

DOI:10.19651/j.cnki.emt.2415809

# 基于改进 PSO-GSA 算法的时间最优轨迹规划

游达章<sup>1,2</sup> 赵恒毅<sup>1,2</sup> 汪传文<sup>3</sup>

(1.湖北工业大学机械工程学院 武汉 430068; 2.湖北省现代制造质量工程重点实验室 武汉 430068;  
3.湖北时瑞达重型工程机械有限公司 襄阳 441100)

**摘要:** 传统 3-5-3 多项式插值轨迹规划算法速度和加速度规划过于保守,与机械臂运动极限条件相差较远,没有充分发挥其运动性能,从而导致机械臂完成任务的时间增长。针对上述问题,本文提出一种基于改进 PSO-GSA 算法的 3-5-3 多项式插值轨迹规划算法。首先引入自适应惯性权重与动态学习因子对 PSO-GSA 算法进行改进,然后使用改进算法对 3-5-3 多项式插值算法进行时间优化。在优化过程中,关节速度超速时的粒子组使用了与不超速时不同的适应度函数,引导粒子组朝关节速度减小的方向靠拢,加快了算法收敛速度。仿真结果表明,改进的 PSO-GSA 算法相比原算法及一些同类算法收敛速度更快、搜索精度更高、不易陷入局部最优。对 3-5-3 多项式插值轨迹规划法进行时间优化后相比优化前运行时间缩短了 22.9%,得到的轨迹满足速度限制且平滑稳定,运行更加安全高效。

**关键词:** 轨迹规划;3-5-3 多项式插值;PSO-GSA;时间最优;六轴机械臂

**中图分类号:** TN102 **文献标识码:** A **国家标准学科分类代码:** 510.8050

## Time optimal trajectory planning based on improved PSO-GSA algorithm

You Dazhang<sup>1,2</sup> Zhao Hengyi<sup>1,2</sup> Wang Chuanwen<sup>3</sup>

(1. School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China;  
2. Hubei Key Lab of Manufacture Quality Engineering, Wuhan 430068, China;  
3. Hubei Shi Ruida Heavy Construction Machinery Co., Ltd., Xiangyang 441100, China)

**Abstract:** The speed and acceleration planning of the traditional 3-5-3 polynomial interpolation trajectory planning algorithm is too conservative, which is far from the motion limit conditions of the manipulator, and does not give full play to its motion performance, which leads to the increase of the time for the manipulator to complete the task. To solve the above problems, this paper proposes a 3-5-3 polynomial interpolation trajectory planning algorithm based on improved PSO-GSA algorithm. Firstly, the adaptive inertia weight and dynamic learning factor are introduced to improve the PSO-GSA algorithm, and then the improved algorithm is used to optimize the time of 3-5-3 polynomial interpolation algorithm. In the optimization process, the particle group with overspeed used a different fitness function than the one without overspeed, which led the particle group to move closer to the direction of the joint speed decrease, and accelerated the convergence speed of the algorithm. The simulation results show that the improved PSO-GSA algorithm has faster convergence speed, higher search accuracy and is not easy to fall into local optimum than the original algorithm and some similar algorithms. The running time of the 3-5-3 polynomial interpolation trajectory planning method is reduced by 22.9% compared with that before optimization. The obtained trajectory meets the speed limit and is smooth and stable, and the operation is safer and more efficient.

**Keywords:** trajectory planning;3-5-3 polynomial interpolation;PSO-GSA;time optimization;a six-axis robotic arm

## 0 引言

机械臂轨迹规划研究的是轴体在两点或多点间运动的插补问题,按在不同空间系中进行规划可分为笛卡尔空间轨迹规划和关节空间轨迹规划。笛卡尔空间轨迹规划能使

机械臂末端按用户指定的路径来运动,但需将所有插补点坐标逆解得到关节角,计算量巨大且会遇到机械臂机构奇异性问题;关节空间轨迹规划直接在连续的角度空间进行规划,解决的是不确定路径的点到点问题,计算量小且不存在奇异性问题。在实际使用中,可将确定的连续轨迹分为

若干份小段来重复使用点到点的关节空间规划,以规划确定的机械臂末端运动路径,因此点到点的关节空间轨迹规划更具有研究价值<sup>[1]</sup>。

近年来,针对机械臂关节空间轨迹规划问题,国内外学者做了大量的研究<sup>[2-4]</sup>。文献[5]使用三次多项式插值算法对机械臂进行关节空间轨迹规划,保证了机械臂能平稳的完成任务,但关节加速度存在跳变问题;文献[6]使用了五次多项式插值算法对机械臂进行了轨迹规划,解决了三次多项式插值算法的加速度跳变问题,但同时加大了算法运算复杂程度;文献[7]在三次多项式插值算法基础上引入两个时间节点,与五次多项式插值算法结合为 3-5-3 多项式插值算法,在保证关节加速度连续的同时加快了运算效率,但未能对两时间节点进行智能优化;文献[8]使用粒子群优化算法选择 3-5-3 多项式插值算法时间节点,实现了在运动限制条件内的时间最优轨迹规划,但该优化算法缺乏全局搜索能力,容易陷入局部最优;文献[9]和文献[10]分别使用改进粒子群算法和改进蝴蝶算法来选择 3-5-3 多项式插值算法时间节点,缩短了机械臂运行时间,但在运行过程中都将不满足运动约束条件的粒子直接淘汰,降低了算法的运行效率。

基于以上研究,为在机械臂平稳完成任务的前提下,尽可能缩短其运行时间,本文提出了一种改进的粒子群引力搜索混合算法 (particle swarm optimization-gravitational search algorithm, PSO-GSA)<sup>[11]</sup>,对 3-5-3 多项式插值规划算法的时间节点进行优化。算法在改进过程中引入自适应惯性权重与动态学习因子,加强了算法在前期的全局搜索能力和后期的局部搜索能力;在粒子组不满足机械臂运动约束条件时采用了和约束条件内不同的适应度函数,提高了算法运行效率;通过仿真验证,该方法能有效缩短机械臂运行时间。

## 1 机械臂轨迹规划

### 1.1 机械臂运动学模型

本文以一种六自由度机械臂作为研究对象,模型参数如图 1 所示。为了描述机械臂各个关节的坐标以结合轨迹规划,本文根据 D-H 参数<sup>[12]</sup>表对机械臂进行运动学建模,参数表如表 1 所示。

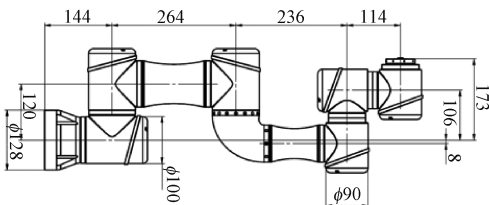


图 1 机械臂模型参数

Fig. 1 Mechanical arm model parameters

根据以上机械臂参数进行其正解、逆解等运动学分析,即可完成对机械臂的运动学建模。

表 1 机械臂 D-H 参数

Table 1 Robot arm D-H parameters

$i$	连杆转角 $\alpha_{i-1}$	杆长 $a_{i-1}$	连杆偏距 $d_i$	关节转角 $\theta_i$
1	0	0	144	$\theta_1(0)$
2	90	0	0	$\theta_2(-90)$
3	0	-264	0	$\theta_3(0)$
4	0	-236	106	$\theta_4(-90)$
5	90	0	114	$\theta_5(0)$
6	-90	0	67	$\theta_6(0)$

### 1.2 3-5-3 多项式插值方法

本文针对在关节空间中的点到点轨迹规划问题进行研究。即已知各关节起点与终点的角度、角速度、角加速度、时间点,结合关节角速度、角加速度的限制,规定插补周期,得到一条包含角度信息的路径点轨迹。

3-5-3 多项式插值方法是目前工业中一种常用的机械臂轨迹规划方法,它将三次和五次插值法结合,优化了三次多项式插值法加速度轨迹不连续造成的机械臂抖动问题,同时又优化了五次多项式插值法带来的计算量剧增问题。其思想是在三次多项式轨迹规划的路径中选取两个插值点,对两点进行五次多项式插值,即形成了三次、五次、三次多项式插值的三段式结构,分段图如图 2 所示:

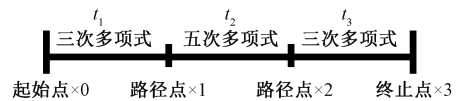


图 2 分段示意图

Fig. 2 Sectional diagram

已知 3 段的时长和 4 个点的角度值,则 3-5-3 多项式插值的通式为

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

其中,  $t_1, t_2, t_3$  为每一段运行的时间。为了路径平滑连续,需要保证在多项式的分段处角度、角速度、角加速度前后相等,即添加以下约束条件

$$\begin{cases} \theta_1(t_1) = \theta_2(0), \theta_2(t_2) = \theta_3(0) \\ \dot{\theta}_1(t_1) = \dot{\theta}_2(0), \dot{\theta}_2(t_2) = \dot{\theta}_3(0) \\ \ddot{\theta}_1(t_1) = \ddot{\theta}_2(0), \ddot{\theta}_2(t_2) = \ddot{\theta}_3(0) \end{cases} \quad (2)$$

将约束条件代入通式中能得到下列关系,即可求得通式中的系数矩阵  $a$ , 其中  $\theta_i, \dot{\theta}_i, \ddot{\theta}_i$  为 4 个路径点的角度、角速度、角加速度。

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

$$b = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \theta_3 \ 0 \ 0 \ \theta_0 \ 0 \ 0 \ \theta_2 \ \theta_1]^T \quad (4)$$

$$a = A^{-1} \cdot b = [a_{13} \ a_{12} \ a_{11} \ a_{10} \ a_{25} \ a_{24} \ a_{23} \ a_{22} \ a_{21} \ a_{20} \ a_{33} \ a_{32} \ a_{31} \ a_{30}]^T \quad (5)$$

传统工业中使用 3-5-3 多项式轨迹规划方法时, 一般人为定义 3 段运行时间相同, 或者根据人工经验来确定 3 段的运行时间, 而这种方法通常得到的运行速度较为保守, 任务执行时间较长, 没有充分发挥机械臂的运动性能, 而且在执行不同任务时, 需要制定不同的时间选择方案, 缺乏便利性。显然, 使用智能优化算法来确定 3 段的运行时间, 能较好解决上述轨迹规划算法的缺陷, 势在必行。

## 2 PSO-GSA 智能优化算法

### 2.1 PSO-GSA 算法原理

粒子群优化算法 (particle swarm optimization, PSO)<sup>[13-14]</sup> 模拟了鸟群或鱼群寻找食物时的行为, 通过模拟每个粒子在解空间中的飞行, 以及个体间信息的共享来搜索最优解。PSO 的基本工作原理是不断地更新每个粒子的速度和位置, 使其朝着局部最优和全局最优的方向移动。通过迭代优化过程, 粒子群逐渐收敛于最优解。引力搜索算法 (gravitational search algorithm, GSA)<sup>[15-16]</sup> 是一种基于自然现象中物体间引力和质量之间相互作用的启发式优化算法。GSA 的基本工作原理是不断更新引力加速度和位置, 引力加速度用于计算每个解受到的引力作用, 然后根据引力加速度来更新解的位置。通过迭代这两步操作, GSA 逐渐收敛于最优解。

PSO 算法参数少、收敛快, 可以有效解决复杂的高维非线性问题, 但是其寻优结果与种群初始化有很大关系, 缺乏全局的搜索能力, 容易陷入局部最优。GSA 算法具有较强的全局搜索能力, 但由于粒子间的相互吸引, 容易在局部震荡收敛较慢, 且没有储存之前最优解的步骤。Mirjalili 等人提出一种 PSO 和 GSA 的融合算法, 改善了 PSO 算法容易陷入局部最优的问题, 弥补了 GSA 算法中缺乏信息交

换与记忆的问题, 加快了收敛速度, 算法性能得到了显著提升, 其算法解析如下。

万有引力定律的公式为:

$$F = G \frac{M_1 M_2}{R^2} \quad (6)$$

式中:  $G$  为引力常数,  $R$  为两天体间的距离,  $M$  为天体的质量。在 PSO-GSA 中, 对万有引力公式进行了改写, 得到粒子  $i$  和粒子  $j$  间的引力公式为:

$$F_{ij}(t) = G(t) \frac{M_i(t)M_j(t)}{R_{ij}(t) + \epsilon} [x_j(t) - x_i(t)] \quad (7)$$

其中,  $M_i(t)$  和  $M_j(t)$  为第  $i, j$  个粒子的质量,  $R_{ij}(t)$  为两个粒子之间的距离,  $\epsilon$  为常数,  $x_j(t)$  和  $x_i(t)$  为两粒子在  $t$  时刻的位置。引力常数  $G(t)$  随着时间变化进行以下更新:

$$G(t) = G_0 \exp(-\alpha \times \frac{t}{t_{\max}}) \quad (8)$$

其中,  $G_0$  为初始化值,  $t$  为迭代次数,  $t_{\max}$  为最大迭代次数,  $\alpha$  为减小函数系数。

每个粒子间的引力相互作用, 则由式(8)可推出每个粒子受到的合力为:

$$F_i(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}(t) \quad (9)$$

其中,  $rand$  为  $0 \sim 1$  之间的随机值。得到粒子所受合力后, 进而可得到加速度更新公式为:

$$a_i(t) = \frac{F_i(t)}{M_i(t)} \quad (10)$$

其中,  $a_i(t)$  为粒子的加速度, 分母第  $i$  个粒子的质量更新公式为:

$$m_i(t) = \frac{fit_i(t) - fitMax(t)}{fitMin(t) - fitMax(t)} \quad (11)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (12)$$

其中,  $fit_i(t)$  为粒子  $i$  在  $t$  时刻的适应度,  $fitMax(t)$  和  $fitMin(t)$  为  $t$  时刻所有粒子适应度的最大值和最小值。最后对各粒子的速度和位置进行如下更新:

$$v(t+1) = \omega \times v_i(t) + r_1 \times rand_{1i} \times a_i(t) + r_2 rand_{2i} \times [gbest - x_i(t)] \quad (13)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (14)$$

其中,  $v$  为粒子的速度,  $\omega$  为惯性权重,  $r_1$  和  $r_2$  为加速度常数,  $x$  为粒子的位置,  $gbest$  为粒子群目前的最好位置。

基于以上公式, PSO-GSA 算法运行的步骤为:

步骤 1) 对算法中的粒子进行初始化, 设定种群数量、维度、迭代次数, 确定合适的重力常数  $G_0$ 、减小函数系数  $\alpha$ 、加速度常数  $r_1$  和  $r_2$ 、惯性权重  $\omega$ 。

步骤 2) 评估并记录粒子当前适应度, 更新  $gbest$ 、 $fitMax$ 、 $fitMin$ 。

步骤 3) 依据式 (3) 更新引力常数。

步骤 4) 依据式 (2)、(4) 更新粒子所受合力。

步骤 5) 依据式 (6)、(7) 更新粒子质量。

步骤 6) 依据式 (5)、(8)、(9) 依次更新粒子的加速度、速度、位置。

步骤 7) 判断是否达到迭代结束条件或达到最大迭代次数。是, 则输出最优值; 否, 则返回步骤 2) 继续运行。

## 2.2 改进的 PSO-GSA 算法

基于 PSO-GSA 算法的原理, 为了能够在迭代初期增强算法的全局搜索能力, 在后期增强算法的局部搜索能力, 考虑到粒子迭代过程中的迭代次数和迭代目标, 本文引入了自适应惯性权重与动态调整学习因子的方法对 PSO-GSA 算法进行改进。对于公式 8, 本文重新定义惯性因子  $\omega$  以及学习因子  $r_1(t)$  和  $r_2(t)$  如下:

$$\omega = \begin{cases} \omega_{\max} - (\omega_{\max} - \omega_{\min}) \frac{t}{T_{\max}}, & \text{fitMin}(t) < \text{fitAvg}(t) \\ \omega_{\max}, & \text{fitMin}(t) \geq \text{fitAvg}(t) \end{cases} \quad (15)$$

$$r_1(t) = r_{1\text{init}} + (r_{1\text{final}} - r_{1\text{init}}) \frac{t}{T_{\max}} \quad (16)$$

$$r_2(t) = r_{2\text{init}} + (r_{2\text{final}} - r_{2\text{init}}) \frac{t}{T_{\max}} \quad (17)$$

其中,  $\omega_{\max}$ 、 $\omega_{\min}$  为惯性因子最大值和最小值,  $r_{1\text{init}}$  和  $r_{1\text{final}}$  为学习因子的初始值和最终值。  $t$  和  $T_{\max}$  为当前迭代次数和最大迭代次数。  $\text{fitMin}(t)$  为当前一代最优值,  $\text{fitAvg}(t)$  为历代最优结果的平均值, 自适应  $\omega$  在搜索初期增大增加全局搜索能力, 在最优值附近搜索时减小  $\omega$  增强局部搜索能力。动态调整  $r_1(t)$  和  $r_2(t)$ , 随着算法迭代线性增加, 同样使算法从开始的强化全局搜索逐渐转变为后期的强化局部最优搜索。

基于以上改变, 增加了惯性因子和学习因子的更新步骤, 改进后的 PSO-GSA 算法流程图如图 3 所示:

## 3 基于时间最优的轨迹规划

由 1.2 节的分析可知, 传统 3-5-3 多项式轨迹规划方法需要由人工经验确定 3 段时长, 且难以充分发挥机械臂运动性能以提升工作效率。为解决上述缺陷, 本文使用改进的 PSO-GSA 算法来对 3-5-3 多项式轨迹规划方法的参数进行智能优化。

本文的优化目标是缩短机械臂轨迹的运动时间, 因此种群的维度采用 3 维, 粒子位置代表每段的运行时间, 适应度函数为 3 段时间的总和即:

$$\text{fit} = t_1 + t_2 + t_3 \quad (18)$$

机械臂在实际使用中, 关节的最大角速度一定是受限的, 而已知角速度为角度对时间的导数, 因此增加速度限制条件:

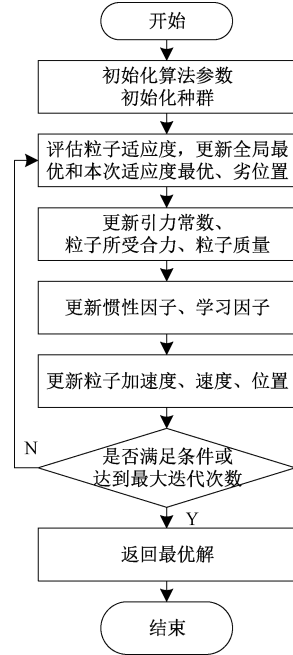


图 3 改进 PSO-GSA 算法流程图

Fig. 3 Flowchart of the improved PSO-GSA

$$\begin{cases} v_1(t) = a_{11} + a_{12}t^2 + a_{13}t^3, t \in [0, t_1] \\ v_2(t) = a_{21} + a_{22}t^2 + a_{23}t^3 + a_{24}t^4 + a_{25}t^5, t \in [0, t_2] \\ v_3(t) = a_{31} + a_{32}t^2 + a_{33}t^3, t \in [0, t_3] \end{cases} \quad (19)$$

$$\begin{cases} |v_1(t)| < V_{\max} \\ |v_2(t)| < V_{\max} \\ |v_3(t)| < V_{\max} \end{cases} \quad (20)$$

其中,  $v_1, v_2, v_3$  为关节在每段的速度,  $V_{\max}$  为关节最大速度限制。在传统的轨迹优化步骤中, 如果粒子组造成关节运动速度超过最大速度限制, 通常选择淘汰掉粒子组, 即直接将其适应度设置为无穷大。但粒子组得到的关节速度曲线如果只是轻微超速, 其结果可能十分靠近理想的最优解, 即时间最短。因此本文采用了引导超速关节粒子组朝关节减速方向靠拢的策略, 关节速度超速的粒子组使用了与未超速时不同的适应度函数:

$$\text{fit} = \frac{\text{MAX}[|v_{1\max}|, |v_{2\max}|, |v_{3\max}|]}{V_{\max}} \times g_{\text{best}}, (t \neq 1) \quad (21)$$

其中,  $g_{\text{best}}$  为当前的全局最优结果,  $t$  为当前迭代次数,  $v_{i\max}$  为第  $i$  段的最大速度, 则分子为全段的最大速度。即此时适应度函数为当前全局最优结果的动态倍数, 该动态倍数始终大于 1, 且随着超速程度的减小而减小。即不会取代全局最优位置, 而超速程度越大则会离最优解越远。下一次循环粒子会往速度减小的方向进化, 如果得到的结果满足了速度限制条件且适应度最优, 则使得该粒子组位置成为了当前最优位置。

因此, 智能优化算法与 3-5-3 多项式轨迹规划方法相



融合后增加了速度限制和适应度函数分段的步骤,则整体流程如图 4 所示。

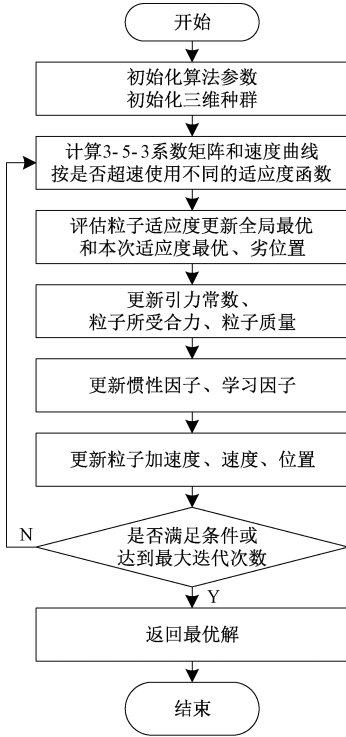


图 4 时间最优轨迹规划算法流程图

Fig. 4 Flowchart of the time-optimal trajectory planning algorithm

## 4 仿真实验与分析

### 4.1 改进的 PSO-GSA 算法性能分析

为验证改进算法的有效性,本文选择 CEC23(国际进化计算会议)测试函数中的 6 个经典函数,分别使用 PSO-GSA、改进的 PSO-GSA、鲸鱼算法 (whale optimization algorithm, WOA)、蜻蜓算法 (dragonfly algorithm, DA) 来对他们进行优化对比。所使用的基准函数如表 2 所示:

4 种算法的种群数量均为 30,最大迭代次数均为 500。PSO-GSA 中  $\omega = 0.5, r_1 = 0.5, r_2 = 1.5, G_0 = 1, \alpha = 23$ ; 改进的 PSO-GSA 中  $\omega_{\max} = 0.8, \omega_{\min} = 0.1, r_{1init} = 0.1, r_{1fnt} = 0.5, r_{2init} = 0.5, r_{2fnt} = 1.5, G_0 = 1, \alpha = 23$ 。每个算法优化每个基准函数 20 次并记录数据,表 3 为数据的分析结果,选取最优解和标准差两个指标进行分析,图 5 为运行结果最接近平均值时各函数中各算法的适应度曲线。其中,最优解越小则算法的寻优精度越高,曲线下降的越早则算法收敛的越快,标准差越小则算法越不容易陷入局部最优。

从表 3 中可以看到改进的 PSO-GSA 算法在各函数中的最优解均为最小,除 F2 和 F5 外标准差均为最小值。在图 5 中各测试函数下,PSO-GSA 算法的曲线下降时间相对较早。在 F2 和 F5 中虽标准差不为最小,但最优解明显优于其他算法。因此,改进的 PSO-GSA 算法有较好的优化精度和较快的收敛速度,不易陷入局部最优,具有较好的稳

表 2 测试函数表

Table 2 Test function table

函数公式	维度	搜索范围
$F_1(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65.536, 65.536]$
$F_2(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$
$F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$
$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	30	$[-100, 100]$
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$
$F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	2	$[-100, 100]$

定性。相比原算法各指标均有提升,在复杂函数场景下能够有效的进行寻优。

### 4.2 时间最优轨迹优化仿真与性能分析

本文以第一章中的 6 自由度机械臂为实验对象进行算法仿真验证,仿真实验平台为 Intel(R) Core(TM) i5-

6500 CPU@3.20 GHz 3.19 GHz, WIN10, 64 位操作系统。使用 MATLAB R2016a 进行仿真。

首先给出每轴 4 个路径点,位置如表 4 所示,设定每轴最大速度不超过  $80^\circ/\text{s}$ ,然后采用改进 PSO-GSA 算法优化的 3-5-3 轨迹优化方法分别对每轴轨迹进行规划,种群数

表 3 函数测试结果  
Table 3 Function test results

函数	算法	最优解	标准差	函数	算法	最优解	标准差
$F_1$	改进 PSO-GSA	1	0.503	$F_4$	改进 PSO-GSA	$3.48 \times 10^{-5}$	$3.34 \times 10^{-3}$
	PSO-GSA	1.999	2.764		PSO-GSA	$6.00 \times 10^{-4}$	4.072
	WOA	1	3.894		WOA	17.447	23.184
	DA	1	0.525		DA	0.725	1.823
$F_2$	改进 PSO-GSA	-12 560	2 713	$F_5$	改进 PSO-GSA	1.842	34.492
	PSO-GSA	-3 084	433		PSO-GSA	3.599	68.898
	WOA	-12 320	3 343		WOA	27.802	0.328
	DA	-3 178	425		DA	31.468	2325.684
$F_3$	改进 PSO-GSA	$1.09 \times 10^{-10}$	$6.23 \times 10^{-6}$	$F_6$	改进 PSO-GSA	$9.11 \times 10^{-22}$	$4.02 \times 10^{-22}$
	PSO-GSA	$3.54 \times 10^{-8}$	2 106		PSO-GSA	$1.03 \times 10^{-20}$	$4.54 \times 10^{-21}$
	WOA	15 769	11 774		WOA	0.174	0.294
	DA	9.15	318		DA	$1.71 \times 10^{-4}$	20.607

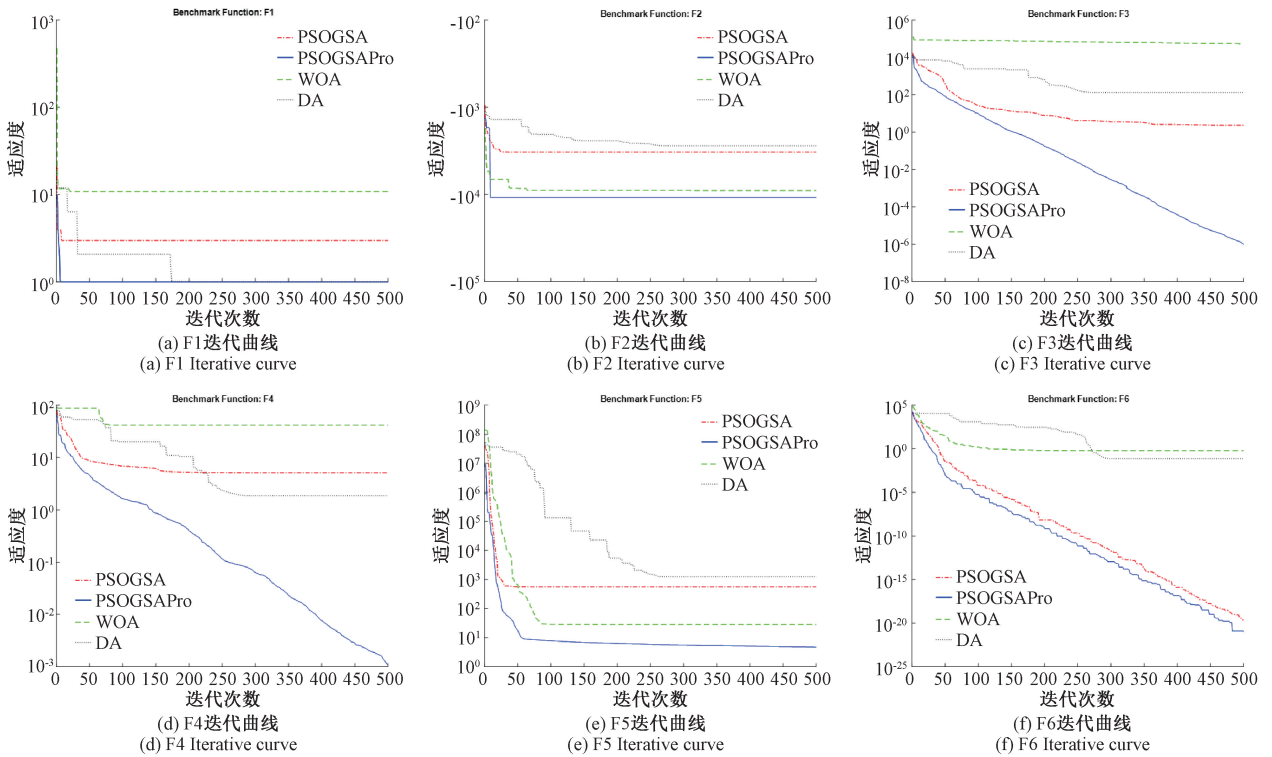


图 5 F1~F6 函数迭代曲线图

Fig. 5 Graph of the function iteration for F1~F6

表 4 路径点位置表

Table 4 Table of waypoint locations (°)

关节 i	起始角度	路径角度 1	路径角度 2	终止角度
1	-30	30	15	40
2	60	90	100	120
3	60	30	0	-60
4	-15	15	-15	15
5	0	100	60	30
6	-15	60	15	55

为 30, 迭代次数为 100, 得到 6 组运动轨迹。6 组轨迹数据中最长的运行时间即为机械臂整体的运行时间, 其迭代曲线如图 6 所示。

为验证该方法能够更好的完成机械臂运行时间优化的任务, 分别对改进 PSO-GSA 算法、原 PSO-GSA 算法、不经过智能优化的结果进行对比, 每种情况运行并记录 20 次, 设最优运行时间和平均运行时间两个对比指标。其结果如表 5 所示, 可以看到相比原算法, 改进后的平均时

表 5 机械臂运行时间对比表

算法	最优时间	平均时间
改进 PSO-GSA	9.436 9	9.437 5
PSO-GSA	9.437 1	10.594 3
不做优化处理	12.24	12.24

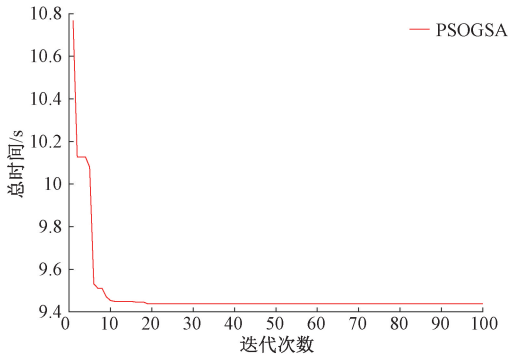


图 6 机械臂运行时间迭代曲线图

Fig. 6 Iterated plot of the running time of the manipulator

间明显减少,相比不做优化处理的情况下,平均时间减少了 22.9%,说明改进后的算法优化效果显著。

对于优化出的 6 组数据,本文选取出 3-5-3 插值法每

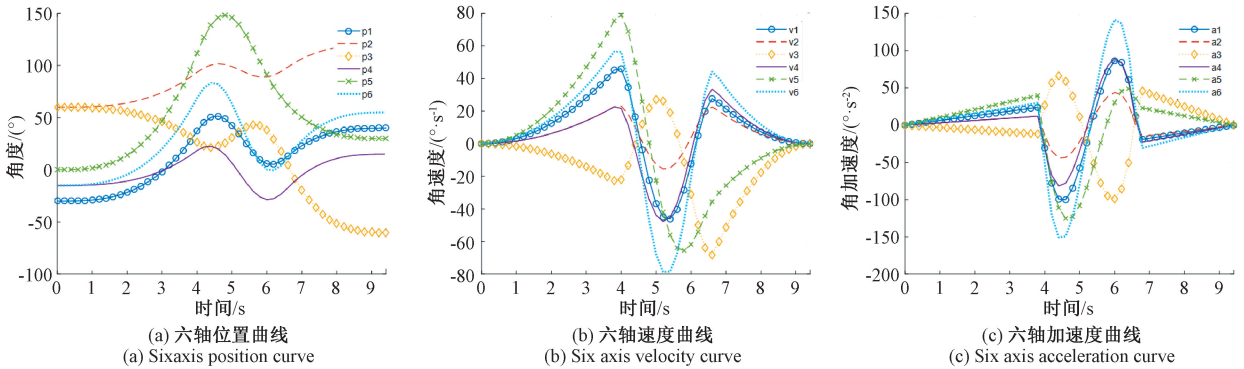


图 7 六轴运动曲线图

Fig. 7 Motion graph of six axes

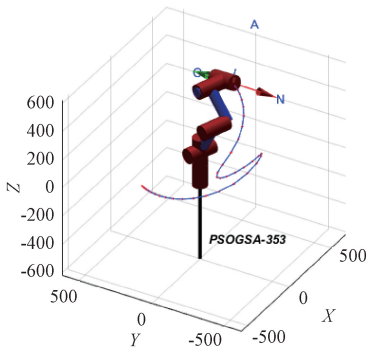


图 8 机械臂运行轨迹图

Fig. 8 Robot arm running track diagram

## 5 结 论

本文使用改进的 PSO-GSA 算法对 3-5-3 多项式插值轨迹规划算法进行时间优化。通过引入自适应惯性权重与动态学习因子的方法提高算法优化性能。在对 3-5-3 多项式插值轨迹规划算法进行优化时,根据粒子组是否满足运动限制条件,对适应度函数分类讨论,以加快算法收敛

段的最长时间组成一组 3-5-3 插值时间,得到的位置、速度、加速度曲线如图 7 所示,可以看到六轴的位置均通过表 4 设定的位置点,六轴速度和加速度均未发生跳变,各轴速度均在控制范围内且充分发挥了机械臂的运动性能。最后使用 MATLAB 机器人工具箱按第一章中的机器人参数进行模型建立,并输入得到的插补点序列以运行轨迹,其路径如图 8 所示,可以看到所得到的路径平滑,机械臂能够在缩短运行时间的情况下较好的完成工作任务。

速度。仿真结果表明,改进的 PSO-GSA 算法相比原算法及一些同类算法收敛速度更快、搜索精度更高、不易陷入局部最优。对 3-5-3 多项式插值轨迹优化算法优化后,机械臂能在平稳完成工作任务的同时缩短运行时间。

在未来的工作中,将会聚焦于完成多目标的轨迹优化,同时将运行时间与运动冲击作为优化目标,使机械臂更好的完成工作任务。

## 参考文献

[1] 蔡自兴,谢斌. 机器人学(第四版)[M]. 北京:清华大学出版社,2022:265-277.  
CAI Z X, XIE B. Robotics(4th Edition)[M]. Beijing: Tsinghua University Press, 2022:265-277.

[2] MA CH Y, ZHANG Y, CHENG SH. Time optimal trajectory planning based on redundant manipulator[C]. Shanghai: 2021 6th International Conference on Control, Robotics and Cybernetics (CRC), IEEE, 2021.

[3] WANG T, YAO SH D, ZHU SH Q. Energy-saving trajectory optimization of a fluidic soft robotic arm[J].

- Smart Materials and Structures, 2022,31(11):115011.
- [4] 游达章,杨智杰,张业鹏. 基于改进 RRT-Connect 算法的机械臂运动规划[J]. 电子测量技术, 2023, 46(8): 112-119.  
YOU D ZH, YANG ZH J, ZHANG Y P. Motion planning of robot arm based on improved RT-CONNECT algorithm [J]. Electronic Measurement Technology, 2019, 46(8): 112-119.
- [5] SABARIGIRISH S, MIJA S J. Obstacle avoiding trajectory planning for 5 degree of freedom robot[J]. 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems, 2016:1-5, DOI:10.1109/ICPEICES.2016.7853202.
- [6] FANG S, MA X, ZHAO Y, ZHANG Q, LI Y. Trajectory planning for seven-DOF robotic arm based on quintic polynormial [J]. 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2019(2): 198-201.
- [7] 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 CH, LU W Q, LIN J H. Trajectory optimization of six-axis robot based on 3-5-3 polynomial interpolation [J]. Machinery & Electronics, 2019, 41(10): 75-80.
- [8] 付荣,居鹤华. 基于粒子群优化的时间最优机械臂轨迹规划算法[J]. 信息与控制, 2011, 40(6): 802-808.  
FU R, JU H H. Trajectory planning algorithm for time-optimal manipulator based on particle swarm optimization [J]. Information and Control, 2011, 40(6): 802-808.
- [9] 李虎,刘泓滨. 基于改进 PSO 算法的时间最优机械臂轨迹规划[J]. 组合机床与自动化加工技术, 2023(1): 29-33.  
LI H, LIU H B. Trajectory planning of time-optimal manipulator based on improved PSO algorithm [J]. Modular Machine Tool & Automatic Machining Technology, 2023(1): 29-33.
- [10] 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.  
ZHOU M Y, ZHOU M W, LIU G Q. Time optimal trajectory planning of robotic arm based on improved butterfly algorithm [J]. Computer Science, 2023, 50(S2): 119-126.
- [11] BEIGYAND S D, ABDI H, LA SCALA M. Optimal operation of multicarrier energy systems using time varying acceleration coefficient gravitational search algorithm [J]. Energy, 2016, 114: 253-265.
- [12] 郭发勇,梅涛,赵江海. D-H 法建立连杆坐标系存在的问题及改进[J]. 中国机械工程, 2014, 25(20): 2710-2714.  
GUO F Y, MEI T, ZHAO J H. Problems and improvements in establishing connecting rod coordinate system with D-H method [J]. China Mechanical Engineering, 2014, 25(20): 2710-2714.
- [13] 孙燕成,陈富安. 基于 PSO 优化 LSTM 神经网络的机械臂逆运动学求解研究[J]. 电子测量技术, 2022, 45(13): 40-45.  
SUN Y CH, CHEN F AN. Research on inverse kinematics solution of robotic arm based on PSO-optimized LSTM neural network [J]. Electronic Measurement Technology, 2022, 45(13): 40-45.
- [14] 李鹏. 基于改进 PSO-BP 算法的机器人目标位姿识别方法[J]. 国外电子测量技术, 2023, 42(1): 7-12.  
LI P. Based on improved PSO and BP algorithm of robot pose target recognition method [J]. Electronic Measurement Technology Abroad, 2023, 42(1): 7-12.
- [15] AVALOS O. GSA for machine learning problems: A comprehensive overview [J]. Applied Mathematical Modelling, 2021, 92: 261-280.
- [16] 赵东升,李艳军. 基于 BGSA 算法的 SVM 分类模型设计研究[J]. 电子测量技术, 2019, 42(5): 56-59.  
ZHAO D SH, LI Y J. The SVM classification model based on BGSA algorithm design study [J]. Electronic Measurement Technology, 2019, 42(5): 56-59.

### 作者简介

**游达章**, 教授, 博士, 主要研究方向为机器人与智能控制、故障预测与可靠性技术。

E-mail: 2052563465@qq.com

**赵恒毅**(通信作者), 硕士研究生, 主要研究方向为机器人控制。

E-mail: 459151311@qq.com