

文章编号: 2095-2163(2024)03-0223-05

中图分类号: TP391.41

文献标志码: A

基于多策略差分进化算法的机械臂末端轨迹规划

周佳颖, 黄俊超, 赵宇凡, 宋泽凯

(上海工程技术大学 机械与汽车工程学院, 上海 201620)

摘要: 针对机械臂末端执行器轨迹规划问题, 本文提出一种基于多策略差分进化算法的机械臂末端轨迹规划。在基本差分进化算法的基础上, 引入变异策略和控制参数的候选集合, 同时在进化过程中根据进化信息实时从候选集合中选择合适的变异策略和控制参数。以机械臂执行时间为优化目标, 结合实时速度与加速度约束限制, 采用五次多项式插值方法建立关节空间轨迹规划数学模型, 采用多策略差分进化算法求解该轨迹规划数学模型。实验结果表明, 相比基本差分进化算法, 多策略差分进化算法能在较短时间内完成操作, 同时能够使机械臂末端执行器的运行轨迹平滑, 是一种有效的轨迹规划方法。

关键词: 机械臂; 轨迹规划; 差分进化算法; 多策略; 最优时间

Trajectory planning of the manipulator using a multi-strategy-based differential evolution

ZHOU Jiaying, HUANG Junchao, ZHAO Yufan, SONG Zekai

(School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: To solve the trajectory planning problem of the manipulator, this paper proposes an optimal time planning algorithm based on multi-strategy differential evolution algorithm. Firstly, a candidate pool of mutation strategies and control parameters are incorporated into differential evolution algorithm. At each iteration, some appropriate mutation strategies and control parameters are selected from these pools for creating aggressive solutions. The planning problem is posed to minimize the travelling time subjected to motion (velocity, acceleration and jerk) constraints and the five order polynomial function is used in order to get a smooth trajectory. The planning problem is solved using the multi-strategy differential evolution algorithm. The experimental results show that, compared with the basic differential evolution algorithm, the multi-strategy differential evolution algorithm got shorter moving time with smooth travelling path. It is an effective method to solving trajectory planning problems.

Key words: manipulator; trajectory planning; differential evolution algorithm; multi-strategy; optimal time

0 引言

多关节机械臂是一种可以模拟人类手臂运动的工业机器人, 具有动作灵活、通用性强等特点, 广泛应用于工业生产、医疗服务等领域。为了保证机械臂作业过程的快速性和平稳性, 机械臂末端轨迹的轨迹规划成为重要研究内容。末端执行器轨迹规划是指根据任务要求, 在满足规定的约束条件下, 完成从起始点到终止点的移动过程, 具有难以高效求解等特点。

智能优化算法凭借其优秀的全局搜索能力在轨迹规划过程得到广泛应用。朱宏震等学者^[2]利用个体与群体之间的有效信息, 提高粒子群算法种群丰富度, 跳出局部最优, 实现全局最短路径。李小为

等学者^[3]针对遗传算法参数控制问题, 设计了一种新的自适应参数调整策略, 利用改进的遗传算法离线解决了机械臂轨迹运行时间最优问题。郭清达等学者^[4]在考虑各类约束的情况下, 以 PUMA560 机器人为例, 进行轨迹规划实验仿真研究, 结果表明采用遗传算法优化后明显缩短了运行时间。窦艳艳^[5]结合遗传算法与栅格法, 解决机器人避障的路径规划问题, 并通过仿真证明该方法可以在较复杂的环境中规划出长度最短的可行路径。付荣等学者^[6]设计了一种基于多适应度函数评价机制的粒子群算法, 用于求解机械臂轨迹规划问题。李小为等学者^[7]提出了一种速度约束下的粒子群六自由度机械臂时间最优轨迹规划方法。孙倩等学者^[8]基于超粒子引导的自适应知识迁移的多任务差分进

作者简介: 周佳颖(2000-), 女, 本科生, 主要研究方向: 机械工程, Email: 1464086316@qq.com; 黄俊超(2000-), 男, 本科生, 主要研究方向: 机械工程; 赵宇凡(2001-), 男, 本科生, 主要研究方向: 机械工程; 宋泽凯(2001-), 男, 本科生, 主要研究方向: 机械工程。

收稿日期: 2023-03-06

哈尔滨工业大学主办 ◆ 科技创新与应用

化算法,通过实验表明该算法可以有效规避任务之间的负向迁移,提高相似度较低的任务组的优化性能。郭旺平^[9]提出改进型粒子群优化算法,证实了该算法在不相关多源频域载荷识别中的适用性和准确性。苏婷婷等学者^[10]运用PH曲线实现笛卡尔空间轨迹的转换,后采用3-4-5次多项式进行轨迹规划,有效改善了轨迹平滑性。郭明明等学者^[11]采用改进差分进化算法实现了协作型机器人轨迹规划。王学琨等学者^[12]利用差分进化算法求解3-5-3多项式插值法,使系统运行的时间达到了最小值、且运行较为平稳。

本文基于基本差分进化算法,引入多策略进化思想,提出一种改进的差分进化算法,并在位置、速度以及加速度等限制下,对机械臂末端执行器轨迹规划进行优化求解,进一步用实验和仿真验证该算法的有效性。

1 基本差分进化算法

差分进化算法是一种新的启发式搜索算法,具有结构简单、鲁棒性高等特点,在工程优化中得到广泛应用。该算法基本原理是通过种群初始化产生种群后,不断执行变异和交叉操作,生产新的中间个体,同时通过竞争选择完成主群体的更新,算法运行过程如下:

首先,通过式(1)对于随机生成的一组解进行初始化:

$$X_{ij}^G = \min_j + \text{rand}[0,1] \times (\max_j - \min_j) \quad (1)$$

其中, $i = 1, 2, \dots, N$ 、即 N 个个体, $j = 1, 2, \dots, D$ 、即 D 个染色体; $\text{rand}[0,1]$ 表示产生 0 至 1 均匀分布的随机数; \max_j 为该组解中的最大值; \min_j 为该组解中的最小值。

通过式(2)对于种群初始化后的个体采取变异处理,生成变异中间个体:

$$H_i^{G+1} = X_1^G + F \times (X_2^G - X_3^G) \quad (2)$$

其中, X_1^G, X_2^G, X_3^G 为群体内随机的 3 个不同个体; F 为缩放比例因子,又称变异因子。

F 越大,算法对局部搜索能力越强; F 越小,算法越能跳出局部最小点,但收敛速度也随之变慢。同时, H_i^{G+1} 需要满足边值条件,不能超出解空间的范围。

通过式(3)对 $X_{i,j}^G$ 与式(2)中生成的变异中间个体 $H_{i,j}^{G+1}$ 进行交叉操作,目的是增加种群的多样性:

$$V_{i,j}^{G+1} = \begin{cases} H_{i,j}^{G+1}, & \text{rand}(0,1) \leq CR \text{ or } j = \text{rand}[1,D] \\ X_{i,j}^G, & \text{otherwise} \end{cases} \quad (3)$$

其中, CR 为交叉概率,可以通过提高交叉概率,使信息量交换程度增大; $j = \text{rand}[1,D]$ 表示至少有一位变异传入,防止出现无效新个体。

通过式(4)对新的个体 V_i^{G+1} 和当前种群中的目标个体 X_i^G 进行比较:

$$X_i^{G+1} = \begin{cases} V_i^{G+1}, & \text{if } f(V_i^{G+1}) < f(X_i^G) \\ X_i^G, & \text{otherwise} \end{cases} \quad (4)$$

2 多策略自适应差分进化算法

基本差分进化算法存在进化策略单一和固定控制参数问题,本文提出一种多策略差分进化算法。该算法以基本差分进化算法的框架为基础,增加了潜在变异策略和控制参数的候选集合、同时在求解过程实时确定有效的控制策略和参数,从而确保在迭代过程中能够自设置更为可行的参数进行计算,有利于提升算法的搜索效率,防止陷入局部最优的现象。多策略差分进化算法流程如图1所示。

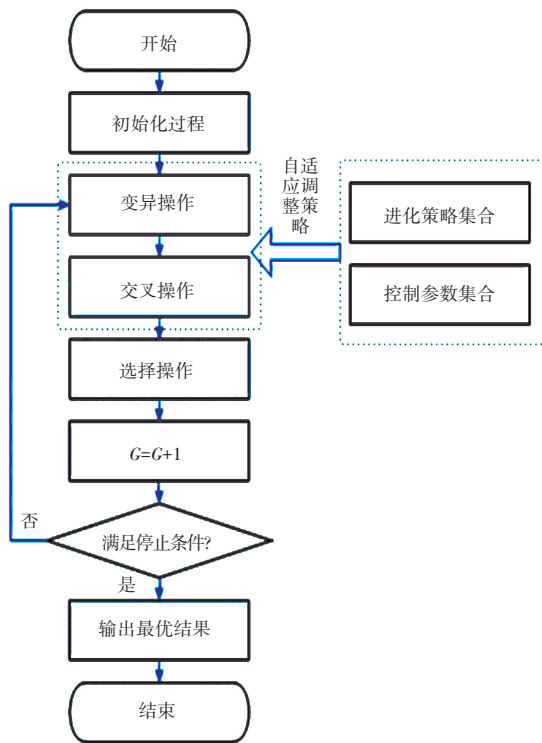


图1 多策略自适应差分进化算法流程图

Fig. 1 The flow chart of multi-strategy differential evolution algorithm

2.1 进化策略集合

为了兼顾多策略差分进化算法各阶段的探索与发展能力,选择4种常见的变异策略构成一个变异策略候选集合^[15],计算公式见式(5):

$$\begin{cases} H_i^{G+1} = X_1^G + F \times (X_2^G - X_3^G) \\ H_i^{G+1} = X_1^G + F \times (X_2^G - X_3^G) + F \times (X_4^G - X_5^G) \\ H_i^{G+1} = X_i^G + F \times (X_b^G - X_i^G) + F \times (X_1^G - X_2^G) \\ H_i^{G+1} = X_b^G + F \times (X_1^G - X_2^G) + F \times (X_3^G - X_4^G) \end{cases} \quad (5)$$

其中, $X_1^G, X_2^G, X_3^G, X_4^G, X_5^G$ 为区间 $[1, N]$ 中互不相等的随机整数; F 是缩放比例因子, 又称变异因子, 为区间 $[0, 1]$ 中的一个正的常数。同样, H_i^{G+1} 需要满足边值条件, 不能超出解空间的范围。

2.2 控制参数集合

选择变异策略后, 需要确定合适的变异因子 F 及交叉概率 CR 的值。多策略差分进化算法就是基于不同类型的离散值构成候选参数集合, 再在其中选取合适的值代入进化过程参与计算。

对于变异因子 F , 在区间 $[0.2, 1.0]$ 中, 以 0.2 为步长, 选取 4 个离散值构成变异因子 F 候选集合 FS , 而变异因子 F 决定算法局部搜索能力以及能否跳出局部最小点。对于交叉概率 CR , 在区间 $[0.0, 1.0]$ 中, 以 0.2 为步长, 选取 5 个离散值构成交叉概率 CR 候选集合 CRS 。

2.3 自适应调整策略

在进化过程中, 此自适应调整的方式可以根据搜索到的信息动态选择合适的变异策略, 以确定合适的位置进行变异, 其调整方式具体如下:

$$S_i^{t+1} = \begin{cases} randi(SS), & \text{if } rand[0, 1] < \tau_1 \\ S_i^t, & \text{otherwise} \end{cases} \quad (6)$$

其中, $randi(SS)$ 表示在变异策略候选集合 SS 范围内随机选取一个元素, $\tau_1 = 0.1$ 表示变异策略 S 的调整概率。

变异因子 F 的调整方式具体如下:

$$F_i^{t+1} = \begin{cases} randi(FS), & \text{if } rand[0, 1] < \tau_2 \\ F_i^t, & \text{otherwise} \end{cases} \quad (7)$$

其中, $randi(FS)$ 表示在变异因子候选集合 FS 范围内随机选择一个元素, $\tau_2 = 0.1$ 表示变异因子 F 的调整概率。

交叉概率 CR 的调整方式具体如下:

$$CR_i^{t+1} = \begin{cases} randi(CRS), & \text{if } rand[0, 1] < \tau_3 \\ CR_i^t, & \text{otherwise} \end{cases} \quad (8)$$

其中, $randi(CRS)$ 表示在交叉概率候选集合 CRS 范围内随机选取一个元素, $\tau_3 = 0.1$ 表示交叉概率 CR 的调整概率。

2.4 多策略差分进化算法基本流程

将候选进化策略集合、候选参数集合以及自适应调整策略融入基本差分进化算法, 可得到改进的

多策略自适应差分进化算法。该算法设计实现流程和步骤分述如下:

(1) 种群初始化。随机产生 n 维空间内 M 个满足约束条件的染色体, 随机选取第 i 个染色体的第 j 个维 $X_{i,j}^G$, 随机从变异策略候选集合 SS 、变异因子候选集合 FS 、交叉概率候选集合 CRS 中以自适应的方式选取变异策略 S_i^t, F_i^t 及 CR_i^t ;

(2) 根据对应的 S_i^t, F_i^t 及 CR_i^t , 从每个父代个体 X_i^G 中生成一个子代个体 H_i^G ;

(3) 依据式(4)、即差分进化算法的目标个体的选择步骤, 筛选出子代个体和父代个体中适应度较高的个体更替作为下一代进行后续操作。

(4) 采用式(6)~式(8)自适应调整和更新下一代进化时刻对应的变异策略 S_i^t, F_i^t 及 CR_i^t 。

(5) 判断是否满足结束条件, 若不满足停止条件, 则令 $G = G + 1$ 并返回步骤(2); 若满足停止条件, 则算法结束并输出最优结果。

3 末端执行器轨迹规划模型

相比笛卡尔空间规划, 关节空间的规划只需计算每个关节的插补点, 具有实现方便、计算简单等特点, 在实际控制系统设计过程中得到广泛应用^[16]。多项式插值可以保证运动轨迹的位姿、速度、加速度上的连续性, 是常用的插值计算方法^[17]。

3.1 五次样条轨迹插值函数轨迹规划方法

借鉴了梁萌等学者^[18]保证加速度和速度不发生突变的思想, 选择五次多项式差值进行拟合。假定从起始点到终点的整个完整路径可分为 2 个阶段, 那么对于第 j 个关节, 从起始位置到中间位置的路径方程, 式(9):

$$\theta_{i,i+1}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (9)$$

其中, 多项式常数系数 $a_0 \sim a_5$ 可以根据运动约束使用式(10)得到:

$$\begin{cases} \theta_i = a_0 \\ \theta_{i+1} = a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 + a_4 t_1^4 + a_5 t_1^5 \\ \dot{\theta}_i = a_1 \\ \dot{\theta}_{i+1} = a_1 + 2a_2 t_1 + 3a_3 t_1^2 + 4a_4 t_1^3 + 5a_5 t_1^4 \\ \ddot{\theta}_i = 2a_2 \\ \ddot{\theta}_{i+1} = 2a_2 + 6a_3 t_1 + 12a_4 t_1^2 + 20a_5 t_1^3 \end{cases} \quad (10)$$

其中, t_1 是起始位置到中间位置所用的时间。

中间位置到终止位置的路径方程, 可写为:

$$\theta_{i+1,f}(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \quad (11)$$

其中,多项式常数系数 $b_0 \sim b_5$ 可以按运动约束使用式(12)得到:

$$\begin{cases} \theta_{i+1} = b_0 + b_1 t_1 + b_2 t_1^2 + b_3 t_1^3 + b_4 t_1^4 + b_5 t_1^5 \\ \theta_f = b_0 + b_1 t_2 + b_2 t_2^2 + b_3 t_2^3 + b_4 t_2^4 + b_5 t_2^5 \\ \dot{\theta}_{i+1} = b_1 \\ \dot{\theta}_f = b_1 + 2b_2 t_2 + 3b_3 t_2^2 + 4b_4 t_2^3 + 5b_5 t_2^4 \\ \ddot{\theta}_{i+1} = 2b_2 + 6b_3 t_1 + 12b_4 t_1^2 + 20b_5 t_1^3 \\ \ddot{\theta}_f = 2b_2 + 6b_3 t_2 + 12b_4 t_2^2 + 20b_5 t_2^3 \end{cases} \quad (12)$$

其中, t_2 是中间位置到终止位置所用的时间。

3.2 优化目标函数

为了减少运动时间,以总体运行时间为优化目标,速度、加速度以及加加速度为约束条件,得到末端执行器优化轨迹规划数学模型^[19],推得的公式为:

$$f_1 = \text{运行时间} = t_1 + t_2 \quad (13)$$

约束条件: $|\dot{\theta}_j| \leq \omega_c$, $|\ddot{\theta}_j| \leq \alpha_c$, $|\dddot{\theta}_j| \leq J_c$ 。

其中, ω_c 为关节 j 的角速度约束; α_c 为关节 j 的加速度约束; J_c 为关节 j 的加加速度约束; t_1, t_2 分别表示第 j 个关节的运行轨迹中的起始位置到中间位置及中间位置到终止位置所用的时间。

设计变量 X 是从机械臂起始位置到中间位置所用的时间 t_1 、从中间位置到终止位置所用的行程时间 t_2 以及中间点的位置 $\theta_{i+1,j=1}, \theta_{i+1,j=2}, \dots, \theta_{i+1,j=n}$ 、速度 $\dot{\theta}_{i+1,j=1}, \dots, \dot{\theta}_{i+1,j=n}$ 和加速度 $\ddot{\theta}_{i+1,j=1}, \dots, \ddot{\theta}_{i+1,j=n}$, 由此推得:

$$X = \{t_1, t_2, \theta_{i+1,j=1}, \dots, \theta_{i+1,j=n}, \dot{\theta}_{i+1,j=1}, \dots, \dot{\theta}_{i+1,j=n}, \ddot{\theta}_{i+1,j=1}, \dots, \ddot{\theta}_{i+1,j=n}\} \quad (14)$$

其中, $n = 6$ 表示机械臂关节的数量,设计变量 X 的维度是 20。

4 实验仿真与结果分析

采用多策略自适应差分进化算法求解关节空间轨迹规划数学模型,同时与基本差分进化算法进行比较。参数设置如下:种群规模为 100,进化迭代次数为 100 次。令起始点和终止点的速度与加速度都为 0,同时机械臂在各个位置的关节角度的设置见表 1,每个关节的限位,如各关节角度、角加速度、角加速度的限制条件见表 2^[20]。

基本差分进化算法和本文算法在求解过程目标函数值变化曲线如图 2 所示。由图 2 可以看出,在相同的条件下,多策略差分进化算法的目标函数值具有较快的收敛速度,同时得到的最优解也相对较

好,表明本文多策略自适应差分进化在求解轨迹规划问题上具有一定的优势。

表 1 机械臂在各个位置的关节角度

Table 1 The angular positions of all joints deg

| 关节 | θ_i | θ_f |
|----|------------|------------|
| 1 | -10 | 55 |
| 2 | 20 | 35 |
| 3 | 15 | 30 |
| 4 | 150 | 10 |
| 5 | 30 | 70 |
| 6 | 120 | 25 |

表 2 每个关节的限位

Table 2 The constraints for all joints deg · s⁻²

| 关节 | ω_c | α_c | J_c |
|----|------------|------------|-------|
| 1 | 100 | 60 | 60 |
| 2 | 95 | 60 | 66 |
| 3 | 100 | 75 | 85 |
| 4 | 150 | 70 | 70 |
| 5 | 130 | 90 | 75 |
| 6 | 110 | 80 | 70 |

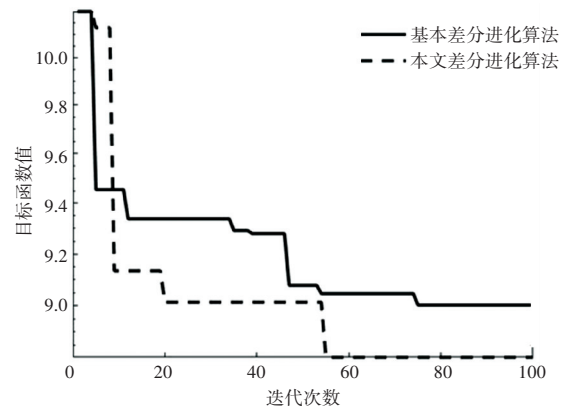


图 2 目标函数收敛曲线图

Fig. 2 The convergence graphs of objective functions

为了直观显示机械臂各关节运动过程,机械臂运动仿真如图 3 所示,令起始点和终止点的速度与加速度都为 0,分别针对各关节角位移、角速度、角加速度、二阶角加速度进行处理,各关节运动情况见图 3。

结合最优设计变量对应的关节运动过程,可见机械臂各关节运动过程中角位移曲线光滑、平稳,在满足表 1 和表 2 中关于 ω_c, α_c, J_c 约束预定义值的情况下更好地发挥机械臂的性能优势。