

粒子群算法在装配序列规划中的应用方法研究

耿育科¹, 孙荟博²

(1. 中航飞机西安飞机分公司, 西安 710089;
2. 上海中信信息发展股份有限公司, 上海 200333)

[摘要] 将粒子群优化算法引入到装配序列规划中, 详细讨论了粒子群算法在装配规划中的数学表示, 给出了适应度函数评价方法, 并实现了装配干涉矩阵的构建, 以发动机化油器装配为例, 分析了粒子群算法在装配序列规划中的具体应用。

关键词: 装配序列规划; 粒子群算法; 智能优化算法

Study on Application Method of Particle Swarm Optimization Algorithm on Assembly Sequence Planning

GENG Yuke¹, SUN Huibo²

(1. AVIC Xi'an Aircraft Industry (Group) Company LTD., Xi'an 710089, China;
2. Shanghai CITIC Information Development Limited by Share Ltd., Shanghai 200333, China)

[ABSTRACT] The particle swarm optimization (PSO) algorithm, which is a kind of intelligent optimization algorithm, is introduced to solve the assemble sequence planning (ASP) problem. The mathematical representation of PSO in ASP is discussed in detail. The evaluation method of fitness function is presented and the matrices of assembling interference are constructed also. The engine carburetor assembly is applied as an example to analyze the application of particle swarm algorithm in assembly sequence planning.

Keywords: ASP (assembly sequence planning); PSO (particle swarm optimization); Intelligent optimization algorithm

DOI: 10.16080/j.issn1671-833x.2016.09.066

装配序列规划(Assembly Sequence Planning, 简称 ASP)是产品生产过程中非常重要的一个环节。装配序列的优劣直接影响了产品装配的质量、装配的效率以及成本, 甚至影响可装配性^[1]。为了提高找寻可行装配序列的效率, 一些智能优化算法被应用到装配序列规划中。Lazzerini 等^[2]将遗传算法应用到了装配序列规划中。Ong^[3]和 Milner^[4]等在各自的研究中将模拟退火算法应用到了装配序列规划中, 不足之处是单独的模拟退火算法对最优搜寻空间的寻找能力表现不佳。Wang^[5]和 Failli^[6]等提出了一种结合蚁群优化算法的装配序列规划方式。

粒子群算法(Particle Swarm Optimization, PSO)具有便于实现、占用计算机内存少、全局搜寻能力强等特点, 在解决不同的优化问题时有着良好的表现, 吸引了很多工程应用领域内学者的注意^[7-10], 因此本文采用粒子群算法进行装配序列的规划。

1 粒子群算法在装配序列规划中的应用

将粒子群算法引入到装配序列规划当中, PSO 中的粒子对应每一条装配序列, 然后根据装配过程中影响成本和时间因素, 构建适应度函数, 作为序列的评判标准, 并且根据几何关系构建数学模型, 确保装配序列的可行性, 对不满足的序列实行惩罚, 降低其作为最优解的可能。

1.1 适应度函数评价

影响装配序列规划的因素有很多, 而装配过程中装配方向或者装配工具的改变通常会导致装配时间以及装配成本的增加, 因此本文将工具变换次数、装配方向变换次数以及装配操作类型变换次数作为衡量一个装配序列的标准, 加入到适应度函数之中。

根据 Lu 等^[11]的研究, 对于一个装配产品, 若给出了一个可行的装配序列, 那么该装配序列所对应的适应

度函数为:

$$F = 2S - \omega_1 n_1 - \omega_{or} n_{or} - \omega_{op} n_{op}, \quad (1)$$

式中, S 为装配体中的零部件总数量; n_1 、 n_{or} 以及 n_{op} 分别代表了装配工具的变换次数、装配方向的变换次数和装配操作的变换次数; ω_1 、 ω_{or} 和 ω_{op} 分别为 n_1 、 n_{or} 以及 n_{op} 的权重, 值为在 $[0, 1]$ 上平均分布的随机数。可以看出, n_1 、 n_{or} 和 n_{op} 越小, 即装配成本和时间越小, 适应度函数 F 的值越大。

对于一个装配产品, 给出了一个不可行的装配序列, 即装配序列不满足装配干涉关系的时候, 其对应的适应度函数值按式(2)求取, m 为该装配序列中不满足干涉关系的次数, 有:

$$F = (2S - \omega_1 n_1 - \omega_{or} n_{or} - \omega_{op} n_{op}) / 2m. \quad (2)$$

1.2 装配干涉矩阵的构建

产品的可装配性是装配规划的第一要求。在算法执行的过程中, 产品的装配干涉矩阵是判断某一条装配序列是否可行的依据, 对于不符合装配干涉矩阵的序列, 视为不满足约束条件, 进行惩罚处理^[12]。

根据装配体零部件之间的几何关系, 构建在三维空间上装配干涉矩阵。装配干涉矩阵以 M_a 表示, $a \in (\pm X, \pm Y, \pm Z)$, 则一个具有 d 个零部件的装配体在方向 a 上的装配干涉矩阵可表示为:

$$M_a = \begin{matrix} & C_1 & C_2 & \dots & C_d \\ \begin{matrix} C_1 \\ C_2 \\ \dots \\ C_d \end{matrix} & \begin{pmatrix} C_{11} & C_{12} & \dots & C_{1d} \\ C_{21} & C_{22} & \dots & C_{2d} \\ \dots & \dots & \dots & \dots \\ C_{d1} & C_{d2} & \dots & C_{dd} \end{pmatrix} \end{matrix}, \quad (3)$$

式中, C_1, C_2, \dots, C_d 分别代表了装配体的 d 个零部件; $C_{ij}=1$ 表示零件 C_i 在方向 a 上, 往装配位置移动时, 将与零件 C_j 发生碰撞; 由于零件不与自身发生碰撞, 因此 $C_{ij}=0$; 由于 a 方向上的 C_{ij} 等同于 $-a$ 方向上的 C_{ji} , 因此取 3 个正方向上的干涉矩阵 M_{+X} , M_{+Y} 和 M_{+Z} , 即可对装配序列的可行性进行评价。

对于给出的装配序列, 通过提前定义, 每一步装配操作都对应了固定的工具以及装配操作类型, 从而使工具变换次数 n_1 和装配操作类型变换次数 n_{op} 较容易获得。而装配方向的变换次数 n_{or} 以及 m 可由 3 个装配干涉矩阵得出。

1.3 粒子群在装配序列规划中的数学表示

相对标准群粒子算法, 应用在装配序列规划中的 PSO 有着相同的执行步骤, 和不同的粒子位置、速度表达式以及更新公式。装配序列是一串标示着各个零件在装配过程中序号的数字, 因此最优解也应该分布在离散的正整数空间上。零件的总个数 d 即为粒子所在的维度^[13-14]。

粒子 i 的位置表示为:

$$x_i(t) = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}), \quad (4)$$

式中, x_{ij} 代表了粒子 i 在第 j 维度上的位置的值。 $x_i(t)$ 中任意两个元素的值都不相同, 且为正整数。在算法进行初始化的时候, 应该随机将 $1 \sim d$ 的正整数分配给 d 个元素。随机产生的 $x_i(t)$ 有利于提高种群的多样性, 增强算法在迭代初期的局部寻优能力。

粒子 i 的速度表达式与位置表达式形式上相近:

$$v_i(t) = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id}), \quad (5)$$

式中, v_{ij} 代表粒子 i 在第 j 维度上的速度的值。与粒子的位置一样, 随机产生初始速度, 并且 $v_i(t)$ 中的任意两个非零元素的值都不相同。

位置相减得到一个新的速度:

$$v = x_1 - x_2, \quad (6)$$

式中, x_1, x_2 为两个位置, 代表两个装配序列。对于第 j 维, x_{1j} 等于 x_{2j} , 则 $v_j=0$; 若 x_{1j} 不等于 x_{2j} , 则 $v_j=x_{1j}$, 即速度 v 继承了 x_1 的有效元素。

位置的更新如下:

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (7)$$

式中, $x_i(t)$ 为粒子在第 t 次迭代时的位置, $v_i(t+1)$ 为第 $t+1$ 次迭代时的速度, 两者相加得到第 $t+1$ 次迭代时的位置 $x_i(t+1)$ 。若第 j 维的速度 $v_{ij}(t+1)$ 等于零, 则 $x_{ij}(t+1)=x_{ij}(t)$; 若 $v_{ij}(t+1)$ 不等于零, 则交换 $x_{ij}(t)$ 与 $v_{ij}(t+1)$ 的值。

参数乘以速度用以调整速度, 控制速度 v_2 从 v_1 中继承到的元素个数, 如下:

$$v_2 = \begin{cases} v_1, r < c \\ 0, r \geq c \end{cases}, \quad (8)$$

式中, r 为 $[0, 1]$ 上平均分布的随机数。可知 c 的取值越大, v_2 继承 v_1 的元素越多。速度相加得到一个新的速度, 其各个维度上元素的值根据式(9)得出。

$$v_j = \begin{cases} v_{1j}, v_{1j} \neq 0, v_{2j} = 0 \\ v_{1j}, v_{1j} \neq 0, v_{2j} \neq 0, r < \text{const}(j = 1, 2, \dots, d), \\ v_{2j}, \text{else} \end{cases}, \quad (9)$$

式中, r 为 $[0, 1]$ 上平均分布的随机数, 常数 const 越大, v 从速度 v_1 中继承的元素越多。在本文中, 取 $\text{const}=0.5$ 。

粒子速度及位置更新如下:

$$v_i(t+1) = \omega v_i(t) + c_1(p_i(t) - x_i(t)) + c_2(p_g(t) - x_i(t)), \quad (10)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (11)$$

式中, ω 控制粒子现有速度对新速度的影响; c_1 控制粒子历史自身最优位置(自身最优装配序列)对新速度的影响; c_2 控制着粒子全局最优位置(全局最优装配序列)对新速度的影响; ω 、 c_1 和 c_2 在 $[0, 1]$ 范围内取值。式中

包含了位置与速度相加、位置相减、速度相加以及速度的乘法。式(6)~(9)所定义的特殊运算确保了装配序列的可行性。

2 案例研究与分析

以发动机化油器的装配为例,对粒子群算法在装配序列中的应用进行分析与验证,如图1、图2所示。图

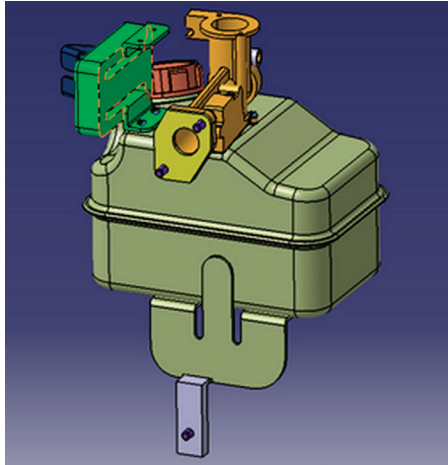


图1 发动机化油器装配体

Fig.1 Engine carburetor assembly

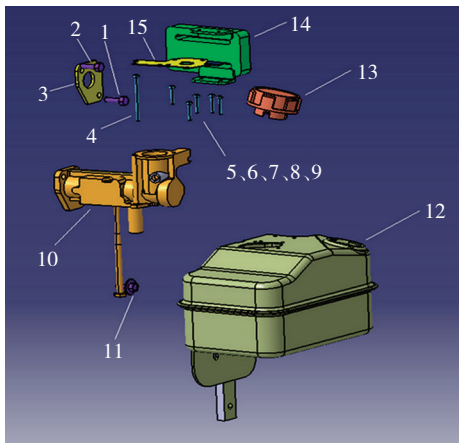


图2 发动机化油器展开图

Fig.2 Engine carburetor unfolded drawing

表1 发动机化油器零件

编号	零件名称	编号	零件名称
1	螺栓	9	螺栓
2	螺栓	10	化油器
3	垫片	11	螺栓
4	长螺栓	12	油箱
5	螺栓	13	邮箱盖
6	螺栓	14	风门
7	螺栓	15	垫片
8	螺栓		

2中,发动机化油器由15个零件组成,每个零件沿着±X、±Y、±Z轴有6个可能的装配方向。按图2中的编号,依次对应的零件如表1所示。

装配体所有零件沿着+X、+Y、+Z 3个坐标轴方向上的装配干涉矩阵为 M_{+X} 、 M_{+Y} 、 M_{+Z} :

$$M_{+X} = \begin{matrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \\ C_{10} \\ C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \\ C_{15} \end{matrix} \end{matrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M_{+Y} = \begin{matrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \\ C_{10} \\ C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \\ C_{15} \end{matrix} \end{matrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M_{+Z} = \begin{matrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \\ C_{10} \\ C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \\ C_{15} \end{matrix} \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

发动机化油器的零件装配过程中所用的装配工具以及装配操作类型如表 2 所示。

其中,零件装配工具更换次数、装配方向变换次数以及装配操作类型变换次数的权值分别设定为 $\omega_r=0.3$ 、 $\omega_{or}=0.4$ 、 $\omega_{op}=0.3$ 。总迭代次数 $t=100$,取式(9)中常数 $\text{const}=0.5$ 。算法各参数及适应度函数各评价指标的权值保持不变, ω 、 c_1 、 c_2 的取值均为 0.5,粒子种群容量分别取 10、20、40、60、80、100,在 Matlab 中编写、运行程序,不同种群容量的算法程序分别运行 20 次,获取优化或者近似优化解。根据适应度函数式(1)计算可知,对于可行的装配序列,其适应度函数值 F 的最大值为 28.3,因此优化过程中适应度函数值接近或等于 28.3 的装配序列即可认为是优化结果。

表 3 为种群容量大小取 100 时,适应度函数值取得最优值 28.3 的一条装配序列的相关信息。表 3 种群容量为 100 的最优装配序列。

从表 2 中可以看出,装配工具变换次数 $n_r=2$,装配方向变换次数 $n_{or}=2$,装配操作类型变换次数为 $n_{op}=1$ 。结合装配干涉矩阵 M_{+X} 、 M_{+Y} 、 M_{+Z} 可以得到:(1)零件 3 必须在零件 1 和零件 2 之前安装;(2)零件 15 必须在零件 10 之前安装;(3)零件 10 必须在零件 4、5、6、7 之前安装;(4)零件 14 必须在零件 8 和 9 之前安装;(5)装配过程中存在 3 个装配方向;(6)在寻找最优装配序列的过程中,装配方向的变换次数、装配工具变换次数与装配类型的变换次数,这 3 者之间往往存在着矛盾。

表2 零件装配工具及操作类型

零件编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
工具类型	1	1	2	1	1	1	1	1	1	3	1	3	3	3	2
操作类型	1	1	0	1	1	1	1	1	1	0	1	0	0	0	0

表3 种群容量为100的最优装配序列

序号	3	15	12	10	14	13	4	8	9	5	7	6	1	2	11
工具	2	2	3	3	3	3	1	1	1	1	1	1	1	1	1
类型	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
方向	Z	X	X	X	X	X	X	X	X	X	X	X	Z	Z	Z

3 结束语

本文根据装配序列规划的特点和要求,采用粒子群算法来解决装配序列规划问题。将装配过程工具变换次数、装配方向变换次数以及装配操作类型变换次数作为衡量一个装配序列的标准,加入到适应度函数之中,完成了适应度函数的构建。根据装配体零部件之间的几何关系,构建了三维空间上的装配干涉矩阵。最后通

过发动机化油器装配实例,验证了本文方法的可行性。

参考文献

- [1] 刘继红,王峻峰.复杂产品协同装配设计与规划[M].武汉:华中科技大学出版社,2011.
- [2] LIU Jihong, WANG Junfeng. Complex product collaborative assembly design and planning[M]. Wuhan: Huazhong University of Science and Technology Press,2011.
- [3] LAZZERINI B, MARCELLONI F. A genetic algorithm for generating optimal assembly plans[J]. Artificial Intelligence in Engineering, 2000, 14(4): 319-329.
- [4] ONG S K, DING J, NEE A Y C. Hybrid GA and SA dynamic set-up planning optimization[J]. International Journal of Production Research, 2002, 40(18): 4697-4719.
- [5] MILNER J M, GRAVES S C, WHITNEY D E. Using simulated annealing to select least-cost assembly sequences[C]//Proceeding of Robotics and Automation. New York: IEEE, 1994: 2058-2063.
- [6] WANG J F, LIU J H, ZHONG Y F. A novel ant colony algorithm for assembly sequence planning[J]. The International Journal of Advanced Manufacturing Technology, 2005, 25(11-12): 1137-1143.
- [7] FAILLI F, DINI G. Ant colony systems in assembly planning: a new approach to sequence detection and optimization[C]//Proceedings of the 2nd CIRP international seminar on intelligent computation in manufacturing engineering. 2000: 227-232.
- [8] CLERC M, KENNEDY J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.
- [9] LOVBJERG M, RASMUSSEN T K, KRINK T. Hybrid particle swarm optimiser with breeding and subpopulations[C]//Proceedings of the Genetic and Evolutionary Computation Conference. San Francisco, 2001: 469-476.
- [10] SHI Y, EBERHART R C. Empirical study of particle swarm optimization[J]. Academic Press, 1967.
- [11] NOEL M M, JANNETT T C. Simulation of a new hybrid particle swarm optimization algorithm[C]//Symposium on System Theory. New York: IEEE, 2004: 150-153.
- [12] LU C, WONG Y S, FUH J Y H. An enhanced assembly planning approach using a multi-objective genetic algorithm[J]. Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture, 2006,220(2):255-272.
- [13] BONNEVILLE F, PERRARD C, HENRIOD J M. A genetic algorithm to generate and evaluate assembly plans[C]//Proceedings of Emerging Technologies and Factory Automation. New York: IEEE,1995: 231-239.
- [14] WANG Junfeng, LI Shiqi, LIU Jihong, et al. Computer aided assembly planning: a survey[J]. Journal of Engineering Graphics, 2005,2:1-7.
- [15] 杨冰,张林鎔.面向装配工艺规划的集成式装配模型研究[J].系统仿真学报, 2008, 18(2):533-536.
- [16] YANG Bing, ZHANG Linxuan. Introduction of integrated assembly model for assembly process planning[J]. Journal of System Simulation, 2008,18(2):533-536.

(责编 大漠)