

基于改进粒子群算法的取件机械手轨迹综合优化设计

黄裘俊¹, 张 凯², 宋锦春¹, 于忠亮¹
(1. 东北大学 机械工程与自动化学院, 辽宁 沈阳 110819; 2. 沈阳化工大学 装备可靠性研究所, 辽宁 沈阳 110142)

摘 要: 为优化取件机械手进入模具时的运动轨迹,提出了二次拉格朗日插值粒子群算法(QLIPSO),此算法引入了二次拉格朗日插值局部搜索的方法,能够扩大搜索空间,避免局部收敛发生早熟,有效提高收敛精度.对比研究了典型粒子群算法(PSO)和几种改进型粒子群算法对取件机械手结构优化的设计效果.数值实验表明,QLIPSO算法具有最快的收敛速度,并且能够获得更好的优化结果.经该方法优化设计后,机械手在进入模具阶段的运动轨迹,与未优化前相比,直线度误差减少了98.06%,说明该方法能够有效优化取件机械手,获得更精确的运动轨迹.

关 键 词: 取件机械手;平面四杆机构;结构优化;轨迹综合;粒子群算法

中图分类号: TP 241 **文献标志码:** A **文章编号:** 1005-3026(2018)11-1636-06

Path Synthesis Optimal Design of Pick-Up Manipulator Based on Modified Particle Swarm Optimization

HUANG Qiu-jun¹, ZHANG Kai², SONG Jin-chun¹, YU Zhong-liang¹
(1. School of Mechanical Engineer & Automation, Northeastern University, Shenyang 110819, China; 2. Institute of Equipment Reliability, Shenyang University of Chemical Technology, Shenyang 110142, China. Corresponding author: ZHANG Kai, E-mail: zk217@163.com)

Abstract: To optimize the motion curve for the pick-up manipulator when going into the mold, an novel method called quadratic Lagrange interpolation particle swarm optimization (QLIPSO) algorithm which can expand the search space, prevent premature convergence, and improve the convergence accuracy effectively is proposed. To verify the effectiveness of QLIPSO, four modified PSO algorithms are compared to synthesize the four-bar linkage of the manipulator. The numerical simulation results show that a better fitness value can be obtained with the QLIPSO algorithm. The linearity error of the curve of the manipulator, optimized by the proposed method, is decreased by 98.06% comparing with that non-optimized. It indicates that the proposed method can effectively improve the motion performance of the manipulator.

Key words: pick-up manipulator; planar four-bar linkage; structural optimization; path synthesis; particle swarm optimization(PSO)

取件机械手是通过四杆机构来实现从模具夹取产品后按理想的轨迹将产品送至指定区域的自动化设备.模具开模的空间一般比较狭长,取件机械手手爪进入模具的轨迹影响了产品的合格率、模具开模的空间设计和取件机械手的应用范围,因此,有必要对机械手运动轨迹进行优化.

四杆机构是一类十分重要的机构,从简单设备,如挡风玻璃擦和开关门机构,到复杂设备,如岩石破碎、缝纫机、汽车悬架系统和机械手等都有广泛的应用^[1].轨迹综合是平面四杆机构设计的基本问题之一,其主要目的是找到一个平面四杆结构,使得连杆机构上的点能沿某一预先给定的轨迹行走.传统的图解法存在误差大、综合精度低等缺点,目前几乎无人使用^[2].随着计算机的发展和运算能力的提高,越来越多学者利用智能计算方法求解平面四杆机构的轨迹综合问题. Felezi

等基于多目标均匀多样性遗传算法优化设计了水稻插秧机上的可调四杆机构^[3]. Lin 提出一种遗传-差分混合算法用于求解四杆机构的路径综合问题^[4]. Ramanpreet 等对比研究遗传算法(GA)、粒子群算法(PSO)和教与学优化算法(TLBO)用于合成人体膝关节外骨骼的曲柄摇杆机构的效果^[5]. Kafash 和 Nahvi 提出近似圆形函数,并用差分进化算法进行求解平面四杆机构的轨迹综合^[6]. Ahmadi 等利用斯塔克尔伯格博弈论和神经网络算法,求解了四杆机构的多目标优化合成问题^[7]. 虽然这些算法都能较好求解轨迹综合问题,但是,由于求解问题的复杂性,对于求解方法的精度和可靠性,仍然有可以改进的空间^[8].

对此,本文针对取件机械手进入模具的运动轨迹优化设计问题进行研究. 首先,建立了取件机械手数学模型,定义了最小直线度误差目标函数,其次,提出了基于二次拉格朗日局部搜索插值粒子群算法(QLIPSO)的平面四杆结构优化方法,最后,通过数值实验对比验证了所提出优化设计方法的有效性.

1 取件机械手数学模型

1.1 取件机械手四杆机构的位置关系

本文将取件机械手简化成一个曲柄摇杆机构,结构简图如图 1 所示,AE 杆可视为固定的机架,AB 杆为摇杆,ED 杆为曲柄,BCD 为连杆,F 点表示手爪的中心位置,B,C,D 和 F 点是一个固定整体. 以 A 点为原点建立坐标系,设 AB,BC,CD,BD,DE,AE 和 DF 长度分别为 $l_1, l_6, l_7, l_2, l_3, l_4$ 和 l_5 ,AB,BC,DE 和 DF 与水平方向夹角分别为 $\varphi_1, \varphi_2, \varphi_4$ 和 φ_6 ,BC 和 CD,CD 和 DB 夹角分别为 φ_5, φ_3 .

已知 A 点坐标(x_A, y_A)和 E 点坐标(x_E, y_E),杆长 l_1, l_6, l_7, l_3 和 l_5 ,角 φ_5 和 φ_3 ,求手爪 F 点的坐标.

则 BD 的长度由余弦定理可得

$$l_2 = \sqrt{l_6^2 + l_7^2 - 2\cos\varphi_5 l_6 l_7}. \quad (1)$$

B 点坐标和 D 点坐标如式(2)和式(3)所示:

$$\begin{cases} x_B = x_A + l_1 \cos\varphi_1, \\ y_B = y_A + l_1 \sin\varphi_1. \end{cases} \quad (2)$$

$$\begin{cases} x_D = x_B + l_2 \cos\varphi_2 = x_E + l_3 \cos\varphi_4, \\ y_D = y_B + l_2 \sin\varphi_2 = y_E + l_3 \sin\varphi_4. \end{cases} \quad (3)$$

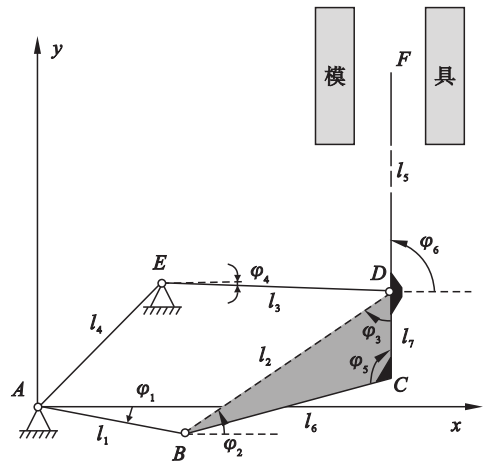


图 1 取件机械手结构简图

Fig. 1 Schematic diagram of the pick-up manipulator

将式(2)代入式(3),可得

$$\begin{cases} x_D = x_A + l_1 \cos\varphi_1 + l_2 \cos\varphi_2 = x_E + l_3 \cos\varphi_4, \\ y_D = y_A + l_1 \sin\varphi_1 + l_2 \sin\varphi_2 = y_E + l_3 \sin\varphi_4. \end{cases} \quad (4)$$

将关系式(4)移项平方相加,消去 φ_4 项,可得 φ_2 关于 φ_1 的表达式:

$$C + D\cos\varphi_2 + E\sin\varphi_2 = 0. \quad (5)$$

式中: $C = K_1^2 + K_2^2 + l_2^2 - l_3^2$; $D = 2K_1 l_2$; $E = 2K_2 l_2$;
 $K_1 = x_A + l_1 \cos\varphi_1 - x_E$; $K_2 = y_A + l_1 \sin\varphi_1 - y_E$.

由式(5)可解得:

$$\varphi_2 = \arctan\left(\frac{-E \pm \sqrt{E^2 + D^2 - C^2}}{C - D}\right). \quad (6)$$

式中,正负号需根据实际情况选择,在这里取负号.

F 点坐标为

$$\begin{cases} x_F = x_D + l_5 \cos\varphi_6, \\ y_F = y_D + l_5 \sin\varphi_6. \end{cases} \quad (7)$$

将式(4)代入式(7)可求得 F 点坐标:

$$\begin{cases} x_F = x_A + l_1 \cos\varphi_1 + l_2 \cos\varphi_2 + l_5 \cos\varphi_6, \\ y_F = y_A + l_1 \sin\varphi_1 + l_2 \sin\varphi_2 + l_5 \sin\varphi_6. \end{cases} \quad (8)$$

F 点坐标中, φ_2 是关于 φ_1 的表达式, $\varphi_6 = \varphi_3 + \varphi_2$, φ_3 角已知,因此 F 点坐标是关于角 φ_1 的数学模型.

1.2 设计变量

因为要求手爪 F 点的轨迹,因此,设定与 F 点坐标相关的 φ_1, l_1, l_2 和 l_5 为输入变量,而 l_2 由 l_6, l_7 和 φ_5 所确定,因此,本文设定 $\varphi_1, l_1, l_6, l_7, l_5, \varphi_5$ 为输入变量,即

$$\mathbf{X} = (\varphi_1, l_1, l_6, l_7, l_5, \varphi_5). \quad (9)$$

设定(x_A, y_A), (x_E, y_E), l_3 和 l_4 为固定参数.

1.3 目标函数

为了评价取件机械手爪在进入模具时所行走的轨迹,在 φ_1 取值为 $[\varphi_{1\min}, \varphi_{1\max}]$ 区域均匀选取 m 个 F 点,设其横坐标为 $x_{Fi} (i=1,2,\dots,m)$,横坐标平均值为

$$\bar{x}_F = (\sum_{i=1}^m x_{Fi})/m. \quad (10)$$

根据 Bahman 对四杆机构的轨迹跟踪误差 (tracking error) 的定义^[7],定义直线度误差 (linearity error) 为 m 个 F 点的横坐标与其平均值差值的平方和用于评价轨迹的直线度,即

$$f(X) = \sum_{i=1}^m (x_{Fi} - \bar{x}_F)^2. \quad (11)$$

用 $\delta_{AB}(X)$ 表示算法 A 相比算法 B 减小的直线度误差:

$$\delta_{AB}(X) = 1 - f_A(X)/f_B(X). \quad (12)$$

需要优化的综合问题为

$$\left. \begin{array}{l} \min(f(X)), \\ \text{s. t. } X_{\min} < X < X_{\max}. \end{array} \right\} \quad (13)$$

1.4 约束条件

将取件机械手简化成曲柄摇杆机构 $ABDE$,首先需要满足曲柄摇杆 2 个杆长约束条件:

- 1) 摇杆 AB 为最短杆,即 $l_1 < l_2, l_3, l_4$;
- 2) 摇杆 AB 和连杆 DE 杆长之和小于另外 2 杆 AE 和 BD 杆长之和,即 $l_1 + l_2 < l_3 + l_4$.

另外,还需要使设计变量在设定的变化范围内.

2 轨迹优化问题求解

2.1 粒子群算法

Kennedy 和 Eberhart 在 1995 年第一次提出粒子群优化 (particle swarm optimization, PSO) 算法^[9],用于求解非线性函数,该算法是模拟鸟群飞行觅食行为,鸟之间通过集体的协作达到最优的目的. PSO 算法没有像遗传算法的选择、交叉和变异等操作,使用简单结构进行迭代计算,参数易调整、计算速度快,在控制优化领域被广泛地应用.

假设粒子的种群数为 n ,搜索空间为 D 维,则第 i 个粒子所在位置表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $i=1,2,\dots,n$,其位置分量 x_{ij} 在 $[x_{j\min}, x_{j\max}]$ ($j=1,2,\dots,D$) 范围内取值,每个粒子的位置就是潜在的解,根据 X_i 所对应的目标函数适应值大小,判断粒子的优劣.第 i 个粒子飞行速度为 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ ($i=1,2,\dots,n$),其速度分量 v_{ij} 在 $[v_{j\min}, v_{j\max}]$ ($j=1,2,\dots,D$) 范围内取值, V_i 决定

了粒子 i 在问题空间搜索的“步伐”.则对于每个粒子,其第 d 维 ($1 \leq d \leq D$) 的位置和速度,将分别根据式 (14) 和式 (15) 迭代:

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_g^k - x_{id}^k), \quad (14)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \quad (15)$$

其中: v_{id}^k 为经过 k 次迭代后粒子 i 在 d 维方向上的速度分量; x_{id}^k 为经过 k 次迭代后粒子 i 的 d 维方向上的位置分量; 粒子 i 经历 k 次搜索后达到的最优适应值位置 (p_{best}) 为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, 所有粒子的 k 次搜索中达到的全局历史最优适应值位置 (g_{best}) 为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$; c_1 和 c_2 为学习因子 (或称加速系数); r_1 和 r_2 为介于 0 到 1 的随机数.

2.2 拉格朗日插值粒子群算法

然而,标准 PSO 算法存在着早熟和精度低等问题,许多学者在标准 PSO 算法基础上致力于改进粒子群算法性能.如, Liang 等改进了标准 PSO 算法,提出了综合自学习粒子群算法 (comprehensive learning particle swarm optimizer, CLPSO)^[10]; Nasir 基于 CLPSO 提出动态邻域自学习粒子群算法 (dynamic neighborhood learning based particle swarm optimizer, DNLP SO)^[11]; Zhang 等在 CLPSO 基础上提出拉格朗日局部插值算法 (local search with Lagrange interpolation particle swarm optimization, LILPSO)^[12], 该算法初期具有较大的搜索空间,后期能帮助加速收敛,具有良好的鲁棒性.这些算法都有效改善了标准 PSO 由于陷入局部最优和提前收敛产生的早熟问题.为了进一步探索算法收敛精度,求解本文需要解决的轨迹综合问题,在 LILPSO^[12] 算法基础上,引入二次拉格朗日插值局部搜索方法,提出了一种改进型 PSO 算法,称之为二次拉格朗日插值粒子群算法 (quadratic lagrange interpolation particle swarm optimization, QLIPSO).

QLIPSO 算法的主要思路为

- 1) 迭代初期扩大搜索范围防止粒子陷入局部收敛,迭代后期缩小搜索范围,加快收敛速度.

粒子速度迭代公式为

$$v_{id}^{k+1} = wv_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k). \quad (16)$$

速度迭代公式 (16) 相比 PSO 算法,只保留自身的惯性部分和向自身历史最优 p_{best} 学习部分,删除了向全局最优粒子 g_{best} 学习部分,这样可以避免局部收敛.速度迭代公式 (16) 还加入了按式 (17) 线性递减的惯性权重 w :

$$w = w_{\max} - (w_{\max} - w_{\min}) \frac{k}{\text{MaxDT}}. \quad (17)$$

其中： $w_{\max} = 0.9$ ； $w_{\min} = 0.4$ ； k 为迭代次数；MaxDT 为最大迭代次数。

惯性权重 w 是用于平衡全局搜索和局部搜索的， w 越大，全局搜索能力越强； w 越小，局部搜索能力越强。因此，初期使用较大的 w ，扩大搜索区域，防止局部收敛，后期使用较小的 w ，提高局部搜索能力，加快收敛速度。

2) 利用拉格朗日二次插值进行局部搜索，提高局部寻优能力。

利用拉格朗日插值法可以找出一个恰好穿过二维平面上若干个已知点的多项式函数。因此，为了根据目前已知的有限信息，能够预测比较准确的未知信息，获取更佳的最优解，本文引入拉格朗日插值法， n 阶拉格朗日插值公式为

$$f_n(x) = \sum_{k=0}^n \left(\prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j} \right) y_0. \quad (18)$$

这里只用到二阶的拉格朗日插值，因此式 (18) 的 n 取 2，需要三个插值点。为探索比现有 g_{best} 更佳的位置，本文以 $g_{\text{best}}(X_0)$ 和 g_{best} 附近的两个随机点 X_1, X_2 作拉格朗日插值， X_1, X_2 生成公式：

$$x_{1j} = x_{0j} - \text{delta}(j), \quad (19)$$

$$x_{2j} = x_{0j} + \text{delta}(j), \quad (20)$$

$$\text{delta}(j) = \alpha \times (\text{rand} - 0.5). \quad (21)$$

其中， α 是一个非常小的系数，这里设定 $\alpha = 0.01$ 。

利用二次拉格朗日插值，可以求得一条经过 X_1, X_2 和 $g_{\text{best}}(X_3)$ 三点的曲线，曲线在 X_1, X_2 之间的最小值点 X_3 即为潜在的全局最优解。

2.3 算法步骤

QLIPSO 算法具体实现步骤如下：

步骤 1 设定初始种群参数，如学习因子 c_1 、惯性权重 w 、种群维数 D 、群体个体数目 n 、最大迭代次数 MaxDT，以及粒子各维度的位置变化范围等。

步骤 2 随机初始化种群粒子的初始位置和速度，计算各粒子初始适应值，各个粒子初始适应值即为个体粒子历史最优 p_{best} ，其中最佳的 p_{best} 即为全体粒子历史最优 g_{best} 。

步骤 3 分别按式 (14) 和式 (15) 更新粒子位置和速度。

步骤 4 计算各粒子更新后的适应值。

步骤 5 对比更新后的每个粒子和其 p_{best} 的适应值，若更新后的粒子的适应值更佳，则用该粒子更新后的信息更新 p_{best} 的位置和适应值，否则保留 p_{best} 信息；对比每个 p_{best} 和迭代至今的全体历

史最优 g_{best} 的适应值，若存在比 g_{best} 的适应值更佳的 p_{best} ，则以 p_{best} 信息更新 g_{best} 位置和适应值，否则保留 g_{best} 信息。

步骤 6 当迭代次数达到 g 的倍数，则分别以全局最优解 g_{best} 各维度的分量为中心进行拉格朗日插值计算：

1) 取 $g_{\text{best}}(X_0)$ 点，按式 (19) 和式 (20) 获得 g_{best} 附近 2 个插值点 X_1, X_2 ，计算 X_1, X_2 的适应值；

2) 利用 X_0, X_1, X_2 进行拉格朗日插值计算，求得最小值点 X_3 ，并计算 X_3 的适应值；

3) 若 X_1, X_2 或 X_3 比 X_0 的适应值更佳，则更新 g_{best} 信息。

步骤 7 判断是否达到最大迭代次数，若达到最大迭代次数则优化过程结束，否则返回步骤 3。

QLIPSO 算法流程如图 2 所示。

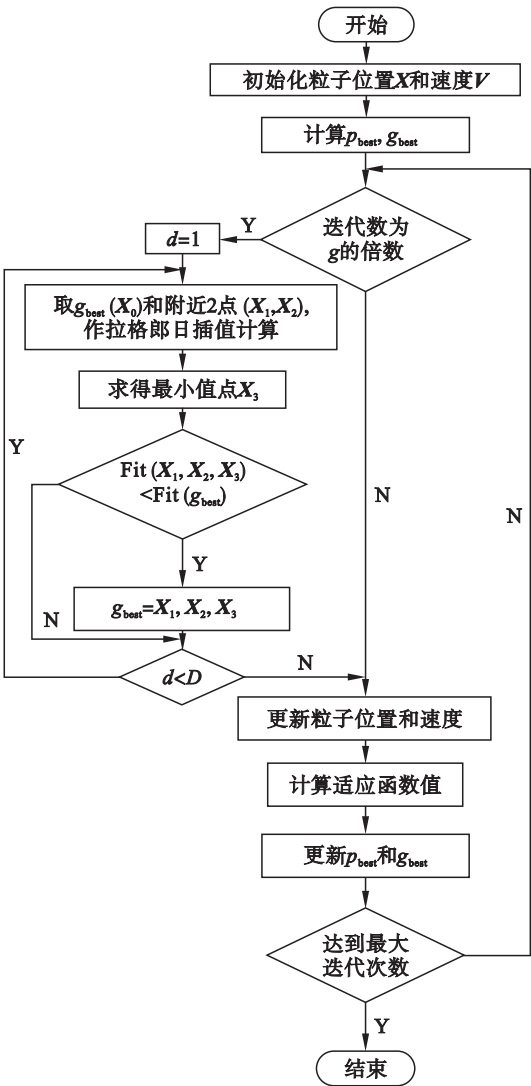


图 2 QLIPSO 算法流程图
Fig. 2 Flowchart of QLIPSO algorithm

3 数值实验与结果分析

3.1 参数设置

设置种群参数如下:种群维数 $D=6$, 群体个体数目 $n=40$ 、最大迭代次数 $\text{MaxDT}=10\ 000$, 学习因子 $c_1=2$, 惯性权重 $w_{\max}=0.9, w_{\min}=0.4$, A 点坐标 $(0,0)$, E 点坐标 $(230,230)$, DE 杆长 $l_3=430$, 迭代速度变化范围限制为 $0.1 \times (X_{\max} - X_{\min})$, 目标函数所取 F 点个数 $m=6$, φ_1 变化范围 $\varphi_{1\min}=-1.4, \varphi_{1\max}=-0.15$, 粒子各维度的位置变化范围如表 1 所示.

表 1 变量 X 变化范围表						
Table 1 Variation rang of variable X						
变量	φ_1 rad	l_1 mm	l_6 mm	l_7 mm	l_5 mm	φ_5 rad
X_{\min}	-1.2	286	412	171	950	1.8
X_{\max}	-1.6	266	392	151	850	1.6

3.2 实验结果

在 3.1 节设定的参数条件下, 分别使用 $\text{PSO}^{[9]}$, $\text{CLPSO}^{[10]}$, $\text{DNLPSO}^{[11]}$, $\text{LILPSO}^{[12]}$ 和 QLIPSO 算法对取件机械手的四杆结构进行优化设计, 得到的目标函数适应值随迭代次数的变化曲线如图 3 所示. 可见, PSO 和 DNLPSO 算法受迭代初值影响, 随后并没有继续收敛, CLPSO , LIPSO 和 QLIPSO 算法迭代初期都具有一定收敛速度, 中后期都陷入了局部最优, 相比之下, QLIPSO 收敛速度最快, 收敛精度最优, 这是因为 QLIPSO 算法初期没有向 g_{best} 学习, 扩大搜索空间, 有效避免了早熟现象, 在迭代后期, 又利用了二次拉格朗日插值局部搜索, 加快了收敛速度.

几种算法迭代 1 000 次后与未优化过的变量 X 及其直线度误差对比如表 2 所示, 经 QLIPSO 优化后的直线度误差为 $2.007\ 1\ \text{mm}^2$, 优于未优化过的和经 PSO , CLPSO , DNLPSO , LILPSO 算法优化过的直线度误差分别为 $103.589\ 5, 18.875\ 8, 15.543\ 6, 10.162\ 9, 3.053\ 4\ \text{mm}^2$, 直线度误差减小了 $98.06\%, 89.37\%, 80.25\%, 34.27\%, 15.22\%$. 这表明具有二次拉格朗日插值局部搜索的 QLIPSO 算法在处理四杆机

构优化设计的问题上, 具有更好的收敛速度和精度.

因此, 本文选择 QLIPSO 算法对取件机械手进行优化设计, 优化前后的取件机械手手爪运动轨迹对比如图 4 所示. QLIPSO 算法优化后的效果十分明显, 在进入模具取件阶段, 机械手手爪运动轨迹近乎是条直线, 直线度误差减少了 98.06% . 说明该方法能够有效优化取件机械手的运动轨迹. 经 QLIPSO 算法优化后的机械手夹取产品时更不易与模具发生刮碰, 不仅能提高产品的合格率, 还可以节约模具开模设计空间, 使取件机械手能应用到更多的生产设备中.

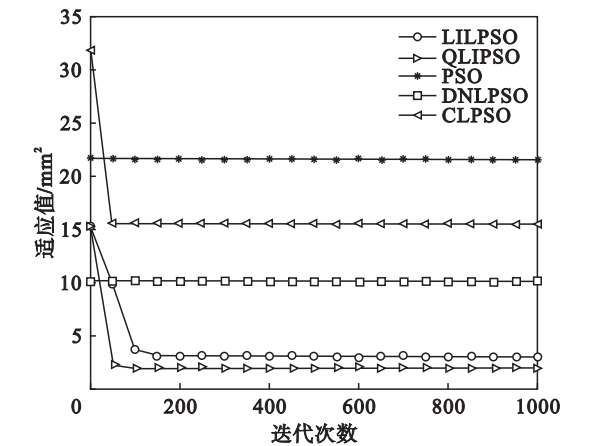


图 3 QLIPSO 与 PSO 和几种改进型算法收敛情况对比
Fig. 3 Comparison of the convergence of QLIPSO, PSO and other modified algorithms

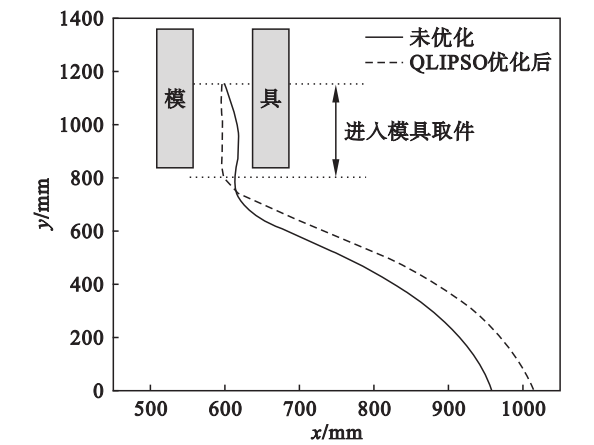


图 4 QLIPSO 算法优化前后取件机械手手爪运动轨迹对比
Fig. 4 Comparison of pick-up manipulator movement path before and after using the QLIPSO algorithm

表 2 几种算法优化前后的变量 X 及直线度误差对比
Table 2 Comparison of the variable X and its linearity error before and after using algorithms

变量	优化前	PSO ^[9]	CLPSO ^[10]	DNLPSO ^[11]	LILPSO ^[12]	QLIPSO
φ_1/rad	-1.4	-1.221 9	-1.226 5	-1.402 3	-1.319 4	-1.323 5
l_1/mm	276	270.588 8	271.518 6	268.052 6	266	266
l_6/mm	402	408.7340	398.980 2	394.826 0	397.932 9	412
l_7/mm	161	167.625 9	171	169.710 6	155.531 8	159.472 7
l_5/mm	900	913.259 6	943.300 4	939.926 4	950	950
φ_5/mm	1.7	1.624 3	1.666 2	1.708 3	1.737 1	1.629 3
线性误差/ mm^2	103.589 5	18.875 8	15.543 6	10.162 9	3.053 4	2.007 1

4 结 论

1) 本文研究了取件机械手的四杆结构的优化设计问题,提出了基于二次拉格朗日插值粒子群算法 (QLIPSO) 的平面四杆结构优化设计方法。

2) 通过数值实验对比验证,在设定相同参数条件下,分别使用 PSO, CLPSO, DNLPSO, LILPSO 和 QLIPSO 算法对取件机械手的四杆结构进行优化设计,QLIPSO 算法后期收敛效果明显要优于其他算法.迭代 1 000 次后,QLIPSO 的直线度误差为 2.007 1 mm²,优于其他几种算法.验证了具有二次拉格朗日局部插值搜索的 QLIPSO 算法不仅能够扩大搜索空间,避免局部收敛,发生早熟,还具有更高收敛精度。

3) 运用所提出的 QLIPSO 算法优化后的取件机械手在进入模具阶段的运动轨迹几乎是条直线,与未优化前的取件机械手相比,直线度误差减少了 98.06%,显著提升了取件机械手运动效果,从而说明所提出的优化设计方法可以获得更精确的轨迹。

参考文献:

[1] Acharyya S K,Mandal M. Performance of EAs for four-bar linkage synthesis[J]. *Mechanism and Machine Theory*,2009, 44(9):1784-1794.

[2] 于红英,赵彦微,许栋铭. 平面铰链四杆机构的轨迹综合方法[J]. 哈尔滨工业大学学报,2015,47(1):40-47.
(Yu Hong-ying, Zhao Yan-wei, Xu Dong-ming. A path synthesis method of planar hinge four-bar linkage [J]. *Journal of Harbin Institute of Technology*,2015,47(1):40-

47.)

[3] Felezi M E, Vahabi S, Nariman-Zadeh N. Pareto optimal design of reconfigurable rice seedling transplanting mechanisms using multi-objective genetic algorithm [J]. *Neural Computing and Applications*, 2016, 27 (7): 1907-1916.

[4] Lin W Y. A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage [J]. *Mechanism and Machine Theory*,2010,45(8):1096-1107.

[5] Singh R, Chaudhary H, Singh A K. Defect-free optimal synthesis of crank-rocker linkage using nature-inspired optimization algorithms [J]. *Mechanism and Machine Theory*,2017,116:105-122.

[6] Kafash S H, Nahvi A. Optimal synthesis of four-bar path generator linkages using circular proximity function [J]. *Mechanism and Machine Theory*,2017,115:18-34.

[7] Ahmadi B, Nariman-Zadeh N, Jamali A. Path synthesis of four-bar mechanisms using synergy of polynomial neural network and Stackelberg game theory [J]. *Engineering Optimization*,2017,49(6):932-947.

[8] Lin W Y, Hsiao K M. A new differential evolution algorithm with a combined mutation strategy for optimum synthesis of path-generating four-bar mechanisms [J]. *Proceedings of the Institution of Mechanical Engineers Part C*,2017,231(14):2690-2705.

[9] Kennedy J, Eberhart R. Particle swarm optimization [C]// *Proceedings of IEEE International Conference on Neural Networks*. Piscataway: IEEE,1995:1942-1948.

[10] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. *IEEE Transactions on Evolutionary Computation*,2006,10(3):281-295.

[11] Nasir M, Das S, Maity D, et al. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization [J]. *Information Sciences*,2012,209:16-36.

[12] Kai Z, Jinchun S, Ke N, et al. Lagrange interpolation learning particle swarm optimization [J]. *PLoS One*, 2016, 11 (4):e0154191.