文章编号：1003－1251(2026)05－0055－08

基于四重引导机制 IPSOGSA 算法的机械臂时间最优轨迹规划

罗旭涛，吴东升

（沈阳理工大学 自动化与电气工程学院，沈阳 110159）

摘　要：针对传统 3-5-3 多项式轨迹规划中时间分配固化、优化效果有限，以及现有粒子群优化－引力搜索混合算法(PSOGSA)忽略个体认知学习导致的结构性缺陷，本文提出一种基于 IPSOGSA 算法的机械臂时间最优轨迹规划方法。通过引入个体最优学习项，构建四重引导机制，增强算法搜索多样性和局部精细搜索能力；通过设计动态参数调整策略和停滞处理机制，提高算法自适应能力和稳定性。通过 6 个基准测试函数验证表明，IPSOGSA 算法相较于 PSOGSA 算法收敛精度提高 1～6 个数量级；在七自由度机械臂轨迹规划实验中，IPSOGSA 算法时间方差为 0.000 001，稳定性相比 PSOGSA 算法约提升 4 倍；生成轨迹满足物理约束且运动平滑，为七自由度机械臂时间最优轨迹规划提供了有效解决方案。

关　键　词：机械臂；轨迹规划；时间优化；PSOGSA 算法；多项式插值

中图分类号：TP242.2　　**文献标志码：**A　**DOI：**10.3969/j.issn.1003－1251.2026.05.008

Time-Optimal Trajectory Planning for Manipulators Based on an IPSOGSA Algorithm with a Four-Component Guidance

LUO Xutao，WU Dongsheng

（Shenyang Ligong University，Shenyang 110159，China）

Abstract：To address the problems of fixed time allocation and limited optimization in traditional 3-5-3 polynomial trajectory planning，as well as the structural deficiencies of existing particle swarm optimization-gravitational search algorithm（PSOGSA）algorithms that ignore individual cognitive learning，a time-optimal trajectory planning method for manipulators based on an IPSOGSA algorithm is proposed in this paper. By introducing individual optimal learning terms，a Four-Component guidance is constructed to enhance search diversity and local fine search ability of the algorithm. Dynamic parameter adjustment strategy and stagnation handling mechanism are designed to improve adaptive capability and stability of the algorithm. Through six benchmark test functions verification，the IPSOGSA algorithm achieves convergence accuracy improvements by 1 to 6 orders of magnitude，compared to PSOGSA algorithm. In seven-degree-of-freedom manipulator trajectory planning experiments，the algorithm achieves a time variance of 0.000 001，indicating a stability improvement of approximately 4 times compared to PSOGSA algorithm. The generated trajectories satisfy physical constraints with smooth motion，providing an practical solution for time-optimal trajectory

收稿日期：2025－08－27

基金项目：辽宁省教育厅重点攻关项目(JYTZD2023006)

作者简介：罗旭涛(2001—)，男，硕士研究生；吴东升(1973—)，通信作者，男，教授，博士。

planning of seven-degree-of-freedom manipulators.

Key words: manipulator; trajectory planning; time optimization; PSOGSA algorithm; polynomial interpolation

随着智能制造和工业 4.0 的发展,机械臂广泛应用于航空航天、汽车制造、精密装配等领域[1]。轨迹规划作为机械臂运动控制的关键技术[2],直接影响着机械臂的工作效率、运动平稳性和任务完成质量[3],已成为机器人学研究的重要课题。

机械臂轨迹规划按规划空间可分为关节空间和笛卡尔空间轨迹规划。关节空间轨迹规划具有计算效率高、无奇点问题等优势[4]。在关节空间轨迹规划中,多项式插值方法是实现平滑轨迹的关键技术,它结合了 3 次和 5 次多项式的优点,既保证了计算效率,又有效改善了轨迹的平滑性[5]。相比更高阶的多项式插值,如 5-7-5 多项式插值[6],3-5-3 方法具有更低的计算复杂度,在轨迹平滑度要求相对宽松和计算资源有限等场景下更适用于实时控制。然而,传统多项式插值法中的时间分配策略往往过于保守,未能充分挖掘机械臂在约束条件内的运动潜力。

针对 3-5-3 多项式插值轨迹规划的时间最优问题,学者们开展了大量研究工作,提出了局部混沌粒子群优化算法[7]、改进粒子群算法[8]、基于终端交叉和导向的扰动粒子群算法[9]等多种粒子群类优化(PSO)算法。PSO 算法因实现简单、收敛速度快等特点[10]受到广泛关注,但该算法存在易陷入局部最优、全局搜索能力不足的固有缺陷。

针对 PSO 算法的局限性,Mirjalili 等[11]提出了粒子群优化 - 引力搜索混合算法(PSOGSA),通过引入引力搜索算法(GSA)来增强全局搜索性能。该混合算法在工程优化领域得到了广泛应用[12-13],但仍存在过早收敛、局部搜索能力不强、搜索精度不高等问题。为改善算法性能,研究者们提出了多种改进策略,Kar 等[14]引入模糊突变机制以缓解过早收敛,游达章等[15]采用自适应参数调整和分层适应度处理来提升整体性能。然而,现有改进策略主要聚焦于参数优化和约束处理层面,未能从算法结构本质上解决局部搜索能力不强这一核心问题。

个体最优位置承载着粒子的历史经验信息,对维持种群多样性和提高算法收敛性能具有重要作用[16]。然而,现有 PSOGSA 算法在速度更新机制中仅保留了惯性引导、全局最优引导和 GSA 的引力搜索,完全忽略了 PSO 算法中个体认知学习这一核心组成部分,造成算法过度依赖全局最优信息,导致粒子的搜索多样性与开发能力不足。

针对上述问题,本文提出一种基于四重引导机制的改进 PSOGSA 算法 IPSOGSA,并应用于机械臂时间最优轨迹规划。该算法首次系统地将个体认知学习机制重新引入 PSOGSA 框架,构建包含惯性保持、个体学习、社会学习和引力搜索的四重引导机制,从算法结构层面解决现有算法的固有缺陷;通过 6 个基准测试函数和七自由度机械臂轨迹规划的系统对比实验,验证改进算法的性能。

1 机械臂轨迹规划方法与建模

机械臂轨迹规划的核心是在给定路径节点间实现平滑运动插补。本文采用关节空间轨迹规划方法,直接在关节角度域内进行插补,通过构建关节角与时间的函数关系描述机械臂运动状态。

1.1 机械臂运动学建模

本文选择珞石公司生产的 xMatePro7 为研究对象,机械臂模型如图 1 所示,该机械臂具有冗余自由度,能够实现复杂的空间运动任务。

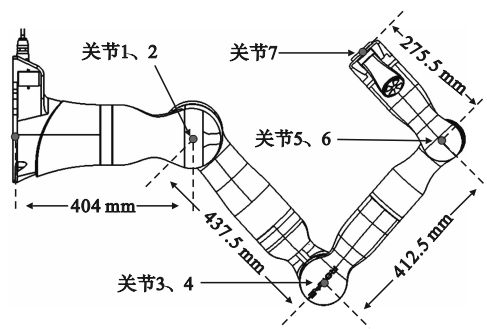


图 1 机械臂模型

Fig. 1 Manipulator Model

机械臂的 D-H 参数如表 1 所示,本文采用标准 D-H 参数法在 Matlab 机器人工具箱中建立运动学模型,并完成正运动学和逆运动学求解,为轨迹规划算法的实施奠定理论基础。

表 1　机械臂 D-H 参数表

Table 1　Maniplulator D-H parameters

关节	连杆长度/mm	连杆扭角/rad	连杆偏移/mm	关节转角/rad
1	0	$-\pi/2$	404	0
2	0	$\pi/2$	0	0
3	0	$-\pi/2$	437.5	0
4	0	$\pi/2$	0	0
5	0	$-\pi/2$	412.5	0
6	0	$\pi/2$	0	0
7	0	0	275.5	0

1.2　多项式插值轨迹规划方法

关节空间轨迹规划的目标是根据给定的路径节点信息,包括各关节的起始角度、终止角度、边界速度和加速度条件,结合关节运动约束条件,生成满足连续性要求的角度 – 时间轨迹序列。3-5-3 多项式插值方法是解决关节空间轨迹规划问题的常用方法。

3-5-3 多项式插值方法的基本思想是:在 3 次多项式轨迹的基础上选择两个插值节点,在节点间采用 5 次多项式进行插值,形成“3 次-5 次-5 次”的分段结构。这种设计既减少了计算量,又保证了轨迹的平滑性。3-5-3 多项式插值通式如式(1)所示,表示关节角度与时间的函数关系。

$$\begin{cases}\theta_1(t)=a_{10}+a_{11}t+a_{12}t^2+a_{13}t^3, t\in[0,t_1]\\ \theta_2(t)=a_{20}+a_{21}t+a_{22}t^2+\\ \qquad a_{23}t^3+a_{24}t^4+a_{25}t^5, \quad t\in[0,t_2]\\ \theta_3(t)=a_{30}+a_{31}t+a_{32}t^2+a_{33}t^3, t\in[0,t_3]\end{cases} \tag{1}$$

式中,$t_i(i=1,2,3)$为每段轨迹的运行时间;$a_{ij}(j=1,2,3)$为多项式中的系数。为便于计算,设定每段轨迹的初始时间为零。在机械臂的运动规划中,一般要求机械臂在起点和终点的加速度和速度为零,且为了使路径平滑且连续,需要使各阶段连接处的角度、速度、加速度相等。根据以上要求,建立约束方程如下,式(2)、(3)、(4)分别表示角度约束、速度约束与加速度约束。

$$\begin{cases}\theta_1(0)=\theta_0',\theta_1(t_1)=\theta_2(0)=\theta_1'\\ \theta_2(t_2)=\theta_3(0)=\theta_2',\theta_3(t_3)=\theta_3'\end{cases} \tag{2}$$

$$\begin{cases}\dot\theta_1(0)=0,\dot\theta_1(t_1)=\dot\theta_2(0)\\ \dot\theta_2(t_2)=\dot\theta_3(0),\dot\theta_3(t_3)=0\end{cases} \tag{3}$$

$$\begin{cases}\ddot\theta_1(0)=0,\ddot\theta_1(t_1)=\ddot\theta_2(0)\\ \ddot\theta_2(t_2)=\ddot\theta_3(0),\ddot\theta_3(t_3)=0\end{cases} \tag{4}$$

式中 θ_0'、θ_1'、θ_2'、θ_3'为 4 个路径点的角度,$\theta_i(t)$、$\dot\theta_i(t)$、$\ddot\theta_i(t)$分别表示各阶段的角度、角速度和角加速度。将此 14 个约束条件代入式(1)中即可构建线性方程组

$$\boldsymbol{A}\cdot\boldsymbol{a}=\boldsymbol{b} \tag{5}$$

式中

$$\boldsymbol{a}=[a_{13}\ a_{12}\ a_{11}\ a_{10}\ a_{25}\ a_{24}\ a_{23}\ a_{22}\ a_{21}\ a_{20}\ a_{33}\ a_{32}\ a_{31}\ a_{30}] \tag{6}$$

$$\boldsymbol{b}=[0\ 0\ 0\ 0\ 0\ 0\ \theta_3'\ 0\ 0\ \theta_0'\ 0\ 0\ \theta_1'\ \theta_2']^{\mathrm{T}} \tag{7}$$

$$\boldsymbol{A}=\begin{bmatrix}
t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\
6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & -1\\
0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & 20t_2^3 & 12t_2^2 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{8}$$

通过求解方程组即可得出系数矩阵中的各个系数与 θ_i、t_i 的关系,从而确定机械臂的运动轨迹。路径点角度值是在机械臂的工作空间中根据具体任务选取,而各阶段的运行时间在传统方法中通常使用相同的时长或者通过某种比例分配三个阶段的时长。这种静态分配方法导致轨迹运行时间较长,无法充分发挥机械臂性能,且往往需要多次试验才能得出满足速度和加速度约束的时间

分配方案。因此,采用 IPSOGSA 优化算法寻找最优时间分配方案对机械臂轨迹规划具有重要意义。

2 PSOGSA 算法理论基础与改进策略

2.1 PSO 与 PSOGSA 算法

PSO 通过个体最优位置(pbest)和全局最优位置(gbest)指导粒子更新速度和位置,其数学模型为

$$v_n^{k+1} = \omega v_n^k + c_1 r(p_n^k - x_n^k) + c_2 r(g^k - x_n^k) \quad (9)$$

$$x_n^{k+1} = x_n^k + v_n^{k+1} \quad (10)$$

式中:v_n^k、x_n^k、p_n^k、g^k 分别为粒子 n 在第 k 次迭代时的速度、位置、个体最优位置和全局最优位置;ω 为惯性权重;c_1、c_2 为学习因子;r 为[0,1]区间的随机数。

PSO 算法收敛速度快、参数少,但处理复杂函数时容易陷入局部最优,全局搜索能力不足。为此,PSOGSA 引入 GSA 算法增强全局搜索性能。

PSOGSA 混合算法[15]的速度更新公式为

$$v_n^{k+1} = \omega v_n^k + c_1 r a_n^k + c_2 r[g^k - x_n^k] \quad (11)$$

式中 a_n^k 为粒子 n 在第 k 次迭代时的引力加速度。根据牛顿运动第二定律,粒子 n 的加速度 a_n^k 及总引力 F_n^k 为

$$a_n^k = \frac{F_n^k}{M_n^k} \quad (12)$$

$$F_n^k = \sum_{m=1,m\neq i}^{N} r F_{nm}^k \quad (13)$$

$$F_{nm}^k = G^k \frac{M_n^k \times M_m^k}{R_{nm}^k + \varepsilon}[x_j^k - x_n^k] \quad (14)$$

式中:N 为粒子总数;F_{nm}^k 为在第 k 次迭代时粒子 m 对粒子 n 的引力;M_n^k、M_m^k 分别为粒子 n、m 的惯性质量;ε 为小常数;R_{nm}^k 为粒子间的欧氏距离;G^k 为引力常数

$$G^k = G_0 \exp\left(-\alpha \frac{k}{k_{\max}}\right) \quad (15)$$

式中:G_0 为初始引力常数;α 为衰减系数;$k_{\max}$ 为最大迭代次数。

2.2 改进的 PSOGSA 算法

传统 PSOGSA 算法仅保留 PSO 的全局最优引导机制,忽略了个体认知学习。个体最优项的重新引入能够提供多样化搜索方向,形成多层次搜索机制,提高算法稳定性。

IPSOGSA 速度更新公式为

$$v_n^{k+1} = \omega^k v_n^k + c_1^k r a_n^k + c_2^k r[g^k - x_n^k] + c_3^k r[p_n^k - x_n^k] \quad (16)$$

式中:各变量上标 k 均表示第 k 次迭代时的取值,ω^k 为惯性权重;c_1^k 为引力系数;c_2^k 为社会学习因子;c_3^k 为个体学习因子;p_n^k 为粒子 n 的个体最优位置。

为实现前期全局搜索与后期局部开发的平衡,设计了基于迭代进度的动态参数调整策略

$$\begin{cases} \omega^k = \omega_{\max} - (\omega_{\max} - \omega_{\min})\beta \\ c_1^k = c_{1\min} + (c_{1\max} - c_{1\min})\beta \\ c_2^k = c_{2\min} + (c_{2\max} - c_{2\min})\beta \\ c_3^k = c_{3\min} + (c_{3\max} - c_{3\min})\beta \end{cases} \quad (17)$$

式中:$\omega_{\max}$ 和 $\omega_{\min}$ 分别为惯性权重的最大值和最小值;$c_{i\max}$ 和 $c_{i\min}$ 分别为各系数的最大值和最小值;迭代进度因子 β 为

$$\beta = \frac{k}{k_{\max}} \quad (18)$$

前期采用较大惯性权重保持探索动量,较小个体学习因子避免早熟收敛,较大社会学习因子提供全局引导,较小引力系数防止过早聚集。后期逐步调整参数以增强局部开发能力。

当算法连续多次迭代无改进时,触发参数调整

$$\begin{cases} \omega^{k+1} = 0.5 w^k \\ c_1^{k+1} = 1.5 c_1^k \\ c_3^{k+1} = 1.2 c_3^k \end{cases} \quad (19)$$

通过减小惯性权重、增强个体学习和引力搜索激活停滞的粒子群,社会学习因子保持不变以维持对全局最优位置的稳定响应。

3 基于 IPSOGSA 算法的时间最优轨迹规划

3.1 时间最优化问题建模

时间最优轨迹规划的目标是在满足机械臂运动约束下,寻找使总执行时间最短的时间分配方案。考虑约束处理需求,构建包含惩罚项 P 的目标函数 F。

$$F = t_1 + t_2 + t_3 + P \quad (20)$$

式中 t_i 反映时间最优目标,惩罚项 P 确保解的物理可行性。

七自由度机械臂设计参数如表 2 所示,由此建立机械臂的约束体系。

表 2 机械臂关节约束

Table 2 Constraints of manipulator joints

关节	角度范围/°	最大角速度/ rad·s^{-1}	最大角加速度/ rad·s^{-2}
1、3、5	[-170, 170]	2.175/2.610/2.175	15/10/15
2、4、6	[-120, 120]	2.175/2.610/2.175	7.5/10/15
7	[-180, 180]	2.610	20

3.2　约束处理与适应度设计

本文采用大数惩罚法处理约束,该方法实现简单,计算开销小。设计的惩罚函数对每次约束违反累加固定惩罚值 P

$$P = P_0 \times (N_{pos} + N_{vel} + N_{acc}) \tag{21}$$

式中:$P_0 = 10^6$;N_{pos}、N_{vel}、N_{acc} 分别表示角度、角速度、角加速度约束违反次数。

设计基于解可行性状态的分层适应度计算机制。当轨迹完全满足约束($P = 0$)时,适应度函数简化为纯时间优化。

$$F = t_1 + t_2 + t_3 \tag{22}$$

当存在约束违反($P > 0$)时,适应度主要由违约程度决定。

$$F \approx P \tag{23}$$

这种分层策略使违约解在进化中处于劣势,形成从违约解向可行解的梯度引导,实现约束满足与目标优化的统一。

3.3　算法集成框架

算法集成的核心是建立粒子 – 轨迹映射机制,每个粒子 j 的位置向量 $\boldsymbol{x}_j = [t_1, t_2, t_3]$,直接对应一组时间分配方案。粒子群初始化采用随机方式

$$t_i = t_{min} + r(t_{max} - t_{min}), i = 1, 2, 3 \tag{24}$$

式中 t_{max} 和 t_{min} 分别为轨迹运行时间的最大值和最小值。

IPSOGSA 算法与 3-5-3 多项式插值法的集成框架如图 2 所示,首先由 3-5-3 多项式插值法根据路径点与粒子群中各粒子的位置生成轨迹;随后 IPSOGSA 算法根据各粒子位置产生的轨迹计算适应度并更新各粒子的位置;重复上述两个步骤直至粒子群停止迭代,输出最优时间参数。

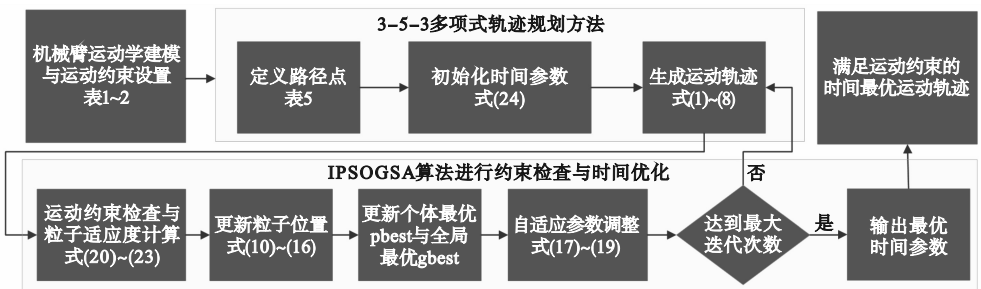


图 2　基于 IPSOGSA 的时间最优轨迹规划算法集成框架

Fig. 2　Algorithm integration framework for time-optimal trajectory planning based on IPSOGSA

4　算法对比与轨迹规划仿真实验

本章通过三层次实验评估 IPSOGSA 算法性能:基于 6 个基准函数评估改进效果;应用于七自由度机械臂轨迹规划验证实用性;通过轨迹生成和虚拟样机仿真验证算法集成的合理性和运动效率。实验在 Intel i7-12700K 平台的 Matlab R2023a 环境下完成。

4.1　算法性能对比与验证

本文选择三种算法与 IPSOGSA 进行性能对比:基础算法 PSOGSA,用于验证改进效果;基于突变的粒子群优化 – 引力搜索混合算法(MP-SOGSA)[14],用于对比不同改进策略;遗传算法(Genetic Algorithm,GA),用于验证其相对传统算法的优势。选择的经典优化测试函数如表 3 所示。

为确保对比实验的公平性,所有算法采用统一的基本参数:种群规模为 30,最大迭代次数为 1 000,其他参数采用对应文献中的设定值。PSOGSA 算法中 $\omega = 0.5$,$G_0 = 1$,$\alpha = 23$,引力系数 $c_1 = 0.5$,社会学习因子 $c_2 = 1.5$;MPSOGSA 算法中 $G_0 = 1$,引力系数 c_1 在 0 ~ 2 内递减,社会学习因子 c_2 在 0 ~ 2 内递增;IPSOGSA 算法中的参数设置尽量参考了 PSOGSA 算法的参数设置,其中 ω 在 0.5 ~ 0.9 内递减,$G_0 = 1$,$\alpha = 23$,引力系数 c_1 在 0.7 ~ 1.4 内递减,社会学习因子 c_2 在 1.2 ~ 1.7 内递减,个体学习因子 c_3 在 1.5 ~ 1.8 内递增。每个算法在 6 个测试函数上运行 10 次。

表 3　测试函数定义

Table 3　Test functions definition

函数	表达式
F_1	$F_1 = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
F_2	$F_2 = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$
F_3	$F_3 = \sum_{i=1}^{n} (x_i + 0.5)^2$
F_4	$F_4 = \sum_{i=1}^{n} (x_i^2 - i)^2$
F_5	$F_5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F_6	$F_6 = \sum_{i=1}^{n-1} (w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_n - 1)^2[1 + \sin^2(2\pi w_n)] + \sin^2(\pi w_1)$

注:x_i 为待优化参数,$w_i = 1 + (x_i - 1)/4$。

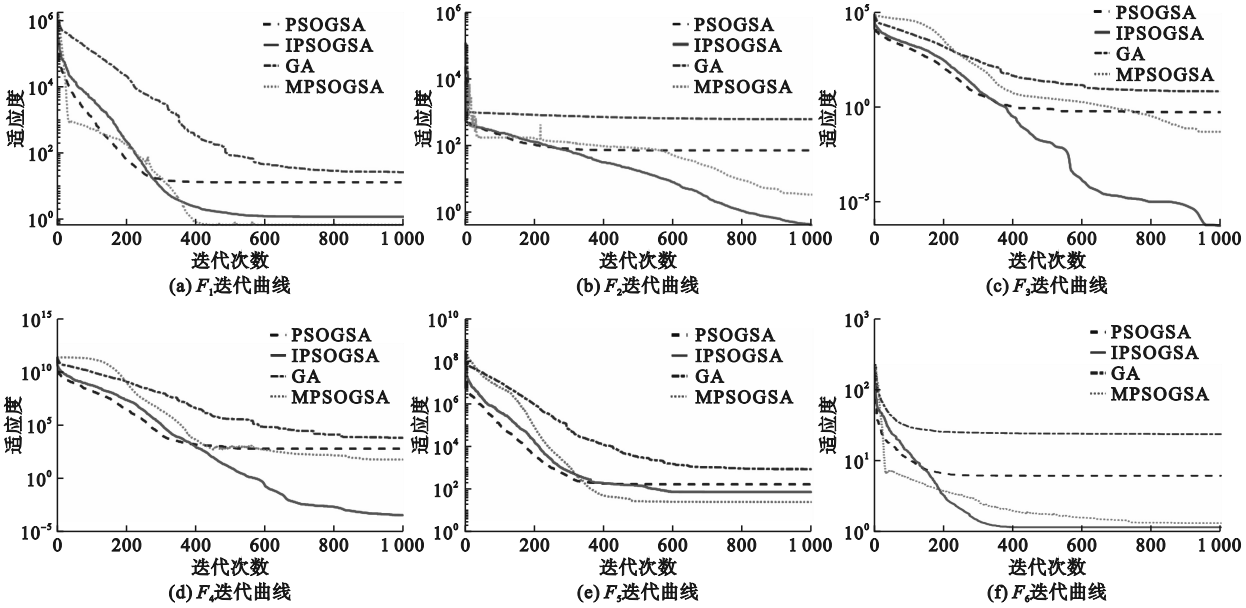


图 3 $F_1 \sim F_6$ 函数迭代曲线图

Fig. 3 Function iteration for $F_1 \sim F_6$

如图 3 所示,从整体上看,GA 算法收敛精度最低。IPSOGSA 算法相较于 PSOGSA 算法收敛精度提高了 1 ~ 6 个数量级。IPSOGSA 算法在 F_2、F_3、F_4 和 F_6 函数上获得最优精度,尤其在 F_3 和 F_4 函数上实现了数个数量级的性能提升。虽然在 F_1 和 F_5 函数上稍逊于 MPSOGSA 算法,但差异较小。改进算法在收敛后期仍能保持持续优化能力,体现了四重引导机制的优势。

表 4 为算法对测试函数的计算结果,GA 算法稳定性最差;IPSOGSA 算法相较于 PSOGSA 算法在全部 6 个函数上稳定性更好。IPSOGSA 算法在 F_3 和 F_4 函数上表现突出,相较于 PSOGSA 算法分别提升约 5 个和 6 个数量级。

尽管数值上相差约 6 个数量级,但由于两个结果都已处于较低水平且接近理论最优解,实际差距在函数整体搜索范围内并不显著。这种现象在优化问题中较为常见,当多个算法都能获得高质量解时,算法间的主要区别体现在数值精度上。

4.2 机械臂轨迹规划应用验证

本文将 IPSOGSA 等算法应用于七自由度机械臂时间最优轨迹规划问题。采用 1.1 节所述机械臂模型,设计四点路径规划任务,具体路径点设置如表 5 所示。

所有算法采用统一参数:种群规模 30,最大迭代次数 100,其他参数保持不变。t_1,t_2,t_3 的取值范围为 2 ~ 6 s,独立运行 30 次。四种算法在机械臂轨迹规划问题上的性能对比结果如图 4 与表 6 所示。

表 4 算法对测试函数的计算结果

Table 4 Computational results of algorithms on test functions

函数	算法	平均适应度	标准差
*F*1	PSOGSA	1.289×10^{1}	3.029×10^{1}
	MPSOGSA	6.667×10^{-1}	1.655×10^{-10}
	GA	2.608×10^{1}	1.765×10^{1}
	IPSOGSA	1.174×10^{0}	1.359×10^{0}
*F*2	PSOGSA	6.945×10^{1}	7.501×10^{1}
	MPSOGSA	3.346×10^{0}	6.352×10^{0}
	GA	5.863×10^{2}	1.239×10^{2}
	IPSOGSA	4.165×10^{-1}	6.820×10^{-1}
*F*3	PSOGSA	5.341×10^{-1}	4.854×10^{-1}
	MPSOGSA	5.013×10^{-2}	1.053×10^{-1}
	GA	6.705×10^{0}	3.752×10^{-0}
	IPSOGSA	5.761×10^{-7}	1.822×10^{-6}
*F*4	PSOGSA	5.510×10^{2}	4.766×10^{2}
	MPSOGSA	4.400×10^{1}	8.928×10^{1}
	GA	5.761×10^{3}	3.049×10^{3}
	IPSOGSA	3.334×10^{-4}	5.985×10^{-4}
*F*5	PSOGSA	1.736×10^{2}	7.637×10^{1}
	MPSOGSA	2.576×10^{1}	5.058×10^{-1}
	GA	9.026×10^{2}	6.916×10^{2}
	IPSOGSA	7.534×10^{1}	6.771×10^{1}
*F*6	PSOGSA	6.093×10^{0}	4.417×10^{0}
	MPSOGSA	1.303×10^{0}	2.559×10^{-1}
	GA	2.339×10^{1}	6.978×10^{0}
	IPSOGSA	1.144×10^{0}	1.094×10^{0}

表 5　轨迹规划路径点设置

Table 5　Trajectory planning waypoint settings

关节	起始点/rad	路径点 1/rad	路径点 2/rad	终点/rad
1	0	$-\pi/4$	$\pi/3$	0
2	0	$\pi/6$	$-\pi/4$	0
3	0	$-\pi/6$	$\pi/8$	0
4	0	$\pi/8$	$-\pi/6$	0
5	0	0	$\pi/6$	0
6	0	$\pi/12$	0	0
7	0	$-\pi/4$	$\pi/3$	0

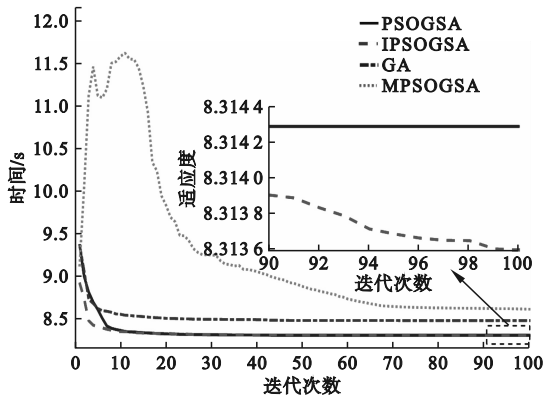


图 4　轨迹规划算法性能对比图

Fig. 4　Performance comparison graph for trajectory planning algorithm

图 4 适应度收敛曲线对比显示,IPSOGSA 算法表现出三个显著优势:收敛速度最快,在前 10 次迭代内迅速接近最优值;优化精度最高,最终稳定在最低值约 8.313 6 s;局部搜索能力最强,在后期仍保持持续优化趋势,体现了卓越的精细搜索能力。

表 6　轨迹规划算法性能对比

Table 6　Performance comparison chart of trajectory planning algorithm

算法名称	最优时间/s	平均时间/s	时间方差/s^2
PSOGSA	8.312 6	8.314 2	0.000 005
IPSOGSA	8.312 6	8.313 5	0.000 001
GA	8.318 7	8.486 9	0.026 974
MPSOGSA	8.312 7	8.619 2	0.050 031

由表 6 可见,IPSOGSA 算法在所有性能指标上均获得最佳结果。虽然各算法在最优执行时间上的差异相对较小,体现出与 F_3 函数测试中类似的现象,但算法在稳定性方面表现突出,时间方差为 0.000 001,相较于 PSOGSA 算法约提升 4 倍,相较于 MPSOGSA 算法约提升 50 000 倍,体现了四重引导机制对算法稳定性显著的改善效果。

4.3　轨迹生成与虚拟样机仿真

为进一步验证优化结果的合理性,本文基于表 6 的平均时间数据,生成各算法的最优时间分配方案对应的七自由度机械臂运动轨迹,并在 Matlab 环境下进行虚拟样机仿真。图 5 展示了 IPSOGSA 算法的关节运动轨迹,图 6 为各算法在虚拟样机仿真中的末端执行器运动轨迹。

从图 5 可见,所有关节平滑地从起始位置经过两个中间路径点到达终点,轨迹连续无突变。角速度和角加速度变化平滑,最大值分别为 2.175 rad·s^{-1}和 3.268 rad·s^{-1},满足机械臂运动约束,体现了 3-5-3 多项式轨迹的平滑特性。

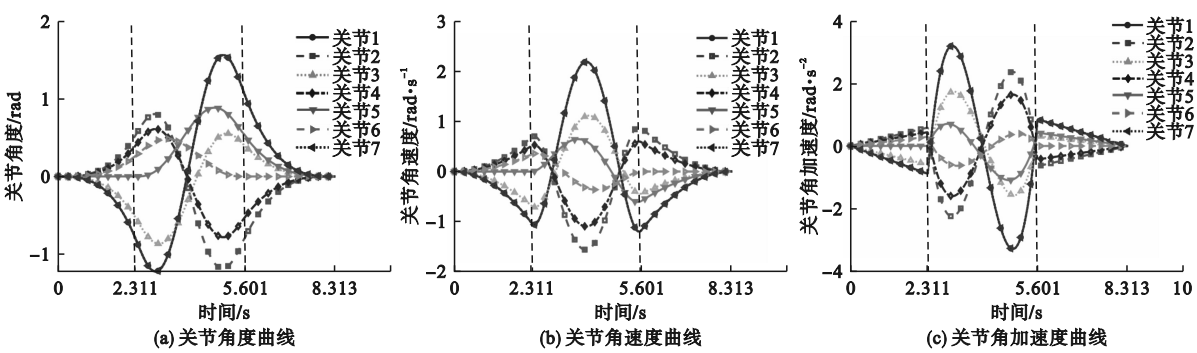


图 5　关节运动曲线

Fig. 5　Joint motion curves

图 6 展示了各算法对应的末端执行器运动轨迹,所有轨迹均经过设定的四个路径点且运动平滑连续。机械臂在本文设置的路径点下有两次较大的运动方向变化,在局部放大图中可以看到本文算法的轨迹在拐弯时拥有最大的拐弯半径,表现出最快的运动速度。经过计算,各算法轨迹曲线长度如下:PSOGSA 为 7.961 5 m;IPSOGSA 为 7.986 3 m;GA 为 7.797 8 m;MPSOGSA 为 7.798 6 m。结合表 6 数据,IPSOGSA 算法在实现最短执行时间的同时,轨迹长度相对较长,表现

出更高的平均运动速率。关节运动曲线和轨迹长度数据表明,该算法找到了更优的时间分配参数组合,使机械臂各关节运动更充分地利用了速度和加速度约束范围。

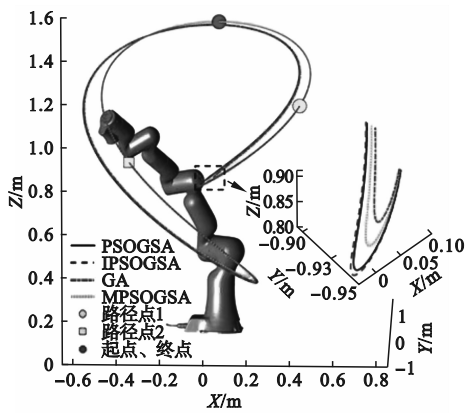


图 6 各算法运动轨迹对比

Fig. 6 Comparison of trajectory with different algorithms

5 结论

本文围绕机械臂时间最优轨迹规划问题,深入分析了传统 PSOGSA 算法的结构性缺陷,提出了基于四重引导机制的改进算法 IPSOGSA。通过首次系统地将个体最优学习项重新引入 PSOGSA 框架,构建四重引导机制,增强了算法搜索多样性和局部精细搜索能力;设计动态参数调整策略和停滞处理机制,提高了算法自适应能力和稳定性。基准测试函数实验表明,IPSOGSA 算法相较于 PSOGSA 算法收敛精度提升 1 ~ 6 个数量级,稳定性显著改善。在基于四路径点 3-5-3 多项式插值的七自由度机械臂轨迹规划实验中,稳定性相比传统算法提升了约 4 倍。虚拟样机仿真中,IPSOGSA 算法的轨迹表现出更高的运动效率。该算法找到了更优的时间分配参数组合,使机械臂各关节运动更充分地利用了速度和加速度约束范围。生成轨迹满足物理约束且运动平滑,为七自由度机械臂时间最优轨迹规划提供了有效解决方案。需要指出的是,本文重点关注算法优化精度和稳定性的理论改进,未来将重点探索计算效率优化、多目标优化和实时轨迹规划等。

参考文献(References):

[1] Haghshenas H, Hansson A, Norrl f M. Time-optimal path tracking for cooperative manipulators: a convex optimization approach [J]. Control Engineering Practice, 2023, 140: 105668.

[2] Liu J. Trajectory planning and control algorithm of industrial robot manipulator[J]. Journal of Vibroengineering, 2023, 25(8): 1516 – 1530.

[3] Gasparetto A, Zanotto V. A new method for smooth trajectory planning of robot manipulators[J]. Mechanism and Machine Theory, 2007, 42(4): 455 – 471.

[4] Hu Y H, Zhang S L, Chen Y H. Trajectory planning method of 6-DOF modular manipulator based on polynomial interpolation[J]. Journal of Computational Methods in Sciences and Engineering, 2023, 23(3): 1589 – 1600.

[5] 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.
Feng J C, Lu W Q, Lin J H, et al. Six-axis robot trajectory optimization based on 3 – 5 – 3 polynomial interpolation[J]. Machinery & Electronics, 2023, 41(10): 75 – 80. (in Chinese)

[6] Lu Z K, You Z C, Xia B H. Time optimal trajectory planning of robotic arm based on improved sand cat swarm optimization algorithm[J]. Applied Intelligence, 2025, 55(5): 323.

[7] Du Y X, Chen Y H. Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization[J]. Chinese Journal of Electronics, 2022, 31(5): 906 – 914.

[8] 隋涛,姜昊,孔刘君,等. 基于改进粒子群算法的机械臂轨迹规划研究[J]. 沈阳理工大学学报, 2023, 42(1): 7 – 12.
Sui T, Jiang H, Kong L J, et al. Research on manipulator trajectory planning based on improved particle swarm optimization algorithm[J]. Journal of ShenYang Ligong University, 2023, 42(1): 7 – 12. (in Chinese)

[9] 许家伟,李磊,汪建华,等. 基于 TCSPSO 算法的机械臂运动时间最优轨迹规划[J]. 现代制造工程, 2025(3): 69 – 76, 83.
Xu J W, Li L, Wang J H, et al. Time-optimal trajectory planning of manipulator based on TCSPSO algorithm[J]. Modern Manufacturing Engineering, 2025(3): 69 – 76, 83. (in Chinese)

[10] 李小为,胡立坤,王琥. 速度约束下 PSO 的六自由度机械臂时间最优轨迹规划[J]. 智能系统学报, 2015, 10(3): 393 – 398.
Li X W, Hu L K, Wang H. PSO-based time optimal trajectory planning for six degrees of freedom robot manipulators with speed constraints[J]. CAAI Transactions on Intelligent Systems, 2015, 10(3): 393 – 398. (in Chinese)

[11] Mirjalili S, Hashim S Z M. A new hybrid PSOGSA algorithm for function optimization[C]//2010 International Conference on Computer and Information Application. Tianjin, China: IEEE, 2010: 374 – 377.

[12] Ghorpade K, Khaparde A. Single-channel speech enhancement by PSO-GSA with harmonic regeneration noise reduction[J]. Bulletin of Electrical Engineering and Informatics, 2023, 12(5): 2895 – 2902.

[13] Shankar R, Ganesh N, ep R, et al. Hybridized particle swarm: gravitational search algorithm for process optimization[J]. Processes, 2022, 10(3): 616.

[14] Kar D, Ghosh M, Guha R, et al. Fuzzy mutation embedded hybrids of gravitational search and Particle Swarm Optimization methods for engineering design problems[J]. Engineering Applications of Artificial Intelligence, 2020, 95: 103847.

(下转第 69 页)