

基于集成干涉矩阵的蚁群装配序列规划

彭涛 李世其 王峻峰 方建新

(华中科技大学机械科学与工程学院 武汉 430074)

摘要 针对装配序列规划问题,构建了面向装配的蚁群算法。利用有向图进行产品的装配建模,有效地表达了装配零件之间的接触以及优先关系。探讨了装配干涉矩阵的形式和性质,提出了集成干涉矩阵的概念和变换方法,采用集成干涉矩阵以及工具列表矩阵来表达模型的基本装配信息。分析了基于干涉矩阵的可行方向推导过程、装配零件的状态转移概率以及蚁群算法的信息素更新规则。从装配可行方向的改变次数和装配工具的改变次数建立目标优化函数,从而建立面向装配的蚁群算法用于序列的搜索及构造,并给出蚁群算法的伪代码。最后通过实例对算法进行了验证说明。

关键词 蚁群算法,装配序列规划,有向装配图,集成干涉矩阵

中图分类号 TP312 **文献标识码** A

Integrated Interference Matrix Based Ant Colony Algorithm for Assembly Sequence Planning

PENG Tao LI Shi-qi WANG Jun-feng FANG Jian-xin

(School of Mechanical Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract An ant colony algorithm was constructed to search the best optimal assembly sequence rapidly in an assembly sequence planning. An assembly model for expressing the touching and preferential relationship among the assembly parts effectively was built by using directed assembly graph. On the basis of discussing the form and property of the assembly interference matrix, a concept of integrated interference matrix and its transformational methods was proposed. The integrated interference matrix and the tool list matrix can express the basic assembling information fully. The derivation of assembly feasible direction calculation based on the integrated interference matrix, the state transition probability of assembly parts, and the update rules of algorithm information were analyzed simultaneously. According to the changing frequency of feasible direction and the changing time of the assembly tools, the optimized functions for ant colony algorithm were then established and two examples were also proposed to prove the practicability of the algorithm.

Keywords Ant colony algorithm, Assembly sequence planning, Directed assembly graph, Integrated interference matrix

装配序列规划是装配规划研究中的一个重要环节。目前对装配序列规划的研究方法主要分为 3 大类:基于图论和经验的方法^[1]、基于启发式搜索的智能算法的方法、基于虚拟现实的方法。第一类方法主要是基于图论的知识,利用 AND/OR 图^[2]、Petri 网^[3,4]、割集等方法来搜索序列。这些方法在产品装配零件个数比较多的情况下,由于实体信息,配合约束信息的急剧膨胀,容易产生装配序列规划的组合爆炸现象。随着智能算法的出现,用智能算法搜索装配序列的方法日渐成熟,并且智能算法能有效避免装配序列等组合爆炸现象^[5,6]。基于虚拟现实的方法虽能有效降低装配规划问题的复杂度、提高装配规划的效率,但其思想仍然是沿用一些基本的装配规划方法,并且由于是基于虚拟现实技术,其应用在当前也受到一定的限制^[7,8]。

1 基于有向图的装配模型

装配规划系统一般是在 CAD 模型的基础上建立的便于

序列规划的数学模型,然后对问题进行描述和求解^[9]。建模过程中的一个重要任务就是确定零件之间的相互位置关系。因此,记录零件之间相互位置关系对于装配序列规划有重要意义^[10]。本文在建立装配模型之前,对装配模型作如下假设:

- 1) 装配体是以零件形式进行装配的,不考虑子装配体的装配。
- 2) 零件之间的装配信息作为已知条件,并以此为基础建立装配模型。
- 3) 装配信息只考虑零件的装配方向以及装配工具,其它装配信息暂不作考虑。

1.1 有向装配图定义

有向装配图在有向图的基础上,对有向边赋予装配语意,以表示一个装配操作。有向装配图的每条边代表一次装配操作,边的两端节点分别表示装配过程连接的两个零件。定义有向装配图为:

到稿日期:2009-05-20 返修日期:2009-07-21 本文受国防基础科研资助项目(B1420060173),国家高技术研究发展计划项目(2006 AA04Z138)资助。

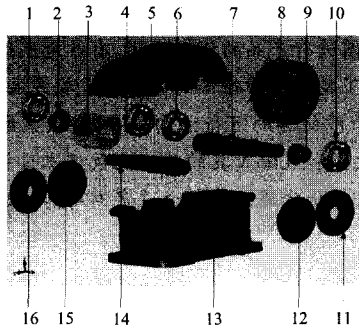
彭涛(1979-),男,博士生,主要研究方向为智能算法与虚拟装配等,E-mail:pntao@smail.hust.edu.cn;李世其(1965-),男,教授,主要研究方向为虚拟样机建模与仿真等;王峻峰(1970-),男,副教授,主要研究方向为虚拟设计与制造等;方建新(1981-),男,主要研究方向为系统建模与优化等。

$$G=(V,E) \quad (1)$$

其中, G 表示装配有向图, V 表示有向图的顶点零件的集合, 即 $V=\{p_1, p_2, \dots, p_n\}$, E 表示有向图的有向边, 即一个装配操作。

装配操作的过程用含有装配方向和装配工具的信息代替^[10], 装配操作信息的描述如下:

$$AO=\{p, d, t\} \quad (2)$$



注: 1-轴承; 2-套筒; 3-小齿轮; 4-轴承; 5-上箱体; 6-轴承; 7-轴; 8-大齿轮; 9-套筒; 10-轴承; 11-轴承盖; 12-轴承盖; 13-下箱体; 14-轴; 15-轴承盖; 16-轴承盖

图1 减速器爆炸图

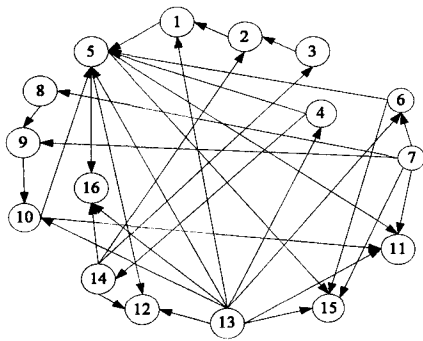


图2 减速器有向装配图

其中, AO 表示零件的装配操作, p 表示有向图的节点零件, d 表示装配该零件的可行方向, t 表示装配该零件所需的工具。

对于 d , 本文只考虑笛卡儿坐标系的 6 个方向, 即 d 为方向集合中的一个可行的装配方向。对于一个由 N 个零件 P_1, P_2, \dots, P_n 组成的装配体, 根据零件之间优先装配及其联接的关系, 可建立相应的有向图表达^[9]。下面是一个减速器的有向图表示示例。该减速器由 16 个主要零部件组成, 其中未考虑螺栓等紧固件。该装配体的爆炸图及零件的编号如图 1 所示, 其初始装配信息有向图表示如图 2 所示。有向图中的箭头表示两个零件之间有连接接触关系, 并且箭头尾端的零件必须优先进行装配。从有向图中可以看出装配中的接触及优先关系, 但是对装配方向和装配工具等装配信息的表示还不够, 因此必须给出装配信息的干涉矩阵表达式。

1.2 有向装配图的矩阵表达

装配过程中零件之间不能有干涉发生。在某一方向上干涉矩阵 (IM: Interference Matrix) 的概念最早由 Dini G. 和 Santoch M.^[11] 提出, 代表着每个零件在某一方向进行装配与其他零件之间的干涉情况。一般形式的干涉矩阵, 通常是按照空间 $\pm X, \pm Y, \pm Z$ 6 个方向分别建立的。如果直接按照这种方法来建立干涉矩阵, 对于一个装配体, 必然要建立 6 个干涉矩阵, 这样数据处理复杂且存在一些信息冗余。为了简化干涉矩阵的求解过程, 将 6 个方向的干涉集成在一起, 并命名为集成干涉矩阵^[5]。假设装配体由 n 个零件组成, 那么集成干涉矩阵是一个 n 行, $3 \times n$ 列的矩阵, 可以表示为:

$$IM = \begin{matrix} P_1 & P_2 & \dots & P_n \\ \begin{matrix} P_1 \\ P_2 \\ \dots \\ P_n \end{matrix} \begin{bmatrix} I_{11x} I_{11y} I_{11z} & I_{12x} I_{12y} I_{12z} & \dots & I_{1nx} I_{1ny} I_{1nz} \\ I_{21x} I_{21y} I_{21z} & I_{22x} I_{22y} I_{22z} & \dots & I_{2nx} I_{2ny} I_{2nz} \\ \dots & \dots & \dots & \dots \\ I_{n1x} I_{n1y} I_{n1z} & I_{n2x} I_{n2y} I_{n2z} & \dots & I_{nx} I_{ny} I_{nz} \end{bmatrix} \end{matrix} \quad (3)$$

其中, I_{ijx} 表示零件 P_j 沿 $+X$ 方向装配时与零件 P_i 的干涉情况: 如果零件 P_j 沿 $+X$ 方向装配时与零件 P_i 发生干涉, 则 $I_{ijx}=1$; 否则, $I_{ijx}=0$ 。零件 P_j 沿 $-X$ 方向装配与零件 P_i 干涉的情况和零件 P_i 沿 $+X$ 方向装配与零件 P_j 干涉的情况相同, 由元素 I_{jix} 表示。同样, 可以确定 I_{ijy} 和 I_{ijz} 的值。图 1 中减速器装配体的集成干涉矩阵如图 3 所示。

序号	1(x,y,z)	2(x,y,z)	3(x,y,z)	4(x,y,z)	5(x,y,z)	6(x,y,z)	7(x,y,z)	8(x,y,z)	9(x,y,z)	10(x,y,z)	11(x,y,z)	12(x,y,z)	13(x,y,z)	14(x,y,z)	15(x,y,z)	16(x,y,z)
1	000	011	110	000	010	100	011	000	000	000	000	000	000	000	000	000
2	000	011	011	000	011	000	000	011	000	011	000	000	000	000	000	000
3	011	000	000	000	000	000	011	000	100	100	000	000	011	011	000	000
4	000	100	000	000	000	000	000	000	000	000	000	000	000	000	011	000
5	000	000	000	100	000	000	000	000	000	000	000	000	000	000	000	000
6	000	000	011	000	011	011	000	000	000	000	100	000	000	000	000	000
7	100	100	000	000	000	000	000	000	000	000	000	000	000	000	000	000
8	011	000	011	000	011	000	000	011	000	000	000	000	000	000	000	000
9	000	011	000	000	011	000	000	000	011	011	000	000	000	000	000	000
10	000	000	000	011	000	000	000	000	000	000	111	000	000	100	000	000
11	011	000	000	011	011	011	000	000	000	000	000	000	000	000	000	000
12	000	000	000	000	000	011	000	000	011	011	011	000	000	000	000	000
13	000	100	100	000	000	000	000	000	000	000	000	000	000	000	000	000
14	100	000	000	100	011	000	000	100	000	100	100	000	000	000	000	000
15	000	000	000	100	000	000	000	000	000	000	000	000	100	000	000	000
16	000	000	000	000	000	100	100	000	000	000	000	100	000	000	000	000

图3 减速器集成干涉矩阵

1.3 集成干涉矩阵的变换与求解

由于干涉矩阵中的元素值只有 0 和 1 两种情况, 因此可以视其为一个二进制矩阵。为了减少矩阵计算元素的个数和

简化求解过程, 我们采取了先用十进制干涉矩阵来表达矩阵, 然后将得到的十进制干涉矩阵转换成二进制矩阵。每个零件在十进制干涉矩阵中的元素的值代表该零件在二进制干涉矩

阵中+X,+Y,+Z 3个方向值的十进制转换。因此十进制干涉矩阵的元素值只有 2^3 个,也即0,1,2,3,4,5,6,7这8种情况。十进制矩阵形式如下:

$$IIM_{10} = \begin{matrix} & P_1 & P_2 & \cdots & P_n \\ \begin{matrix} P_1 \\ P_2 \\ \cdots \\ P_n \end{matrix} & \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1n} \\ I_{21} & I_{22} & \cdots & I_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ I_{n1} & I_{n2} & \cdots & I_{nn} \end{bmatrix} \end{matrix} \quad (4)$$

为求解所需的集成干涉矩阵,本文采取以下求解步骤:

(1)首先求解装配体的十进制干涉矩阵;(2)将得到的十进制干涉矩阵转换成对应的二进制矩阵,即集成干涉矩阵。零件在+X,+Y,+Z 3个方向的干涉值设定如下:在X轴上发生干涉时,十进制干涉值设置为4;在Y轴上发生干涉时,十进制干涉值设置为2;在Z轴上发生干涉时,十进制干涉值设置为1;没有发生干涉时,干涉值均为0;如表1所列。

表1 干涉方向的十进制/二进制表达

干涉方向	十进制表达	二进制表达
X轴	4	100
Y轴	2	010
Z轴	1	001
无干涉	0	000

当零件 P_i 在X方向装配时与零件 P_j 发生干涉,十进制干涉矩阵 IIM_{10} 中的元素值计算公式如下:

$$\begin{aligned} \text{在X轴上发生干涉: } I_{ij} &= I_{ij} \cup 4 \\ \text{在Y轴上发生干涉: } I_{ij} &= I_{ij} \cup 2 \\ \text{在Z轴上发生干涉: } I_{ij} &= I_{ij} \cup 1 \end{aligned} \quad (5)$$

1.4 装配可行方向的推导

求解零件的可行装配方向,首先需要建立待装配零件相对于已装配零件的干涉向量。假设零件 C_i 是待装配零件,SA是在当前装配状态下由 m 个已装配好的零件组成的临时子装配体, $SA = \{p_1, p_2, \dots, p_m\}$ 。

$$\exists C_j \in SA, \forall \alpha_{ij} = \{I_{ijx} I_{ijy} I_{ijz} I_{jix} I_{jiy} I_{jiz}\} \quad (6)$$

其中, α_{ij} 就是零件 C_i 相对于子装配体SA的装配干涉向量,装配干涉向量中的元素都可以在装配体集成干涉矩阵中得到。为了方便描述待装配零件的可行装配方向,我们定义了装配方向向量 β_i :

$$\beta_i = \{+I_{ix} \ +I_{iy} \ +I_{iz} \ -I_{ix} \ -I_{iy} \ -I_{iz}\} \quad (7)$$

β_i 中的每个元素就分别表达该零件在+X,+Y,+Z,-X,-Y,-Z方向上是否是可装配的。某元素的值为0,表示待装零件在该方向是可装配的;若为1,则表示待装配零件在该方向上装配将发生干涉。 $+I_{ix}, +I_{iy}, +I_{iz}, -I_{ix}, -I_{iy}, -I_{iz}$ 的值由下列公式计算得到:

$$\begin{aligned} +I_{ix} &= \bigcup_{j=1}^{i-1} I_{jix}, \quad -I_{ix} = \bigcup_{j=1}^{i-1} I_{ijx} \\ +I_{iy} &= \bigcup_{j=1}^{i-1} I_{jiy}, \quad -I_{iy} = \bigcup_{j=1}^{i-1} I_{ijy} \\ +I_{iz} &= \bigcup_{j=1}^{i-1} I_{jiz}, \quad -I_{iz} = \bigcup_{j=1}^{i-1} I_{ijz} \end{aligned} \quad (8)$$

在上述公式中,“ \cup ”表示二进制的“或”操作。待装零件的装配方向向量 β_i 中0值元素所对应的装配方向,组成了待装零件的可行装配方向集 FD 。如果可行装配方向集 $FD = \Phi$,则表示待装配零件在当前状态下没有可行的装配方向。在需要判别的装配序列中,如果存在这样一个没有可行装配方向的零件,则该序列就整体不可行,若在序列中存在多个没

有可行装配方向的零件,则会发生多次干涉。

2 面向装配序列规划的蚁群算法

2.1 蚁群算法原理

自然界中真实蚂蚁具有找到巢穴到食物源之间最短路径的能力,这种能力是通过蚂蚁个体之间的协作来实现的,而蚂蚁个体之间采用的通信方式是一种称为信息素的挥发性分泌物。蚂蚁的初始行走是完全随机的,并在其走过的路径上留下信息素,后来的蚂蚁根据路径上信息素的强度来决定自己行走的方向,一条路径被选择的概率和当时路径上的信息素强度成正比。

2.2 转移概率的定义

蚁群中的蚂蚁在搜索操作序列时从一个装配操作转到下一个装配操作的概率,称为转移概率。设蚂蚁数量为 M ,蚂蚁从一个节点到下一个节点是根据概率大小来选择的。假定蚂蚁从上一次到达的节点转到当前节点的轨迹为 (i, j) ,那么转移概率就由该轨迹上的信息素浓度 τ_{ij} 、方向引导因子 d_{ij} 、工具引导因子 t_{ij} 共同决定。转移概率 $P_{ij}(t)$ 定义如下:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [d_{ij}(t)]^\beta [t_{ij}(t)]^\mu}{\sum_{k \in allow_k} [\tau_{ik}(t)]^\alpha [d_{ik}(t)]^\beta [t_{ik}(t)]^\mu}, & j \in allow_k \\ 0, & \text{其他} \end{cases} \quad (9)$$

$$d_{ij}(t) = \begin{cases} 0.2, & \text{如果方向改变} \\ 1, & \text{否则} \end{cases} \quad (10)$$

$$t_{ij}(t) = \begin{cases} 0.2, & \text{如果工具改变} \\ 1, & \text{否则} \end{cases} \quad (11)$$

其中, $\tau_{ij}(t)$ 表示 t 时刻路径上的信息素浓度; $d_{ij}(t)$ 用以引导蚂蚁选择装配重定向次数较少的路径; $t_{ij}(t)$ 用以引导蚂蚁选择工具更换次数较少的路径; $allow_k$ 表示蚂蚁下一次可能访问 k 个可行装配的节点的集合;参数 α, β, μ 分别为信息素浓度、方向引导因子和工具引导因子的权值,均为非负数。

2.3 信息素更新规则

蚂蚁在走过的路径上留下信息素,以影响后来蚂蚁的搜索。为使得蚂蚁能够搜索到更广范围内的解,在构建序列的过程中,如果蚂蚁从装配操作 i 选择了装配操作 j ,则相应路径上的信息素可根据以下局部更新规则进行更新:

$$\tau_{ij}(t) = (1-\rho) \times \tau_{ij}(t) + \rho \times \tau_0 \quad (12)$$

其中, $0 < \rho < 1$ 是信息素的局部挥发率, τ_0 是初始信息素的大小。局部更新通过挥发一定的信息素,使刚刚被选择的装配操作连接边上的信息素值降低,使之对后面的蚂蚁吸引力变小,这种动态变化避免了算法过早收敛。

在所有蚂蚁完成一次循环后,全部边上的信息素大小先按一定比率衰减,然后对本次循环中蚂蚁走过的所有路径都增加一定的信息素。增加信息素的大小取决于路径评价质量的好坏。本文在装配规划蚁群算法中,只允许一次循环中找到最优序列的蚂蚁在相应的路径上增加全局信息素,使得后续蚂蚁能在本次循环所得到的优解中进行搜索,以起到集中搜索的作用。全局信息素更新规则如下:

$$\tau_{ij}(t+1) = (1-\gamma)\tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (13)$$

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{S}, & \text{路径在第} k \text{个序列中} \\ 0, & \text{否则} \end{cases} \quad (14)$$

其中, $0 < \gamma < 1$ 是信息素全局挥发率, m 是本次循环中最优序列的个数, $\Delta\tau_{ij}^k(t)$ 为增加的信息素大小, Q 是调节参数, S 为序列质量评价函数。序列质量评价函数定义如下:

$$S = w_1 \times D + w_2 \times T + 1 \quad (15)$$

其中, D 是本次循环中得到的最优序列的操作方向改变次数, T 为本次循环中得到的最优序列的装配工具改变次数, w_1 为方向改变次数的权重, w_2 为工具改变次数的权重。

2.4 算法伪代码

1: 系统输入: 装配体的干涉矩阵, 装配工具列表, 蚂蚁数目 m , 装配零件数目 n , 蚁群算法各参数 $\alpha, \beta, \mu, \tau_0, \rho, \gamma, Q$ 。

2: 初始化各参数的值。根据干涉矩阵产生初始可行装配序列节点集。

3: 设置循环次数 $NC=1$;

4: while($NC < NC_{max}$)

5: do{

6: 将 m 只蚂蚁放入初始可行节点位置;

7: while (每只蚂蚁未完成各自的禁忌表)

8: do{

9: 将当前序列置入蚂蚁搜索路径;

10: 通过可行方向公式推导候选序列;

11: 利用概率公式 P 计算每个候选序列的概率;

12: 利用随机轮盘赌选择下一个序列;

13: 将该序列加入蚂蚁的禁忌表;

14: 更新局部信息素;

15: 根据目标函数计算当前最优序列;

16: 更新全局信息素;

17: 更新当前最好序列;

18: 清空候选表、禁忌表;

19: 设置 $NC=NC+1$;

20: 输出最优序列;

3 实例验证与分析

本文采用文献[12]中的第二个实例, 将某控制器装配体作为对象。该装配体为某控制器, 包含 19 个零件, 如图 4 所示。

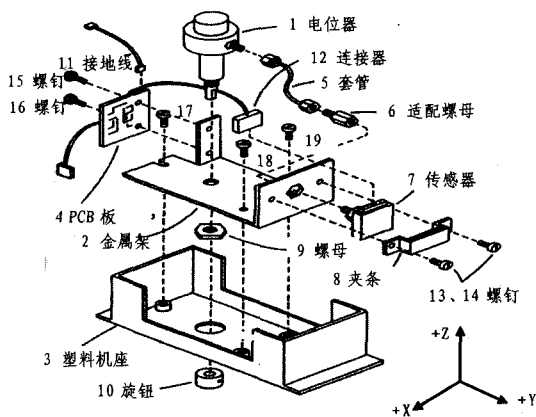


图4 某控制器实例

各零件的装配工具集合如表 2 所列。

表2 零件编号及工具集合

零件编号	工具	零件编号	工具
1	T1 T3	11	T1 T4 T5
2	T1 T3	12	T1 T4 T5
3	T1 T3	13	T2 T4

4	T1 T3	14	T2 T4
5	T1 T3	15	T2 T4
6	T1 T3	T7 16	T2 T4
7	T1 T3	17	T2 T4
8	T1 T3	18	T2 T4
9	T1 T3	T7 19	T2 T4
10	T1 T3 T6		

设定蚁群算法各参数如下: $\alpha=1.0, \beta=0.8, \gamma=0.5, \rho=0.2, \delta=0.2, Q=0.8$, 蚂蚁数量 $M=10, w_1=0.6, w_2=0.4$, 循环次数 $NC=100$ 。根据蚁群算法计算出的序列结果如表 3 所列。

表3 蚁群算法输出的序列优解

蚂蚁	输出的优化装配序列	方向	工具
1 3	(3, -Z, T1)(10-Z, T1)(9, -Z, T1)(2, -Z, T1) (1, -Z, T1)		
4 6	(17, -Z, T2)(18, -Z, T2)(19, -Z, T2)(4, -Z, T1) (8, -Z, T1)	2	4
7 9	(7, -Z, T1)(12, -Z, T1)(5, -Z, T1)(6, -Y, T1) (13, -Y, T4)		
10	(14, -Y, T4)(11, +Y, T4)(15, +Y, T4)(16, +Y, T4) (3, -Z, T1)(10-Z, T1)(9, -Z, T1)(2, -Z, T1) (1, -Z, T1)		
2	(19, -Z, T2)(17, -Z, T2)(18, -Z, T2)(4, -Z, T1) (8, -Z, T1)	2	4
5	(7, -Z, T1)(12, -Z, T1)(5, -Z, T1)(11, +Y, T4) (15, +Y, T4) (16, +Y, T4)(13, -Y, T4)(14, -Y, T4)(6, -Y, T1) (3, -Z, T1)(10-Z, T1)(9, -Z, T1)(2, -Z, T1) (1, -Z, T1)		
8	(17, -Z, T2)(18, -Z, T2)(19, -Z, T2)(4, -Z, T1) (8, -Z, T1) (7, -Z, T1)(12, -Z, T1)(5, -Z, T1)(11, +Y, T4) (15, +Y, T4) (16, +Y, T4)(6, -Y, T1)(13, -Y, T4)(14, -Y, T4)	2	5

算法的收敛曲线如图 5 所示, 综合序列成本最低为 3.8。

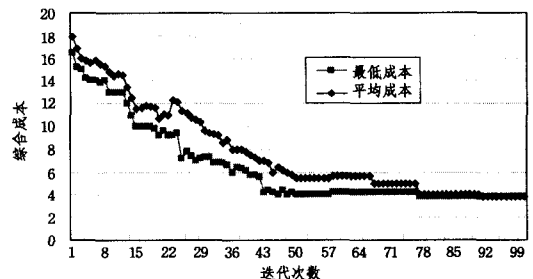


图5 蚁群算法收敛曲线

对蚁群算法得出的序列结果和文献[12]给出的序列结果做比较分析后可以看出, 遗传算法和蚁群算法在求解装配序列规划的问题上都有各自的特点。遗传算法是将一条装配序列作为一个整体来进行交叉变异的, 初始装配序列的优劣直接影响着优化效率。在零件个数和约束较少的情况下, 遗传算法可以在较短的时间内搜索到较优的序列, 但是在零件个数多和约束条件较大的情况下, 这种随机性的交叉变异很有可能产生不可行的序列。

反观蚁群算法, 它是根据蚂蚁在走过的路径上留下的信息素来搜索装配序列, 每条序列中的每一步都是根据信息素搜索出来的局部优解。当走过的路径综合成本越小时, 留下的信息素越多, 就会吸引更多的蚂蚁搜索到这条序列。根据

(下转第 204 页)

参考文献

- [1] Lygeros J. Lecture Notes on Hybrid Systems[R]. Dept. of Electrical and Computer Engineering, University of Patras, 2-6/2/2004
- [2] Pei Hai-long, Krogh B H. Stability Regions for Systems with Mode Transition [A]//Proc. of ACC01[C]. 2001
- [3] Blanchini F. Set Invariance in Control [J]. Automatica, 1999, 35:1747-1767
- [4] Jirstrand M. Invariant Sets for a Class of Hybrid Systems [A]//IEEE, CDC98[C]. 1998
- [5] Li Jian-qiang, Pei Hai-long. Efficient Transition for Invariant Set in a Class of Hybrid Systems [A]//The 6th International Conference on Control and Automation (ICCA2007)[C]. 2007
- [6] Rakovic S V, Grieder P, Kvasnica M, et al. Computation of Invariant Sets for Piecewise Affine Discrete Time Systems subject to Bounded Disturbances[A]//43rd IEEE Conference on Decision and Control[C]. 2004
- [7] Mayne D Q, Rawling J B, Rao C V, et al. Constrained Model Predictive Control: Stability and Optimality [J]. Automatica, 2000, 36:789-814
- [8] Rakovic S V, Kerrigan E C, Kouramas K I, et al. Invariant Approximations of the Minimal Robust Positively Invariant Set [J]. IEEE Transactions on Automatic Control, 2005, 50(3)
- [9] Lin Hai, Antsaklis P J. Robust Invariant Control Synthesis for Discrete-time Polytopic Uncertain Linear Hybrid Systems[A]//Proc. of ACC03[C]. 2003
- [10] Bemporad A, Morari M. Control of systems integrating logic, dynamic, and constraints [J]. Automatica, 1999, 35(3):407-427
- [11] 张聚. 混杂系统理论及在非线形系统中的应用研究[D]. 杭州: 浙江大学, 2005
- [12] Richards A, How J. Mixed-integer Programming for Control
- [13] Bemporad A. Hybrid Toolbox For Real-time Applications. October 2006
- [14] Grieder P, Kvasnica M, Baotic M, et al. Low complexity control of piecewise affine systems with stability guarantee [C] // ACC2004. 2004
- [15] Fletcher R, Leyffer S. Numerical experience with lower bounds for MIQP brand-and-bound [R]. Scotland: Dept. of Mathematics, University of Dundee, 1995
- [15] Torrisi F D, Bemporad A. HYSDEL 2.0-User Manual[M]. 2002
- [16] Wonham W M. Linear Multivariable Control: a Geometric Approach[M]. New York: Springer Verlag, 1985
- [17] 胡一凡. 飞行机器人的建模和控制[D]. 广州: 华南理工大学, 2004
- [18] 席裕庚, 王凡. 非线性系统预测控制的多模型方法[J]. 自动化学报, 1996, 22(4):456-461
- [19] Rakovic S, Grieder P, Kvasnica M, et al. Computation of invariant sets for piecewise affine discrete systems subjects to bounded disturbances[C]//CDC 04. 2004
- [20] Yue Wei, Rodrigues L, Gordon B. Piecewise-affine Control of a Three DOF Helicopter[C]//ACC06. 2006

(上接第 182 页)

此方法, 蚂蚁在可行的路径上探索, 最终能搜索到最优装配序列。这种群体合作方式不仅要走最短的一步, 而且要综合整个蚂蚁群已经走过路径的经验, 选择最优路径, 同时也有一部分蚂蚁在探索新的路径, 从而找到全局最优解。在零件规模较大的情况下, 蚁群算法能更好地搜索出装配最优序列。

结束语 装配序列规划是数字化产品装配规划中的核心技术。本文对装配序列规划问题进行了数学上的描述, 并针对装配序列规划问题的特点, 将蚁群算法应用于求解装配序列规划的问题中。基于有向图的产品装配模型表达产品零件之间的装配优先关系, 集成干涉矩阵和工具列表矩阵表达零件可行装配方向等基本装配信息。本文在分析干涉矩阵的形式、性质的基础上, 采用集成干涉矩阵为算法提供产品可行装配方向推导的信息, 并给出了集成干涉矩阵的十进制和二进制转换方法。在此基础上, 基于蚁群算法的原理和模型, 构建了面向装配序列规划的蚁群算法, 并从装配可行方向的改变次数和装配工具的改变次数建立目标优化函数, 求解装配序列。最后通过两个实例的对比研究, 证明了蚁群算法能较好地搜索出装配最优序列。

参考文献

- [1] 李灿林, 蔡铭, 童若锋, 等. 基于规则和爆炸图的装配序列规划 [J]. 计算机辅助设计与图形学报, 2004, 16(8):1106-1113
- [2] LU Tao, Zhang Bo, JIA Peifa. Assembly Sequence Planning Based on Graph Reduction[C]//Proc. of IEEE Region 10 Int. Conf. on Computers, Communication, Control, and Power Engineering. 1993:119-122

- [3] Caselli S, Zanichelli F. On Assembly Sequence Planning Using Petri Nets [C] // Proceedings of the 1995 IEEE International Symposium on Assembly and Task Planning. 1995:239-244
- [4] Cao Tiejia, Sanderson A C. Task Decomposition and Analysis of Assembly Sequence Plans Using Petri Nets[C]//Proceedings of the Third International Conference on Computer Integrated Manufacturing. 1992:138-147
- [5] Wang Junfeng, Liu Jihong, Zhong Yifang. A Novel Ant Colony Algorithm for Assembly Sequence Planning [J]. Advanced Manufacturing Technology, 2005(25):1137-1143
- [6] Sebaly M F, Fujimoto H. A Genetic Planner for Assembly Automation[C]// Proceedings of the IEEE Conference on Evolutionary Computation. 1996:401-406
- [7] Ye Nong, Banerjee P, Banerjee A, et al. A Comparative Study of Assembly Planning in Traditional and Virtual Environments [J]. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 1999, 29(4):546-555
- [8] 夏平均, 姚英学, 刘江省, 等. 基于虚拟现实和仿生算法的装配序列优化[J]. 机械工程学报, 2007, 43(4):44-52
- [9] 谢龙, 付宜利, 马玉林. 基于复合装配图进行装配序列规划的研究[J]. 计算机集成制造系统, 2004, 10(8):997-1002
- [10] 付宜利, 田立中, 董正卫, 等. 装配关系的有向图表达方法研究 [J]. 计算机集成制造系统, 2003, 9(2):149-153
- [11] Dini G, Santochi M. Automated sequencing and subassembly detection in assembly planning [J]. CIRP Annals-Manufacturing Technology, 1992, 41(1):1-4
- [12] Cao Pengbin, Xiao Renbin. Assembly planning using a novel immune approach [J]. The International Journal of Advanced Manufacturing Technology, 2006(1):770-782