

## 基于深度邻域搜索 PSO 算法的装配序列优化问题

宫 华, 袁 田, 张 彪

(沈阳理工大学 理学院, 沈阳 110159)

**摘 要:** 针对产品结构特征建立几何约束矩阵, 以最大化满足几何约束条件装配次数和最小化装配方向改变次数为目标, 研究产品装配序列优化问题. 利用值变换的粒子位置和速度更新规则, 基于具有随机性启发式算法产生初始种群, 提出一种带有深度邻域搜索改进策略的粒子群算法解决装配序列问题. 通过装配实例验证了所提出算法的性能并对装配序列质量进行了评价, 所得结果表明了该算法在解决装配序列优化问题上的有效性与稳定性.

**关键词:** 装配序列优化; 粒子群算法; 深度邻域搜索; 几何约束

**中图分类号:** TP205

**文献标志码:** A

## Assembly sequence planning problem based on particle swarm optimization algorithm with depth local search

GONG Hua, YUAN Tian, ZHANG Biao

(School of Science, Shenyang Ligong University, Shenyang 110159, China. Correspondent: GONG Hua, E-mail: gonghua1018@sina.com)

**Abstract:** For the structure characters of the products, a matrix of geometric constraints is established. An optimization model of assembly sequence planning is studied to maximize the number of assembly operations with the geometric assembly constraints and minimize the number of changing assembly direction. The updating rules of particle position and velocity are derived based on the value transformation. A heuristic algorithm with randomness characteristic is provided in order to generate the initial populations. A particle swarm optimization algorithm with the depth local search strategy is developed to solve the assembly sequence planning problem. To evaluate the performance of the algorithm and the quality of the assembly sequences, an experiment of an assembly instance is tested. The results show the effectiveness and stability of the proposed algorithm on solving the assembly sequence planning problem.

**Keywords:** assembly sequence planning; particle swarm optimization; depth local search; geometric constraints

### 0 引 言

装配序列优化 (ASP) 是产品各零件的装配顺序和装配方向的排序, 是机械行业及军工企业装配过程的关键环节. 由于不确定性因素影响, 配套零件的计划供应时间与实际供应时间常常存在偏差, 合理优化装配序列不仅能够减少零件堆积时间, 改善产线均衡, 而且对于降低产品成本及提高生产效率具有重要意义. 优化装配序列能够为企业生产运作、物流管理的决策依据, 实现从以粗放式管理为特征的传统机械及军工制造企业向以信息化和集成化为特征的现代制造企业转变.

产品装配序列优化是典型 NP- 难解问题, 随着零

件个数的增多, 求解规模将出现组合爆炸. 因此, 智能优化算法在解决装配序列优化方面得到了广泛应用. Romeo 等<sup>[1]</sup>、Wang 等<sup>[2]</sup> 分别运用遗传算法、蚁群算法解决 ASP 问题; Cao 等<sup>[3]</sup> 提出了免疫和克隆选择相结合的人工免疫系统来研究 ASP 问题; Lü 等<sup>[4]</sup> 基于映射策略、粒子位置和速度编码方式及粒子更新操作, 提出了离散粒子群算法以解决装配序列问题; Wang 等<sup>[5]</sup>、Li 等<sup>[6]</sup> 分别提出了混沌粒子群算法、离散粒子群算法的混合算法以解决 ASP 问题; Zhang 等<sup>[7]</sup> 利用免疫算法能克服粒子群算法早熟收敛的特性来解决 ASP 问题; Somayé 等<sup>[8]</sup> 研究了 ASP 问题的解空间分布, 基于近优解在解空间呈现近似均匀分布的性

**收稿日期:** 2015-07-01; **修回日期:** 2015-12-15.

**基金项目:** 国家自然科学基金项目(71101097); 辽宁省“百千万人才工程”培养项目(2014921043); 辽宁省高等学校优秀人才支持计划项目(2015056); 辽宁省先进制造技术与装备重点实验室开放基金课题.

**作者简介:** 宫华(1976—), 女, 教授, 博士后, 从事组合优化、生产调度等研究; 袁田(1986—), 女, 硕士生, 从事智能优化算法的研究.

质,提出了一种跳出局部搜索(BLS)算法.

现有的群智能算法中种群多样性较差,对初始种群的质量和大小依赖性较强.本文在满足产品几何约束条件下建立干涉矩阵.利用粒子位置和速度更新规则,通过具有随机性特点的启发式算法产生初始种群,提高初始种群的质量.现有的粒子群算法解决ASP问题时容易陷入局部最优,收敛速度不理想.本文在文献[8]的基础上,提出基于4种深度邻域搜索改进策略的粒子群算法来解决装配序列优化问题,增强粒子群算法局部搜索能力,在保证收敛速度的前提下,迅速生成最优装配序列.通过装配实例对算法进行验证,将实验结果与现有文献中的方法进行比较.对基于装配序列方向改变次数、满足几何约束下装配次数以及稳固支撑零件个数进行装配序列质量评价,所得结果表明了采用本文算法解决装配序列优化问题的有效性和稳定性.

## 1 问题描述

### 1.1 问题假设

1) 产品所有零件都是刚性的,在装配过程中不可发生形变;

2) 装配方向包括 $\pm x, \pm y, \pm z$ ;

3) 每个零件只能被装配一次,且装配操作必须逐步进行.

### 1.2 符号说明

$n$ : 组成产品的所有零件个数.

$p_i$ : 第 $i$ 个零件,  $i = 1, 2, \dots, n$ .

$d_i$ : 第 $i$ 个零件的装配方向,  $d_i \in \{\pm x, \pm y, \pm z\}$ .

$X_l$ : 第 $l$ 个装配序列,  $X_l = [x_{l1}, \dots, x_{li}, \dots, x_{ln}]$ .

其中:  $x_{li} = (p_{li}, d_{li})$ ,  $p_{li}$ 代表第 $l$ 个装配序列的第 $i$ 个零件,  $d_{li} = \{1, 2, \dots, 6\} \in \{\pm x, \pm y, \pm z\}$ 代表该零件的装配方向.

### 1.3 约束条件

采用干涉矩阵描述零件在所有装配方向上的几何约束.通常干涉矩阵是根据装配方向建立的,故干涉矩阵可表示为

$$IM_k = \begin{bmatrix} I_{11k} & I_{12k} & \dots & I_{1nk} \\ I_{21k} & I_{22k} & \dots & I_{2nk} \\ \vdots & \vdots & \ddots & \vdots \\ I_{n1k} & I_{n2k} & \dots & I_{nnk} \end{bmatrix}. \quad (1)$$

其中:  $k \in \{\pm x, \pm y, \pm z\}$ 表示装配方向;  $I_{ijk}$ 是一个二进制变量,如果零件 $p_i$ 在 $k$ 装配方向上对零件 $p_j$ 有干涉,则 $I_{ijk} = 1$ ,否则 $I_{ijk} = 0$ .特别地,  $I_{iik} = 0$ ,因为零件不会干涉自身.

干涉矩阵的主要作用是导出下一个装配零件 $p_i$

及其装配方向.假设已有 $m$ 个零件完成装配,临时子装配体可以表示为 $X_{\text{sub}} = [p_1, p_2, \dots, p_m]$ ,零件 $p_i$ 能否顺利装配取决于 $I_{ik}$ 的值,  $k \in \{\pm x, \pm y, \pm z\}$ .  $I_{ik}$ 的值由下式计算:

$$I_{ik} = I_{i1k} \vee I_{i2k} \vee \dots \vee I_{imk},$$

其中 $\vee$ 为布尔操作“或”.如果 $I_{ik} = 0$ ,则零件在 $k$ 装配方向上对所有零件都不干涉;否则,说明零件 $p_i$ 在装配方向 $k$ 上至少对一个零件有干涉.

### 1.4 目标函数

假设一个装配序列为 $X_l = [x_{l1}, \dots, x_{li}, \dots, x_{ln}]$ ,设该序列满足几何约束的次数,即在不干涉已装配零件前提下进行装配的零件个数为

$$n_{lp} = \sum_{i=1}^n PI_{ik},$$

其中 $PI_{ik}$ 可表示为

$$PI_{ik} = \begin{cases} 1, & I_{ik} = 0; \\ 0, & I_{ik} = 1; \end{cases} \quad k \in \{\pm x, \pm y, \pm z\}.$$

如果该序列中零件 $p_{l,i+1}$ 和零件 $p_{l,i}$ 的装配方向相同,则该次装配不需改变方向,记 $Q_{li} = 0$ ;否则,记 $Q_{li} = 1$ ,表示为

$$Q_{li} = \begin{cases} 0, & d_{l,i+1} = d_{l,i}; \\ 1, & d_{l,i+1} \neq d_{l,i}. \end{cases}$$

设序列完成装配需要改变方向的次数为 $n_{lq}$ ,有

$$n_{lq} = \sum_{i=1}^n Q_{li}.$$

本文所研究的装配序列优化问题的目标函数是,最大化满足几何约束条件下的装配次数与最小化装配方向改变次数,即

$$\max f = \max_l \{w_1 n_{lp} + w_2 (n - n_{lq})\},$$

其中 $w_1$ 、 $w_2$ 为两个目标函数的权重.

零件之间的互相影响主要来源于违反几何约束而导致的碰撞.若待装配零件没有稳固支撑,则导致零件移位;若装配方向改变,则导致装配过程不流畅.因此,最优的装配序列应在满足几何约束次数最大化、装配方向改变次数最小化的基础上,确保每一个待装配零件具有稳固支撑.因而,本文将基于装备方向改变次数、满足几何约束装配次数及稳固支撑个数3个指标对装配序列质量进行评价.待装配的零件 $p_{li}$ 是否有稳固的支撑可以通过变量 $s_{li}$ 表示,若有稳固支撑,则 $s_{li} = 1$ ,否则 $s_{li} = 0$ .装配序列 $X$ 中所有具有稳固支撑的零件个数记为 $n_{ls} = \sum_{i=1}^n s_{li}$ .定义第1个装配零件具有稳固支撑,即 $s_{l1} = 1$ .由于随着装配方向改变次数、满足几何约束装配次数及稳固支撑个数对装配质量均呈指数影响,可以建立装配质量评价指

标值

$$e = \alpha_1 e^{n_{1p}/n} + \alpha_2 e^{n_{1q}/n} + \alpha_3 e^{n_{1s}/n},$$

其中  $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$  为各自的权重系数. 指标值  $e$  越大, 说明装配质量越高.

## 2 算法设计

### 2.1 基本粒子群算法

粒子群算法 (PSO) 是由 Kennedy 等<sup>[9]</sup>受鸟群觅食行为的启发而提出的, 通过粒子的位置和速度更新寻找最优解, 更新过程同时受到当前个体最优解和全局最优解的共同引导. 假设第  $i$  个粒子在第  $t$  代的位置和速度分别为  $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{in}^t)$  和  $V_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{in}^t)$ , 第  $i$  个粒子在第  $t$  代的个体最优解为  $P_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{in}^t)$ , 当前的全局最优解为  $P_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gn}^t)$ , 则速度和位置的更新公式为

$$v_{ij}^{t+1} = wv_{ij}^t + c_1 r_1 (p_{ij}^t - x_{ij}^t) + c_2 r_2 (p_{gj}^t - x_{ij}^t), \quad (2)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}. \quad (3)$$

其中:  $w$  为惯性权重,  $c_1$  和  $c_2$  为学习因子,  $r_1$  和  $r_2$  为  $[0, 1]$  之间的随机数. 本文采用线性变化的惯性权重

$$w = w_{\max} - t(w_{\max} - w_{\min})/D_{\max}. \quad (4)$$

其中:  $w_{\max} = 0.9$  为最大权重值,  $w_{\min} = 0.1$  为最小权重值,  $t$  为当前迭代次数,  $D_{\max}$  为粒子群算法最大迭代次数.

### 2.2 基于值变换的粒子位置和速度更新规则

假设粒子  $i$  的位置为  $X_i = [x_{i1}, \dots, x_{ij}, \dots, x_{in}]$ . 其中:  $x_{ij} = (p_{ij}, d_{ij})$ ;  $p_{ij} = 1, 2, \dots, n$ ;  $d_{ij} = 1, 2, \dots, 6$ . 粒子  $i$  的速度为  $V_i = [v_{i1}, \dots, v_{ij}, \dots, v_{in}]$ . 其中:  $v_{ij} = (p_{ij}, d_{ij})$ ;  $p_{ij}$  为零件的编号, 与粒子位置不同的是  $p_{ij}$  为  $[0, n]$  之间的随机整数.

在式 (2)、(3) 中, 粒子位置和速度的操作有 4 项, 分别重新定义如下.

#### 1) 减法运算 (粒子位置和位置求差).

假设粒子  $X_1$  和  $X_2$  分别表示为  $X_1 = [x_{11}, \dots, x_{1j}, \dots, x_{1n}]$  和  $X_2 = [x_{21}, \dots, x_{2j}, \dots, x_{2n}]$ , 由粒子位置的减法运算可得到速度  $V = X_2 - X_1$ . 速度向量  $V_1$  的第  $j$  个元素的值定义为

$$v_j = \begin{cases} x_{1j}, & x_{1j} \neq x_{2j}; \\ 0, & x_{1j} = x_{2j}. \end{cases}$$

#### 2) 加法运算 (粒子位置和速度求和).

在式 (3) 中, 粒子的下一个位置由当前位置与速度的求和获得. 重新定义的位置更新为

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t, & v_{ij}^{t+1} = 0; \\ v_{ij}^{t+1}, & v_{ij}^{t+1} \neq 0. \end{cases}$$

#### 3) 倍乘运算 (速度的倍乘).

在式 (2) 中, 为获得下一个速度, 需要将速度与惯性权重求和. 假设速度  $V_1 = [v_{11}, \dots, v_{1j}, \dots, v_{1n}]$ , 常数  $c \in [0, 1]$ . 速度的倍乘为  $V_2 = c \times V_1$ . 速度  $V_2$  的第  $j$  个元素的值定义为

$$v_{2j} = \begin{cases} v_{1j}, & \text{rand} \leq c; \\ 0, & \text{else.} \end{cases}$$

其中 rand 为  $[0, 1]$  之间均匀分布的随机数.

#### 4) 加法运算 (速度和速度的求和).

在式 (2) 中, 为得到新的速度, 需要进行速度之间的求和. 假设有速度  $V_1$  和  $V_2$ , 新的速度为  $V_3 = V_1 + V_2$ , 速度  $V_3$  的第  $j$  个元素的值定义为

$$v_{3j} = \begin{cases} v_{1j}, & v_{1j} \neq 0, v_{2j} = 0; \\ v_{1j}, & v_{1j} \neq 0, v_{2j} \neq 0, \text{ and } \text{rand} \leq \text{const}; \\ v_{2j}, & \text{else.} \end{cases}$$

其中: rand 为  $[0, 1]$  之间均匀分布的随机数; const 为常数, 可以看出  $v_{3j}$  依赖  $v_{2j}$  或者  $v_{1j}$  的程度取决于 const 的值, 故取 const = 0.5.

### 2.3 初始化粒子种群

ASP 问题的初始解若完全随机产生, 则装配序列极有可能是杂乱无章的, 因此, 本文提出具有随机性的启发式产生初始解, 用于提高初始种群的质量及算法搜索效率. 该方法描述如下.

Step 1: 根据各方向干涉矩阵  $IM_k$ , 计算带方向的干涉矩阵  $DIM = \sum_{k=1}^6 IM_k$ .

Step 2: 随机选择一个装配方向  $d_1$ , 选择与其他所有零件在  $d_1$  方向拥有最少干涉次数的零件作为  $p_1$ , 即

$$p_1 = \{p_i | DIM(p_i, d_1) \leq DIM(p_k, d_1), k = 1, 2, \dots, n\}.$$

Step 3: 随机产生  $n$  个  $[1 - n]$  的随机整数 ( $n$  为零件个数), 记为  $a[n]$ .

Step 4: 依次将  $a[1] \sim a[n]$  赋值给  $p_2 \sim p_n$ , 当  $a[i] = p_1$  时, 从  $a[i + 1]$  继续赋值, 直到  $p_n$  被赋值.

Step 5: 随机产生  $p_2 \sim p_n$  的装配方向  $d_2 \sim d_n$ .

Step 6: 重复 Step 1 ~ Step 5, 直至初始种群所有粒子均被赋值.

### 2.4 深度邻域搜索的改进策略

由于基本粒子群算法容易陷入局部最优, 影响算法寻找全局最优解的能力. Somayé 等<sup>[8]</sup>研究结果表明, ASP 问题存在多个局部最优解且近似均匀地分布在解空间中. 本文根据 ASP 问题解的形式, 基于 4 种增强邻域搜索方式, 提出一种带有深度邻域搜索改进

策略的粒子群算法.

4种深度邻域搜索方式详细描述如下.

1) 反转 (flip): 随机选择当前序列中的一个零件, 随机改变其装配方向;

2) 交换 (exchange): 随机选择当前序列中的两个零件, 交换其装配顺序和装配方向;

3) 插入 (insertion): 随机选择当前序列中的一个零件, 重新安排其顺序, 其余零件依次后移;

4) 倒置 (inversion): 随机选择当前序列的两个零件, 将两个零件之间的所有零件颠倒顺序.

在粒子更新过程中, 若粒子的适应度函数值连续  $I_{\max}$  代没有更新, 则认为该粒子陷入了局部最优, 这时需要选择邻域搜索的操作方式. 上述4种邻域搜索方式有很明显的强弱之分, 即从1)~4), 粒子位置的改变逐渐增强, 相应的跳出局部最优的能力也逐渐增强. 为粒子  $i$  设置计数器  $e_i$ , 随着  $e_i$  的增大, 该粒子陷入局部最优的程度越大, 则相应进行更深度邻域搜索. 本文粒子群最大迭代次数设为  $D_{\max}$ , 粒子的最大未改进代数记为  $LD_{\max}$ , 且  $LD_{\max} = D_{\max}/10$ . 根据  $e_i LD_{\max}$  的值决定邻域搜索采用的方式.

改进的粒子群算法流程如下.

Step 1: 初始化算法各参数、种群规模、迭代次数、学习因子、惯性权重以及各方向干涉矩阵.

Step 2: 产生初始种群. 计算每个粒子的适应度函数, 得到初始个体的最优和全局最优.

Step 3: 基于值变换的粒子位置和速度更新规则为每个粒子更新速度和位置, 计算更新后的适应度函数, 更新个体的历史最优.

Step 4: 根据  $e_i/LD_{\max}$  的值决定邻域搜索采用的方式并进行深度邻域搜索.

若  $0.1 < e_i/LD_{\max} < 0.25$ , 则采用反转策略;

若  $0.25 < e_i/LD_{\max} < 0.50$ , 则采用反转、交换策略;

若  $0.50 < e_i/LD_{\max} < 0.75$ , 则采用反转、插入策略;

若  $0.75 < e_i/LD_{\max} < 1$ , 则采用反转、倒置策略.

Step 5: 增加迭代次数, 若  $t < D_{\max}$ , 则转 Step 3, 否则迭代终止, 输出装配序列规划作为最终解.

### 3 实例验证

下面通过实例来验证本文算法的有效性. 由11个零件组成的装配体<sup>[8]</sup>二维结构如图1所示.

算法参数设置如下: 学习因子  $c_1 = c_2 = 2$ . 粒子种群规模  $K = 80$ , 最大迭代次数  $D_{\max} = 100$ . 利用C语言编写程序, 以2.0GRAM, 2.20GHz CPU的PC机进行实验. 将该算法与文献[8]中提到的BLS算法、

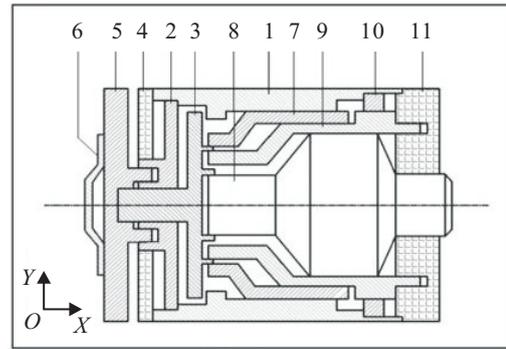


图1 产品结构方框图

模拟退火算法(SA)、遗传算法(GA)、文化基因算法(MA)、免疫算法和粒子群结合算法(IPSO)进行比较.

算法比较结果如表1所示. 表1中列出了相应算法产生的装配序列目标函数值  $f$  及其所对应的方向改变次数  $n_q$  和满足几何约束的装配次数  $n_p$ . 本文提出的带有深度邻域搜索策略的粒子群算法在迭代100次的情况下可以得到最优函数值为110的最优装配序列, 装配方向改变次数为0次, 装配过程满足几何约束11次, 即每个零件的装配操作都不对其他零件产生干涉. 运行30次, 除有1次得到函数值为109的序列外, 其余都可产生函数值为110的序列, 表明该算法具有较好的稳定性. 本文所提出的算法与文献[8]中的BLS、SA、GA、MA、IPSO算法相比, 在求解结果上均具有优势, 收敛速度快. 与当前最好的BLS算法相比, 平均方向改变次数改进了11.86%, 平均满足几何约束的装配次数改进了4.79%. 求解过程收敛情况如图2所示, 可见该算法具有较快且稳定的收敛速率.

表1 算法比较结果

算法	目标值 $f$	$n_q$	$n_p$	平均 $f$	平均 $n_q$	平均 $n_p$
本文算法	110	0	11	109.7	1.56	10.72
BLS	110	1	11	109.7	1.77	10.23
SA	109	1	11	108.8	3.53	10.72
GA	109	1	11	102.6	2.81	9.51
MA	109	1	11	109.0	1.34	10.21
IPSO	109	1	11	108.4	3.23	10.94

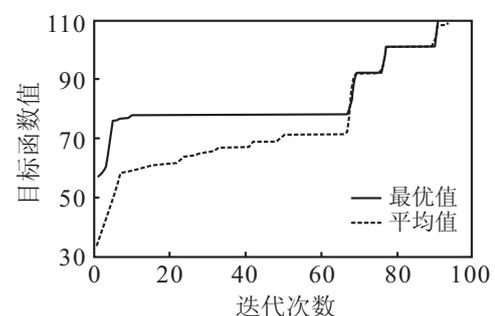


图2 算法求解过程收敛曲线

本文所提出的算法与 BLS 算法得到的装配序列分别为

$$[(11, +x)(8, +x)(9, +x)(7, +x)(10, +x)(1, +x)(3, +x)(2, +x)(4, +x)(5, +x)(6, +x)],$$

$$[(6, -x)(5, -x)(4, -x)(2, -x)(3, -x)(1, -x)(10, -x)(7, -x)(9, -x)(8, -x)(11, -x)],$$

但目标值相同, 均为 110. 为了进一步评价两种不同算法得到的装配序列质量, 现对所得装配序列进行评价, 以选择出对应的装配质量最高的序列进行生产. 取  $\alpha_1 = 10, \alpha_2 = \alpha_3 = 1$ , 对应评价结果如表 2 所示. 结果表明, 本文提出的算法在解决 ASP 问题上具有良好的有效性和稳定性, 装配质量较高.

表 2 装配序列质量比较结果

算法	$n_{lq}$	$n_{lp}$	$n_{ls}$	质量指标值 $e$
本文算法	0	11	10	54.72
BLS	1	11	5	45.65

#### 4 结 论

对于机械制造企业, 合理有效地优化装配序列环节, 不仅能够提高装配车间的装配效率, 缩短生产周期, 而且能够提高生产效率, 降低生产成本. 本文通过对产品装配结构分析, 以各零件之间的干涉关系作为几何约束, 以最大化满足几何约束次数和最小化装配方向改变次数为目标, 建立了产品装配序列优化模型. 利用值变换的粒子位置和速度更新规则, 基于带有随机性的启发式算法产生初始种群, 提出了带有深度邻域搜索策略的粒子群算法解决装配序列优化问题. 通过实例对算法进行验证, 将求解结果与现有的算法进行了比较, 所得结果表明, 算法稳定性较好, 收敛速度较快, 得到的最优装配序列质量较高. 在不同企业的实际装配过程中, 产品具有很强的多样性, 零件的数量可能很大, 带有可形变的零件的产品装配序列优化问题是今后的研究方向.

#### 参考文献(References)

- [1] Romeo M M, Lee H S, Kazen A. A genetic algorithm for the optimization of assembly sequences[J]. Computers & Industrial Engineering, 2006, 50(4): 503-527.
- [2] Wang J F, Liu J H, Zhong Y F. A novel ant colony algorithm for assembly sequence planning[J]. Int J of Advanced Manufacturing Technology, 2005, 25(11): 1137-1143.
- [3] Gao L, Qian W R, Li X Y, et al. Application of mimetic algorithm in assembly sequence planning[J]. Int J of Advanced Manufacturing Technology, 2010, 49(9): 1175-1184.
- [4] Lü H G, Lu C. An assembly sequence planning approach with a discrete particle swarm optimization algorithm[J]. Int J of Advanced Manufacturing Technology, 2010, 50(5): 761-770.
- [5] Wang Y, Liu J H. Chaotic particle swarm optimization for assembly sequence planning[J]. Robotics and Computer-Integrated Manufacturing, 2010, 26(2): 212-222.
- [6] Li M Y, Wu B, Hu Y M, et al. A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation[J]. Int J of Advanced Manufacturing Technology, 2013, 68(1): 617-630.
- [7] Zhang H Y, Liu H J, Li L Y. Research on a kind of assembly sequence planning based on immune algorithm and particle swarm optimization algorithm[J]. Int J of Advanced Manufacturing Technology, 2014, 71(5): 795-808.
- [8] Somayé G, Ellips M. A breakout local search(BLS) method for solving the assembly sequence planning problem[J]. Engineering Applications of Artificial Intelligence, 2015, 39(3): 245-266.
- [9] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of the IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.

(责任编辑: 李君玲)