiming at the shortcomings of the standard sparrow search algorithm in optimization accuracy and local search ability, this paper proposes a robotic arm trajectory optimization method based on the

improved sparrow search algorithm. This method combines 3-5-3 piecewise polynomial interpolation, takes the time optimization as the optimization goal, meets the constraints of joint velocity, acceleration and impact, and introduces dynamic weight adjustment strategy and butterfly algorithm to enhance the search ability and improve the quality of the solution. Simulation analysis was carried out based on UR5 robot arm, and comparison was made with standard Sparrow algorithm and particle swarm optimization algorithm. The results show that the improved sparrow search algorithm is superior to other methods in terms of convergence speed, global search ability and optimization accuracy, and the total trajectory planning time is reduced by 15.67% compared with PSO and 9.94% compared with SSA. This method effectively improves the smoothness and execution efficiency of the manipulator, and provides an efficient trajectory optimization scheme for the manipulator to run complex tasks.

Keywords

Mechanical Arm, Trajectory Optimization, Improved Sparrow Search Algorithm, 3-5-3 Piecewise Polynomial Interpolation

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

机械臂轨迹规划是机器人运动控制领域的重要研究方向，旨在生成符合运动学与动力学约束的最优轨迹，以提升机械臂的运行效率、稳定性及能耗控制能力[1]。高效的轨迹规划方法不仅能够优化机械臂的运动路径，还能有效降低运行时间与关节冲击量，从而提高设备的使用寿命与作业质量。在工业制造、医疗手术、农业采摘及航空航天等领域，精确的轨迹优化对机械臂的稳定运行至关重要[2]。因此，如何在满足约束条件的前提下实现高效轨迹规划，成为当前研究的核心问题之一。目前，轨迹优化研究主要集中在时间最优[3]、能量最优、冲击最优及多目标优化等方向，不同优化方法依据具体应用场景，采用相应的优化策略，以提升运动系统的性能和效率。例如，时间最优优化侧重于缩短运动时间，提高生产效率；能量最优[4]优化旨在降低能耗，提升系统的经济性；冲击最优优化关注减少系统冲击，延长设备使用寿命[5]；而多目标优化则综合考虑多个因素，以实现性能的全方位提升，从而满足不同工程应用的需求[6]。近年来，智能优化算法在轨迹规划问题中得到了广泛应用，尤其是群体智能算法，如粒子群优化(PSO)[7]、遗传算法(GA)、蚁群优化(ACO)及人工蜂群算法(ABC)，在全局搜索能力和收敛性能方面展现了良好的优化效果[8]。然而，这些传统优化算法在高维、多约束或复杂环境下易陷入局部最优解[9]，导致轨迹规划的精度和收敛速度受限。为克服这些问题，研究者们不断探索更具鲁棒性的新型优化算法，以提升轨迹优化的性能[10]。周力等人[11]提出的改进差分进化(DE)算法通过动态参数调整策略，在降低位置误差与能量损耗方面表现优异。此外，崔靖凯等[12]将改进灰狼算法(GWO)应用于冗余机械臂的最优轨迹规划，通过引入非线性收敛因子和自适应权重策略，有效提升了算法在高自由度场景下的搜索效率。张铁等人[13]结合改进的 SARSA 强化学习算法与迭代交互法，对多轴运动系统的时间最优轨迹进行优化，以确保其满足运动学与动力学约束。

麻雀搜索算法(Sparrow Search Algorithm, SSA)是一种近年来提出的群体智能优化算法[14]，受麻雀群体觅食与警戒行为的启发，通过动态调整探索者(发现者)与跟随者(追随者)的行为，实现全局搜索与局部优化的协同优化。该算法具备较强的全局探索能力，能够有效跳出局部最优解，同时具备较快的收敛速

度，使其适用于复杂的轨迹优化任务。然而，标准麻雀优化算法在局部开发能力[15]和搜索稳定性方面仍存在一定不足[16]，可能影响优化结果的精度。因此对麻雀优化算法进行改进，以提升其在轨迹优化问题中的适应性，成为本研究的重点。本文提出了一种基于改进麻雀搜索算法的机械臂轨迹规划方法，融入蝴蝶算法，采用3-5-3分段多项式插值，并以时间最优作为优化目标，在满足机械臂运动学约束(如关节速度、加速度及冲击量)条件下，利用改进麻雀算法进行轨迹优化。通过MATLAB仿真实验，对比不同优化算法的性能，验证所提方法在时间优化方面的有效性。

2. 机械臂模型建立

本文研究对象为UR5机械臂，以其高灵活性、高精度和较强的负载能力广泛应用于工业制造、自动化生产、医疗、科研等领域。UR5机械臂具有6个旋转关节，可实现复杂的空间运动，其关节驱动采用伺服电机，具备较高的动态响应特性。由于UR5机械臂的运动学构型与人类手臂相似，使其在协作机器人领域具有较高的适用性。UR5机械臂模型如图1所示。

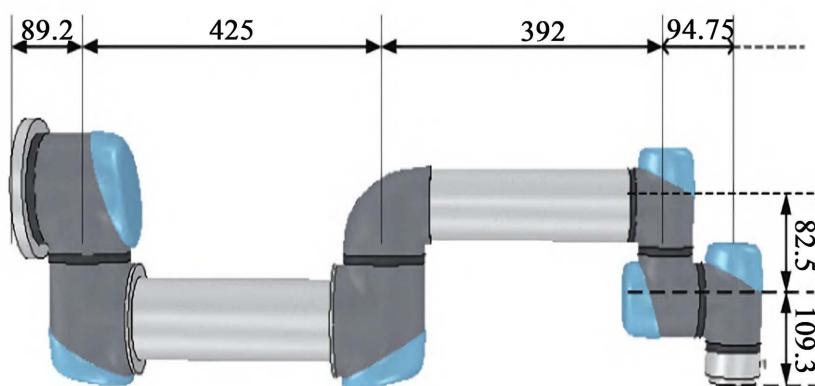


Figure 1. UR5 robot arm model
图 1. UR5 机械臂模型

在机械臂轨迹规划中，首先需要建立其运动学模型，以描述末端执行器相对于基座的位姿(位置与姿态)。本文采用D-H参数法对UR5机械臂进行建模，该方法通过四个参数(连杆长度 a_i 、连杆转角 α_i 、关节角度 θ_i 、关节偏移 d_i)来定义机械臂各关节之间的空间变换关系。根据UR5机械臂的结构，其D-H参数表如表1所示，UR5在MATLAB的仿真模型图2所示。

Table 1. UR5D-Hparameters
表 1. UR5D-H 参数表

i	$\alpha_i/(\circ)$	a_i/mm	$\theta_i (\circ)$	d/mm	转动范围/(\circ)
1	90	0	θ_1	89.20	± 360
2	0	-425	θ_2	0	± 360
3	0	-392	θ_3	0	± 360
4	90	0	θ_4	109.30	± 360
5	-90	0	θ_5	94.75	± 360
6	0	0	θ_6	82.50	± 360

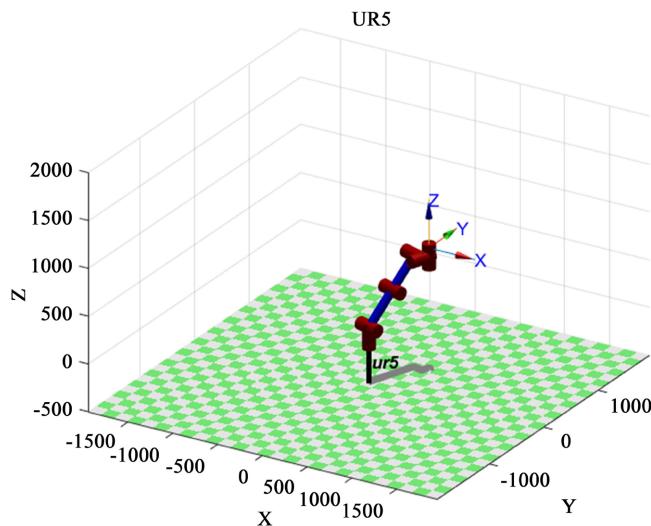


Figure 2. UR5 simulation model
图 2. UR5 仿真模型

3.3-5-3 分段组合多项式插值构造关节轨迹

轨迹插值是机械臂轨迹规划的重要方法之一，其核心目标是通过数学插值方法构造机械臂的关节轨迹，使其运动过程平滑，并满足关节速度、加速度及冲击(Jerk)等运动学与动力学约束。在轨迹生成过程中，常用的插值方法包括三次多项式插值、五次多项式插值及 3-5-3 分段多项式插值。其中，3-5-3 分段组合多项式插值结合了三次多项式与五次多项式的优势，通过三段式结构，将轨迹规划分为三次、五次和三次多项式插值段在保证轨迹光滑性的同时，能够有效优化关节运动时间，减少冲击，提高轨迹的平稳性。

分段结构如图 3 所示。

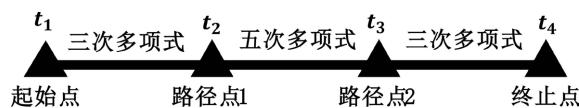


Figure 3. Schematic diagram of piecewise interpolation
图 3. 分段插值示意图

由分段插值示意图可知，对于任意关节在起始点时 $t=t_1$ ，对应关节角度为 θ_1 ；在路径点 1 处 $t=t_2$ ，对应关节角度为 θ_2 ；在路径点 2 处 $t=t_3$ ，对应关节角度为 θ_3 ；在终止点处 $t=t_4$ ，对应关节角度为 θ_4 。

本文采用 3-5-3 分段组合多项式插值方法构造机械臂的关节轨迹，表达式如式(1)所示：

$$\begin{cases} \theta_{j1}(t) = a_{j10} + a_{j11}t_{j1} + a_{j12}t_{j1}^2 + a_{j13}t_{j1}^3 \\ \theta_{j2}(t) = a_{j20} + a_{j21}t_{j2} + a_{j22}t_{j2}^2 + a_{j23}t_{j2}^3 + a_{j24}t_{j2}^4 + a_{j25}t_{j2}^5 \\ \theta_{j3}(t) = a_{j30} + a_{j31}t_{j3} + a_{j32}t_{j3}^2 + a_{j33}t_{j3}^3 \end{cases} \quad (1)$$

式中， $\theta_{j1}(t)$ ， $\theta_{j2}(t)$ 和 $\theta_{j3}(t)$ 分别表示 3-5-3 分段多项式的第一段使用三次多项式，第二段使用五次多项式，第三段使用三次多项式。其中 a_{j1i} 、 a_{j2i} 、 a_{j3i} 分别为 3-5-3 分段插值多项式中，机械臂关节变量与时间之间函数关系的系数。 t_{j1} 、 t_{j2} 、 t_{j3} 分别为分段多项式第一段，第二段，第三段的运行时间。在分段多项式中将角度值，角速度值，角加速度值的约束添加，表达式如式(2)~(4)所示：

$$\begin{cases} \theta_1(0) = \theta_0 \\ \theta_2(0) = \theta_1(1) \\ \theta_3(0) = \theta_2(1) \\ \theta_3(1) = \theta_f \end{cases} \quad (2)$$

$$\begin{cases} \theta'_1(0) = \theta'_0 \\ \theta'_2(0) = \theta'_1(1) \\ \theta'_3(0) = \theta'_2(1) \\ \theta'_3(1) = \theta'_f \end{cases} \quad (3)$$

$$\begin{cases} \theta''_1(0) = \theta''_0 \\ \theta''_2(0) = \theta''_1(1) \\ \theta''_3(0) = \theta''_2(1) \\ \theta''_3(1) = \theta''_f \end{cases} \quad (4)$$

根据插值约束条件, 可构建矩阵 A , 表示插值时间矩阵, 如式(5)所示, 其中 t_1 、 t_2 、 t_3 分别代表组合多项式的三段插值时间。插值位移矩阵 B 如式(6)所示, 其中 θ 代表第 i 段插值的位移量。为求解 3-5-3 组合分段多项式插值系数, 构建方程组如式(7)所示, 并施加约束条件, 以确保插值的平滑性及连续性。

$$A = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 20t_2^3 & 12t_2^2 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

$$B = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \theta_3 \ 0 \ 0 \ \theta_0 \ 0 \ 0 \ \theta_2 \ \theta_1] \quad (6)$$

$$H = A^{-1}B \quad (7)$$

4. 改进麻雀算法

4.1. 基础麻雀搜索算法

麻雀算法(Sparrow Search Algorithm, SSA)是基于麻雀觅食行为的启发式优化算法, 模拟麻雀在自然环境中觅食和寻找食物的过程。算法通过更新麻雀群体的位置信息来找到问题的最优解。在该算法中, 每只麻雀代表了问题的一个解, 而群体中的每个麻雀则会不断通过与其他麻雀的交互来调整自身的位置。麻雀搜索算法的流程可以分为以下几个主要步骤:

- 1) 麻雀的位置信息表示

在麻雀算法中，每只麻雀的位置是一个 D 维的向量，表示在问题空间中的一个解。假设共有 N 只麻雀，每只麻雀的维度为 D ，那么第 i 只麻雀的位置表示为：

$$X_i = [x_{i1}, x_{i2}, \dots, x_{id}]$$

式中： X_i 是第 i 只麻雀的整体位置向量， x_{id} 表示第 i 只麻雀在 d 维度上的位置。

2) 觅食行为模拟

每只麻雀通过搜索邻域区域来寻找食物，而其觅食行为的强度由其他麻雀的位置和适应度所影响。在每一代，麻雀根据当前的搜索状态和邻域麻雀的引导，逐步更新自己的位置。在麻雀算法中，麻雀的位置更新规则通常依赖于适应度的比较。

3) 麻雀位置更新

麻雀的每次位置更新规则由以下公式给出。这个公式分为两种情况，分别根据麻雀的适应度值和环境条件来决定是否跳跃到远处或者继续在当前区域附近搜索。

$$X_{id}^{t+1} = X_i^t \cdot \exp\left(-i \cdot \frac{R}{\alpha \cdot iter_{max}}\right) (R < M) \quad (8)$$

$$X_{id}^{t+1} = X_{id}^t + QL (R \geq M) \quad (9)$$

X_i^{t+1} 为第 i 只麻雀在 $t+1$ 代的新位置； X_i^t 为第 i 只麻雀在当前代的当前位置； $\exp \cdot$ 表示自然对数的指数函数，模拟了麻雀在跳跃过程中位置的扩展； α 为 $(0,1]$ 的均匀随机数； $iter_{max}$ 表示预先设定的最大迭代次数； i 为当前代的代数，通常随着代数的增加逐渐减小，表示搜索过程中的变化速度； Q 表示一个来自标准正态分布的随机数，用来增加随机性，从而避免算法过早陷入局部最优； L 代表随机游走步数，表示麻雀的跳跃步长，通常与麻雀的适应度相关，适应度较高的麻雀跳跃步长较小，适应度较差的麻雀跳跃步长较大； QL 的作用是模拟麻雀在局部搜索区域内的随机扰动； R 表示第 i 只麻雀当前的适应度值。适应度值通常用来衡量该麻雀当前位置的好坏； M 是预设的一个标准值，表示麻雀所处的环境条件。若 $R < M$ 则表明当前解不够好，麻雀需要向更远的区域移动进行探索，如式(8)所示；当麻雀适应度值达到一定条件 ($R \geq M$) 时，它会选择在当前位置附近进行局部优化，式(9)所示。

4) 全局与局部搜索的平衡

麻雀算法通过动态调整全局搜索与局部搜索的比重来提高搜索效率。在适应度变化较大的情况下，算法将加强全局搜索(即使用较大的跳跃步长，广泛搜索解空间)。而在适应度变化较小时，算法将加强局部搜索，细化当前的解。

5) 位置更新与危险预警

麻雀算法不仅依据适应度信息更新麻雀的位置，还会根据局部环境的风险情况来调整搜索策略。假如某个麻雀发现附近的环境不适宜或风险较大，则它可能会调整跳跃方向，从而避免进入不良的区域。跟随者与预警者位置跟新公式如式(10)和式(11)所示。

$$X_{id}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{id}^t}{i^2}\right) & \text{if } i > \frac{n}{2} \\ X_p^{t+1} + |X_{id}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{if } i \leq \frac{n}{2} \end{cases} \quad (10)$$

$$X_{id}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{id}^t - X_{best}^t| & f_i > f_g \\ X_{id}^t + K \cdot \left(\frac{|X_{id}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} \right) & f_i = f_g \end{cases} \quad (11)$$

式中, X_{worst}^t 为当前种群中最差解的位置; i^2 作为归一化因子, 控制搜索步长 X_p^{t+1} 表示当前最优的麻雀位置; A^+ 是随机数, 控制搜索范围; X_{best}^t 为当前种群中的最优解; β 是扰动因子, 使个体向全局最优解靠近; K 为随机扰动因子; $(f_i - f_w) + \varepsilon$ 避免分母为零。

麻雀搜索算法是近年来新兴的群体智能优化方法。与其他群体智能算法类似, 尽管麻雀搜索算法在全局搜索方面表现出色, 但其局部搜索能力相对较弱。为提高算法在局部优化过程中的表现, 本节将在标准麻雀搜索算法的基础上进行改进。

4.2. 改进麻雀搜索算法

将蝴蝶算法引入麻雀搜索算法来提高全局探索和局部搜索的能力。通过调整麻雀位置更新公式, 引入蝴蝶算法的策略, 使得麻雀能够在全局搜索和局部搜索之间动态平衡。蝴蝶算法的优势在于其较强的局部搜索能力, 它通过模拟蝴蝶在花间飞舞的行为, 能够较好地优化局部解, 进行精细的搜索。

在算法的迭代过程中, 动态调整麻雀算法和蝴蝶算法的权重。在早期阶段, 更依赖麻雀算法进行广泛搜索; 在优化过程的后期, 逐步增加蝴蝶算法的权重, 聚焦于更精细的局部优化, 增强全局搜索和局部搜索的平衡。基于适应度(Fitness)的动态调整是一种根据当前解的适应度(即目标函数的值)来实时调整算法权重的方法。适应度反映了当前解的质量, 动态调整策略可以根据适应度变化的情况来决定何时加强全局搜索(使用麻雀算法), 何时加强局部搜索(使用蝴蝶算法)。若当前解的适应度变化较大, 说明解空间中仍有较多潜在解需要探索, 适合使用全局搜索; 而当适应度变化较小时, 说明当前解已经接近最优解, 可以通过局部搜索精细优化解的质量。这种方法能有效地避免在局部最优解附近停滞, 从而提升优化过程的效率和效果。

适应度变化可以通过以下公式计算:

$$\Delta f_{\text{fitness}} = |f_{\text{best}}(t) - f_{\text{best}}(t-1)| \quad (12)$$

式中, $f_{\text{best}}(t)$ 和 $f_{\text{best}}(t-1)$ 分别表示第 t 代和第 $t-1$ 代最优解的适应度值。

若是仅仅通过对当前解和上一代解的适应度值, 计算出适应度变化, 可能会导致算法容易陷入局部最优, 尤其在解决复杂问题时。因此加入计算适应度变化的滑动平均值来提高算法的稳定性和全局搜索能力, 避免因偶然的适应度变化导致过早的权重调整。

计算第 t 代适应度变化的滑动平均值:

$$\Delta f_{\text{smooth}}(t) = \frac{1}{N_{\text{smooth}}} \sum_{i=t-N_{\text{smooth}}+1}^t \Delta f_{\text{fitness}}(i) \quad (13)$$

式中, N_{smooth} 表示滑动平均的窗口大小, 窗口越大, 波动越平滑, 但反应速度可能会慢一些。窗口越小, 反应更灵敏, 但可能对偶然的波动过度反应; $\Delta f_{\text{fitness}}(i)$ 表示当前代最优解的适应度与上一代最优解的差异。它反映了每一代解的变化程度。

对麻雀算法和蝴蝶算法的权重进行动态调整, 根据适应度变化的滑动平均值和阈值来动态调整麻雀和蝴蝶算法的权重。根据具体问题和实验调优预设一个适应度变化阈值 $\Delta f_{\text{threshold}}$ 。

当适应度变化较大, 大于阈值时:

$$\omega_{\text{ssa}}(t) = \min\left(1, \frac{\Delta f_{\text{smooth}}(t)}{\Delta f_{\text{threshold}}}\right) \quad (14)$$

当适应度变化较小, 小于阈值时:

$$\omega_{\text{boa}}(t) = 1 - \omega_{\text{ssa}}(t) \quad (15)$$

融合蝴蝶算法后具体的改进的麻雀搜索算法步骤如下：

步骤一：初始化种群。初始化麻雀群体和蝴蝶群体的个体解。每个个体的初始解为一个随机值。设定以下参数：滑动平均窗口大小 N_{smooth} ；适应度变化的阈值 $\Delta f_{\text{threshold}}$ ，用于判断是否加强全局搜索或局部搜索；最大迭代次数 T_{max} ；初始权重： $\omega_{\text{ssa}}(0)=0.5, \omega_{\text{boa}}(0)=0.5$ ；初始适应度： $f_{\text{best}}(0)$ 为最优解；设置滑动平均适应度变化的初始值 $\Delta f_{\text{smooth}}(0)=0$ 。

步骤二：适应度评估。计算当前种群的适应度，据适应度值，更新全局最优解。

步骤三：计算适应度变化。计算当前代与上一代的适应度变化

步骤四：滑动平均计算。计算滑动平均适应度变化，通过滑动平均来平滑适应度变化的波动，减少短期波动对权重调整的影响。

步骤五：动态权重调整。麻雀算法的权重根据滑动平均适应度变化与预设阈值进行比较，调整麻雀算法的权重。如果滑动平均适应度变化较大(大于阈值)，则增加麻雀算法的权重，表示更依赖全局搜索。否则，减小麻雀算法的权重。根据麻雀算法的权重，计算蝴蝶算法的权重。如果麻雀算法的权重较大，则蝴蝶算法的权重较小，表示依赖全局搜索；反之，则依赖局部搜索。

步骤六：搜索更新。

步骤七：更新全局最优解。

步骤八：停止条件。

改进麻雀算法流程图如图 4 所示。

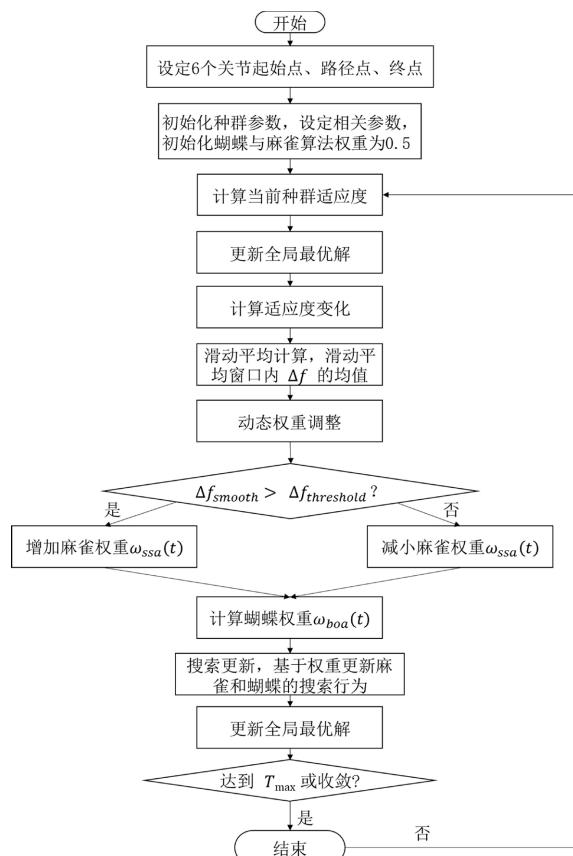


Figure 4. Flow chart of improved Sparrow algorithm

图 4. 改进麻雀算法流程图

5. 仿真结果与分析

5.1. 仿真设计

本文以 UR5 机械臂为研究对象，基于 MATLAB 软件中的 Robotics Toolbox 工具箱开展仿真分析。首先，采用 3-5-3 混合多项式进行轨迹插值，以确保轨迹的平滑性和连续性；随后，基于改进的麻雀算法，对机械臂的轨迹进行时间最优规划。研究选取了笛卡尔坐标系下的四个关键轨迹点，分别为 $P_1[-700, 200, 800]$ 、 $P_2[-300, 125, 1000]$ 、 $P_3[435, 785, 1025]$ 和 $P_4[800, -601, 600]$ ，其运动轨迹如图 5 所示。通过逆运动学计算，将笛卡尔空间中的位姿信息转换为对应的关节角度，其关节空间角度如表 2 所示。此外，UR5 机械臂各关节的运动学约束参数列于表 3，以确保轨迹规划符合机械臂的实际运动能力及动力学约束。

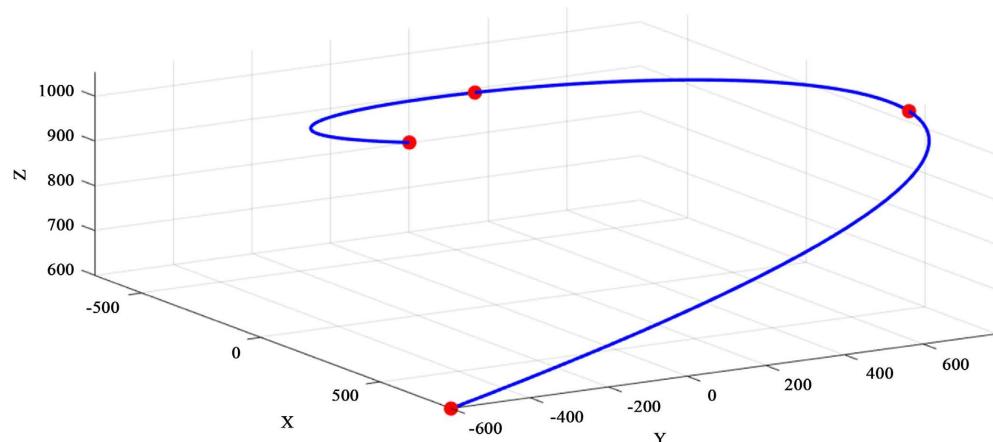


Figure 5. Motion trajectory

图 5. 运动轨迹

Table 2. Corresponding joint angles

表 2. 对应的各关节角度

关节位置	关节 1	关节 2	关节 3	关节 4	关节 5	关节 6
P_1	-25.2188	44.8152	-0.0197	45.0923	89.8880	64.7841
P_2	-42.8992	14.3382	-0.0799	75.2622	89.9529	-47.1029
P_3	70.6249	-42.8564	-0.0439	-46.3651	-89.4308	19.3823
P_4	-29.4114	-64.3680	0.0110	-26.6506	-89.6939	119.4000

Table 3. Constraints of each joint

表 3. 各关节约束条件

约束条件	最大角速度 V_{\max} (°/s)	最大角加速度 A_{\max} (°/s ²)	最大角加加速度 R_{\max} (°/s ³)
关节 1	25	30	40
关节 2	25	30	40
关节 3	25	30	40
关节 4	30	40	50
关节 5	30	40	50
关节 6	35	45	60

5.2. 结果分析

为了验证改进麻雀搜索算法(ISSA)的优化能力，本文将其与标准麻雀搜索算法(SSA)、粒子群优化算法(PSO)进行对比分析。初始设定四个插值点时间间隔为 4 秒，则初始机械臂总运行时间为 12 秒。设定迭代次数为 50，种群为 1000，初始麻雀算法权重 $\omega_{ssa} = 0.5$ ，初始蝴蝶算法权重 $\omega_{boa} = 0.5$ 。以关节 1 为例，做出适应度收敛图。

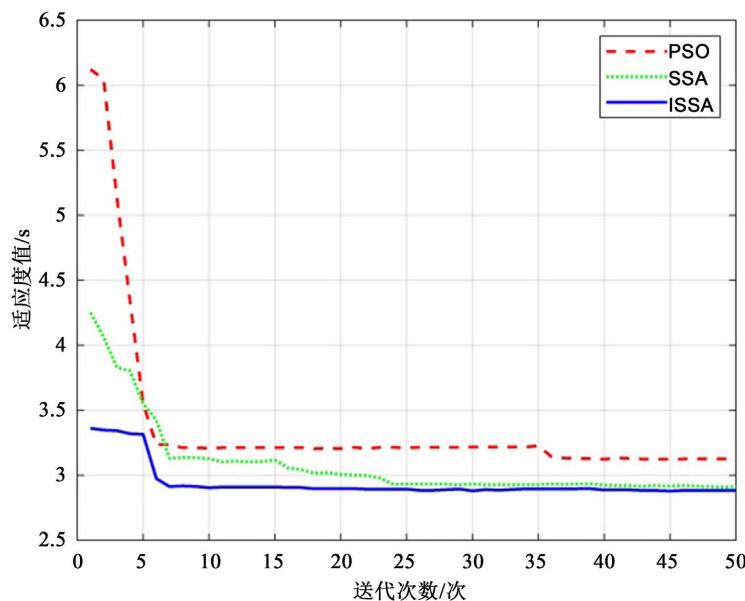


Figure 6. Fitness convergence diagram of joint 1

图 6. 关节 1 适应度收敛图

如图 6 所示，展示了三种算法在关节 1 的适应度值随迭代次数变化的收敛曲线，改进的麻雀算法在前 5 次迭代内就出现了明显的下降，并在 10 次迭代之后逐渐趋于稳定。其优化效率较高。粒子群算法最终稳定在 3.3 s，标准麻雀算法最终稳定在 3.1 s，改进的麻雀算法的最终适应度值约 2.9，是三种算法中最优的，表明其优化能力最强。综合收敛速度和最终优化结果来看，改进麻雀搜索算法具有最优的收敛性能和最优解质量。三种算法的优化的时间段结果如表 4~6 所示。

Table 4. Optimization results of PSO algorithm
表 4. PSO 算法优化结果

关节编号	第一段	第二段	第三段
关节 1	1.045	1.150	1.078
关节 2	1.590	1.892	1.314
关节 3	3.428	1.118	2.280
关节 4	1.962	3.842	1.122
关节 5	1.104	3.015	1.092
关节 6	0.873	3.755	0.918
各段时间最大值	3.428	3.842	2.280

Table 5. Optimization results of SSA algorithm
表 5. SSA 算法优化结果

关节编号	第一段	第二段	第三段
关节 1	0.885	1.198	0.812
关节 2	1.715	1.78	1.315
关节 3	3.102	0.825	1.692
关节 4	1.784	4.148	1.142
关节 5	0.121	2.545	0.092
关节 6	1.473	2.731	0.855
各段时间最大值	3.102	4.148	1.692

Table 6. Optimization results of ISSA algorithm
表 6. ISSA 算法优化结果

关节编号	第一段	第二段	第三段
关节 1	0.823	1.31	0.81
关节 2	1.71	1.84	1.195
关节 3	2.94	0.77	1.19
关节 4	1.79	3.918	1.12
关节 5	0.082	2.615	0.085
关节 6	1.304	3.08	0.83
各段时间最大值	2.94	3.918	1.195

基于表中数据取 6 个关节中耗时最长的阶段计算得出，不同算法在整体规划中的总时间消耗有所差异。其中，粒子群优化算法的总耗时为 9.55 s，标准麻雀搜索算法的总耗时为 8.942 s，而改进麻雀搜索算法的总耗时则为 8.053 s。相比之下，改进的麻雀算法在时间优化方面表现更优，其总耗时较粒子群算法减少 1.497 s (约 15.67%)，较标准麻雀算法减少 0.889 s (约 9.94%)。这一结果表明，改进的麻雀搜索算法在优化轨迹规划时间方面更加高效，能够有效缩短各阶段的运行时间，从而提高整体系统的执行效率。

将改进的麻雀搜索算法得到的时间结果代入 3-5-3 分段多项式中，对机械臂的各个关节角进行仿真分析如图 7 所示。

图 7 显示，在应用改进的麻雀搜索算法后，机械臂的角位移、角速度和角加速度曲线均表现出平稳、连续且无突变的特性。这种改进有效减少了机械臂运动过程中的冲击和振动，满足了其运动需求，并显著提升了工作效率。

6. 结论

本文研究了基于改进麻雀搜索算法的机械臂轨迹优化方法。通过建立 UR5 机械臂的运动学模型，采用 3-5-3 分段多项式插值构造关节轨迹，并利用改进的麻雀搜索算法优化轨迹规划，使其在满足运动学约束的前提下实现时间最优目标。实验结果表明，改进的麻雀搜索算法相较于标准麻雀搜索算法和粒子群优化算法，在收敛速度、全局搜索能力以及时间优化方面均表现出更优的性能。改进麻雀搜索算法在优化过程中收敛更快，最终规划出的轨迹较粒子群减少了 1.497 秒，较标准麻雀算法减少了 0.889 秒，显著提升了机械臂的执行效率。

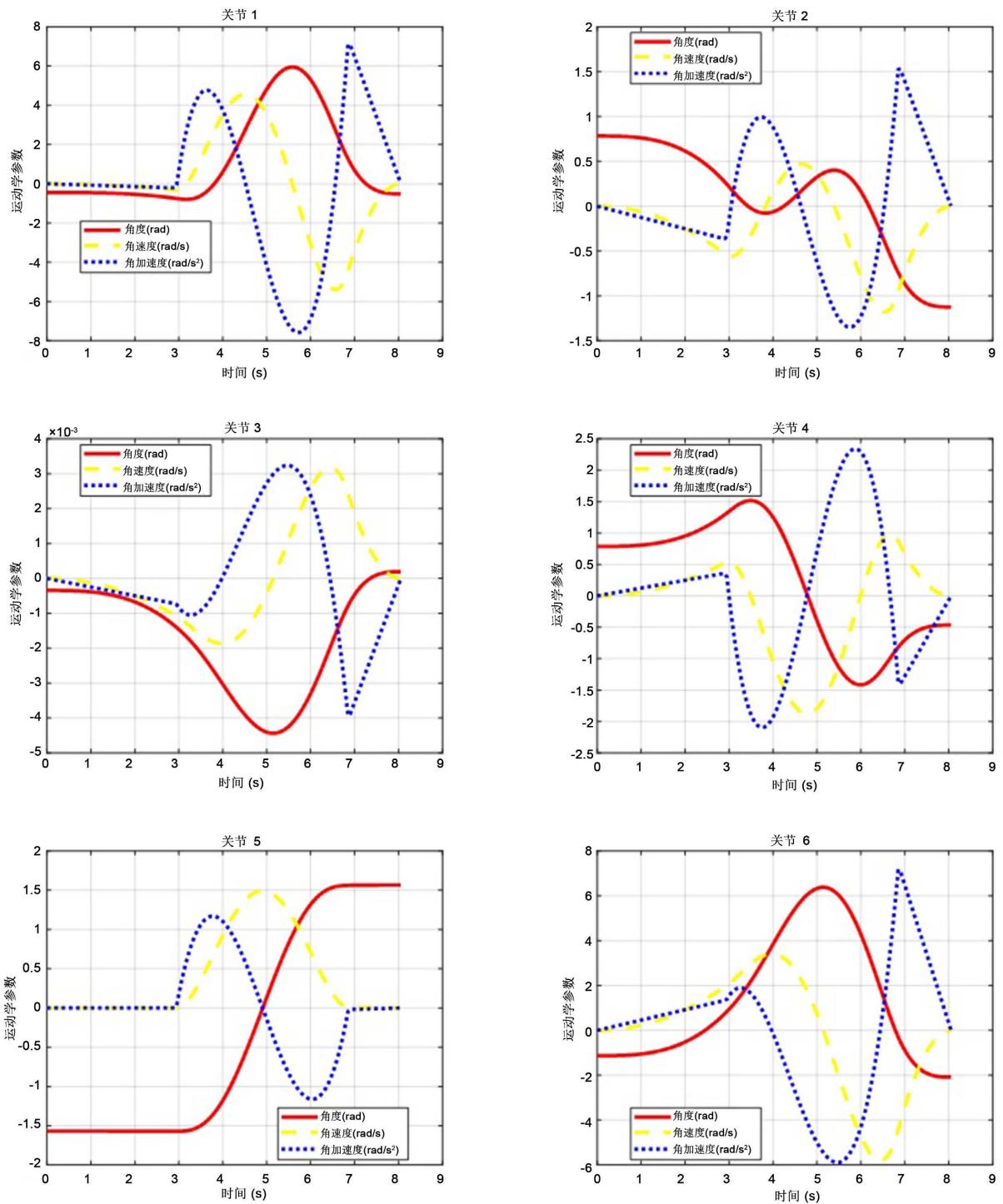


Figure 7. Curves of angular displacement, velocity and acceleration of each joint of the manipulator
图 7. 机械臂各关节角位移、速度、加速度曲线图

本文提出的轨迹优化方法能够有效提升机械臂的运动性能，为机械臂相关应用场景提供了高效、平稳的轨迹规划方案。未来研究可以进一步结合深度强化学习或其他智能优化算法，探索更加智能化、自适应的轨迹优化方法，以提升机械臂在复杂环境下的适应能力和鲁棒性。

参考文献

- [1] 冯建超, 鲁文其, 林军华, 等. 基于 3-5-3 多项式插补的六轴机器人轨迹优化[J]. 机械与电子, 2023, 41(10): 75-80.
- [2] 王磊, 崔嵩, 牛婷婷. 复杂环境下多策略改进麻雀搜索的定位算法[J]. 小型微型计算机系统, 2024, 45(11): 2768-2776.
- [3] 全浩源, 马立东, 孟志娟, 等. 管棒材拆捆机器人系统与轨迹规划研究[J]. 组合机床与自动化加工技术, 2021(7): 40-43, 48.
- [4] Mohammed, A., Schmidt, B., Wang, L. and Gao, L. (2014) Minimizing Energy Consumption for Robot Arm Movement. *Procedia CIRP*, **25**, 400-405. <https://doi.org/10.1016/j.procir.2014.10.055>
- [5] 胡智宇. 基于冲击最优的弧焊机器人轨迹规划研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2019.
- [6] Pu, Q., Xu, X., Li, Q. and Zhang, H. (2024) Robotic Arm Time-Jerk Optimal Trajectory Based on Improved Dingo Optimization. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, **46**, Article No. 198. <https://doi.org/10.1007/s40430-024-04694-4>
- [7] Wu, G. and Zhang, N. (2023) Kinematically Constrained Jerk-Continuous S-Curve Trajectory Planning in Joint Space for Industrial Robots. *Electronics*, **12**, Article 1135. <https://doi.org/10.3390/electronics12051135>
- [8] 马振书, 张云峰, 孙华刚, 等. 排爆机械臂多目标轨迹规划研究[J]. 振动与冲击, 2016, 35(2): 116-123
- [9] 牛明建. 基于多目标优化的串联机械臂轨迹规划方法研究[D]: [硕士学位论文]. 杭州: 浙江理工大学, 2023.
- [10] Hua, Y.X. (2021) Review of Trajectory Planning for Industrial Robots. *Journal of Physics: Conference Series*, **1906**, Article ID: 012041. <https://doi.org/10.1088/1742-6596/1906/1/012041>
- [11] 周力, 何荣誉, 李浩, 等. 基于改进 DE 算法的机械臂轨迹规划的设计[J]. 自动化技术与应用, 2024, 43(9): 12-15, 41.
- [12] 崔靖凯, 王凯, 范正奇, 等. 基于改进灰狼算法的冗余机械臂最优轨迹规划[J/OL]. 控制与决策: 1-10. <https://doi.org/10.13195/j.kzyjc.2024.1072>, 2025-03-12.
- [13] 张铁, 廖才磊, 邹焱飚, 等. 采用强化学习的多轴运动系统时间最优轨迹优化[J]. 西安交通大学学报, 2021, 55(7): 33-40.
- [14] 许艳涛, 邹光明, 王众玄, 等. 基于改进麻雀算法机械臂时间最优轨迹规划[J]. 组合机床与自动化加工技术, 2024(9): 86-90, 97.
- [15] 周明月, 周明伟, 刘桂岐, 等. 基于改进蝴蝶算法的机械臂时间最优轨迹规划[J]. 计算机科学, 2023, 50(S2): 119-126.
- [16] Li, Q., Wu, W., Xiang, J., Li, H. and Wu, C. (2014) A Hybrid Robot for Friction Stir Welding. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, **229**, 2639-2650. <https://doi.org/10.1177/0954406214562848>