DOI: 10.3901/JME.2019.09.107

基于蚁群算法的异步并行装配序列规划方法*

刘晓阳 刘恩福 靳江艳

(河北科技大学机械工程学院 石家庄 050018)

摘要: 为了提高复杂产品的装配序列规划效率,利用蚁群算法的并行性及其搜索过程的独立性,对装配序列进行异步并行求解。首先,调整产品结构树的装配层次关系,并基于多色集合理论(PST)对装配层次结构进行形式化表达,建立 PS 层次结构树;其次,基于“分治”原则,建立“全局并行、区域线性”的蚂蚁搜索策略,对 PS 层次结构树进行逐层规划和整合计算,实现装配序列的异步并行求解,并提高求解效率;然后,为了解决非线性装配序列表达方式问题,建立装配序列树模型及其自相关矩阵。最后,通过实例验证了异步并行装配序列方法的可行性与有效性。

关键词: 装配序列; 蚁群算法; 异步并行规划; PS 层次结构树; 装配序列树模型

中图分类号: TP391

Asynchronous Parallel Assembly Sequence Planning Based on Ant Colony Algorithm

LIU Xiaoyang LIU Enfu JIN Jiangyan

(College of Mechanical Engineering, Hebei University of Science and Technology, Shijiazhuang 050018)

Abstract: In order to improve the efficiency of complex product assembly sequence planning, using the parallelism of ant colony algorithm and the independence of its search process to solve assembly sequence by parallel asynchronous method. Firstly, the assembly hierarchy of the product structure tree is adjusted, and the PS hierarchy tree is established in order to formally express the hierarchy tree based on the polychromatic set theory. Secondly, the search strategy of ant colony algorithm which is “global parallel, local linear” is established based on the principle of “divide and conquer”, and the PS hierarchy tree is layer by layer planned and integration calculated in order to complete parallel asynchronous solution for assembly sequence and improve solving efficiency. And then, assembly sequence tree model and its self-correlation matrix are established to meet the requirement of nonlinear assembly sequence expression. Finally, the feasibility and effectiveness of asynchronous parallel assembly sequence planning is verified by a typical case.

Key words: assembly sequence; ant colony algorithm; asynchronous parallel planning; PS hierarchy tree; assembly sequence tree model

0 前言

装配序列规划(Assembly sequence planning, ASP)是进行装配的前提和基础,其对于确保装配可行性、降低装配成本、缩短产品开发周期具有非常重要的意义。目前,装配序列规划的研究主要集中在串行规划,即装配任务以线性方式依次地执行,如装配优先约束法^[1]、装配割集法^[2]等经典装配序列规划方法以及遗传算法^[3]、粒子群算法^[4]、蚁群算法^[5]等现代启发式装配序列规划方法,上述优化方法在处理线

性装配领域方面都取得了良好的进展,但是都只考虑了线性装配而没有把问题扩展到并行装配领域,不能生成具有层次结构的解序列,无法反映产品协同设计和网络化制造的复杂产品装配规划的特点。

近年来有一些学者提出了一些变传统线性装配为并行装配的方法,以提高装配效率,如:文献[6]对遗传编程算法中有关算子及参数进行了设计与改进,使算法能够处理并行装配序列规划问题,并且装配效率和装配时间要明显优于线性装配方式;文献[7]提出了基于模糊粗糙集的并行装配序列规划方法,将待安装零件作为知识论域,适切隶属度函数作为评价指标的不可辨识关系,并基于知识论域和不可辨识关系生成装配并行序列;文献[8]提出了基于约束解除和简化配合关联图的装配序列并

* 国家自然科学基金(51275234)、河北省应用基础研究计划重点基础研究(14961811D)和河北省高等学校科学技术研究资助项目(QN2017066)。20180309 收到初稿,20180923 收到修改稿

行优化方法,根据联接度值的大小去除零件之间的虚联接图,从而实现串行装配的并行优化;文献[9]提出并行拆卸序列规划问题的染色体编码方案和遗传进化算子,实现了复杂产品并行拆卸序列规划的高效求解。

作者在实际调研中发现,复杂产品需要根据装配工艺要求进行逐层划分生成各层装配体,各层装配体由装配子任务在不同企业或者不同工位上完成装配;装配过程中装配子任务间具有一定的独立性、并行性,并且彼此之间大多不同步,体现复杂产品装配过程的“并行、异步”的特点。

针对这一特点,本文首次采用蚁群算法对装配序列进行异步并行规划:根据装配工艺要求建立装配序列规划(ASP)子任务,利用蚁群算法与生俱来的并行性和搜索过程的独立性^[10],对各个 ASP 子任务以“全局并行,区域线性”的策略进行搜索,将传统的线性序列求解方式转化成多个组合的异步并行求解方式;规划过程采用“分治”原则,有效缩减了参与规划的零部件数量,避免了因零件总数增大而导致的装配序列解空间组合爆炸,提高了规划效率,满足协同设计的需要,同时,生成的装配序列也与产品的“并行、异步”的装配过程相吻合。

1 PS 层次结构树

在一些数字化产品中,产品的层次结构树为

图1 PS 层次结构树

在 PS 层次结构树中,以零部件为核心进行独立装配的组合体称为装配单元,装配单元包括零件、部件以及满足强制执行(PCGE)机制^[5]的并行零件组等;装配单元用 $A(k, i_k, j_{k-1})$ 表示, (k, i_k, j_{k-1})

单一层次或不符合面向装配的设计,需要将面向设计的装配结构转化为面向装配的装配结构^[11],即在不改变物理的空间位置关系前提下,调整装配结构树的装配层次关系,得到更规则的层次化装配结构。

1.1 多色集合理论

多色集合理论(Polychromatic sets theory, PST)^[12-13]是近年来先进制造技术领域中的重要概念和理论之一,是俄罗斯 V. V. Pavlov 教授于 2002 年提出的新的系统理论工具,其在系统建模、复杂层次结构和关系描述、问题的形式化表示以及高效的计算机处理等方面的优势,愈来愈得到产品设计与制造领域相关研究人员的重视。

多色集合(Polychromatic sets, PS)是在普通集合的基础上,将集合本身和组成元素同时涂上一些不同的“颜色”,用来表示研究对象和它的性质。在使用多色集合时,通常用围道的概念来替换术语颜色,围道即为性质、属性、参数、特征、指标等技术概念的抽象和概括^[12-13]。

1.2 PS 层次结构树的建立

基于产品层次结构树的最初层次关系,利用分层分级的思想^[14]重新调整其装配层次关系,将新的层次级别关系依次规划某一个父节点下一层的子节点,规划后的节点以及节点之间的逻辑层次关系用多色集合(Polychromatic sets, PS)进行形式化表达,建立 PS 层次结构树,如图 1 所示。

为装配单元在 PS 层次结构树的编码,表示第 k 层、第 i_k 装配单元,其父装配单元为 $k-1$ 层的第 j_{k-1} 装配单元,装配单元间的递归表示形式如式(1)所示

$$\left\{ \begin{array}{l} A(0,0,0) = \bigcup_{i_1=1}^{n_1} A(1,i_1,0) \\ A(k,i_k,j_{k-1}) = \bigcup_{i_{k+1}=m_0(k)+1}^{m(k)} A(k+1,i_{k+1},i_k) \quad \forall k \in [1,n] \\ m_0(k) = \sum_{i=1}^{i_k-1} n(k+1,i) \\ m(k) = m_0(k) + n(k+1,i_k) \\ n_k = \sum_{i_{k-1}=1}^{n_{k-1}} n(k,i_{k-1}) \quad n_0=1, \forall i_k \in [1,n_k] \end{array} \right. \quad (1)$$

式中 $A(0,0,0)$ ——根节点；

$m_0(k)$ ——第 k 层第 $1 \sim i_k-1$ 装配单元所包含的子装配单元数目之和；

$m(k)$ ——第 k 层第 $1 \sim i_k$ 装配单元所包含的子装配单元数目之和；

$n(k+1,i_k)$ ——第 k 层、第 i_k 个装配单元的子装配单元数目；

n_k ——第 k 层的所有装配单元数目。

定义 1：装配域，是通过分层分级思想^[14]划分的且在同一个 ASP 子任务完成装配的装配单元集合；装配域规定了区域序列规划的范围，用 $R(k,ir_k)$ 表示，其中， (k,ir_k) 为装配域在 PS 层次结构树中的编码， ir_k 表示第 k 层、第 ir_k 装配域，装配域间的递归表示形式如式(2)所示

$$\left\{ \begin{array}{l} R(k,ir_k) = \bigcup_{ir_{k+1}=mr_0(k)+1}^{mr(k)} R(k+1,ir_{k+1}) \\ mr_0(k) = \sum_{ir=1}^{ir_k-1} nr(k+1,ir) \\ mr(k) = mr_0(k) + nr(k+1,ir_k) \\ nr_k = \sum_{ir_{k-1}=1}^{nr_{k-1}} nr(k,ir_{k-1}) \quad nr_1=1, \forall ir_k \in [1,nr_k] \end{array} \right. \quad (2)$$

式中 $nr(k+1,ir_k)$ ——第 k 层、第 ir_k 个装配域的子装配域数目；

$mr_0(k)$ ——第 k 层第 $1 \sim ir_k-1$ 装配域所包含的子装配域数目之和；

$mr(k)$ ——第 k 层第 $1 \sim ir_k$ 装配域所包含的子装配域数目之和；

nr_k ——第 k 层的所有装配域数目。

根据 PS 层次结构树中装配单元与装配域间的逻辑关系，建立二者之间的递归表示形式如式(3)所示

$$\left\{ \begin{array}{l} \forall A(k,i,ir_k) \in R(k,ir_k) \\ m_0(k,ir_k-1) = \sum_{i=1}^{ir_k-1} n(k-1,i) \\ m(k,ir_k) = m_0(k,ir_k-1) + n(k-1,ir_k) \\ m_0(k,ir_k-1) < i \leq m(k,ir_k) \end{array} \right. \quad (3)$$

式中， $m_0(k,ir_k-1)$ 表示 k 层第 $1 \sim ir_k-1$ 装配域所包含的装配单元数目之和，即 $k \square 1$ 层第 $1 \sim ir_k-1$ 装配单元所包含的子装配单元数目之和； $m(k,ir_k)$ 表示 k 层第 $1 \sim ir_k$ 装配域所包含的装配单元数目之和。

1.3 PS 层次结构树的生成算法

根据分层分级思想，将装配单元中满足特定共同特征的零部件划分为同一个具有层次级别关系的特殊装配结构体节点，而高级别的装配单元父节点又可以继续划分为多个低级别的装配单元子节点通过这种父子节点层层嵌套与划分，得到一个基于装配单元新的装配层次结构，生成算法如图 2 所示。

(1) 从数据库导入装配信息数据集，生成产品的 PPR 结构树，总层数为 n 。

(2) 将 Product 节点定义为 $A(0,0,0)$ ，初始化参数： $k=1, i_1=1, ir_1=1, j_0=0$ 。

(3) 遍历 k 层节点，根据装配单元与装配域间的递归表示形式(式 3)，将满足特定共同特征的节点建立装配域 $R(k,ir_k)$ 。

(4) 遍历 $R(k,ir_k)$ 中节点的节点类型，如果包含子节点，则将该节点及其包含的所有子节点规划为节点 $A(k,i_k,j_{k-1})$ ， $i_k=i_k+1$ ；如果不包含子节点，则将该节点规划为节点 $A(k,i_k,j_{k-1})$ ， $i_k=i_k+1$ 。

图 2 PS 层次结构树生成流程图

(5) 遍历 $A(k, i_k, j_{k-1})$ 的层次路径, 计算其新的相对位姿矩阵, 并更新数据库中 $A(k, i_k, j_{k-1})$ 新的位姿矩阵信息。

(6) 重复步骤(4)~步骤(5), 完成装配域 $R(k, ir_k)$ 规划, $ir_k = ir_k + 1$ 。

(7) 重复步骤(6), 完成该层节点规划。

(8) $k = k + 1$, 如果 $k < n$, 重复步骤(4)~步骤(8)。

(9) 得到产品的 PS 层次结构树, 并以式(1)和式(2)的递归表示形式保存数据。

(10) 结束。

2 PS 装配关系模型

PS 装配关系模型是基于多色集合(PS)描述装配域中装配单元之间的装配逻辑关系(包括干涉关系和连接-配合关系)的自相关矩阵 $\mathbf{PS}_{A \times A}$, 由干涉矩阵(Interference matrix, IM)组成和连接-配合关系矩阵(Connection-fit matrix, CM)组成, 如式(4)所示

$$\mathbf{PS}_{A \times A} = \{\mathbf{IM}(k, ir_k), \mathbf{CM}(k, ir_k)\} \quad (4)$$

(1) 干涉矩阵 $\mathbf{IM}(k, ir_k)$: 描述装配域 $R(k, ir_k)$ 中装配单元之间在 $\pm x$, $\pm y$, $\pm z$ 方向的运动可能性, 使序列具有几何可行性, 如式(5)所示

$$\begin{cases} \mathbf{IM}(k, ir_k) = [(A \times A) \times F(A)] = \|e_{ij}^{\mathbf{IM}}(k, ir_k)\|_{(A \times A), F(A)} \\ F_s(A(k, i_1, ir_k), A(k, i_2, ir_k)) \in F(A) \\ A(k, i_1, ir_k), A(k, i_2, ir_k) \in R(k, ir_k) \end{cases} \quad (5)$$

$$e_{ij}^{\mathbf{IM}}(k, ir_k) = \begin{cases} 1 & \text{在给定方向存在干涉} \\ 0 & \text{反之} \end{cases}$$

(2) 关系矩阵 $\mathbf{CM}(k, ir_k)$: 描述装配域 $R(k, ir_k)$ 中装配单元之间在 x , y , z 方向的连接-配合关系(螺纹连接、过盈配合等), 使序列具有一定的连贯性和稳定性, 如式(6)所示

$$\begin{cases} \mathbf{CM}(k, ir_k) = [(A \times A) \times F(A)] = \|e_{ij}^{\mathbf{CM}}(k, ir_k)\|_{(A \times A), F(A)} \\ F_s(A(k, i_1, ir_k), A(k, i_2, ir_k)) \in F(A) \\ A(k, i_1, ir_k), A(k, i_2, ir_k) \in R(k, ir_k) \end{cases} \quad (6)$$

$$e_{ij}^{\mathbf{CM}}(k, ir_k) = \begin{cases} 1 & \text{在给定方向存在连接或配合关系} \\ 0 & \text{反之} \end{cases}$$

将同一装配域的装配单元的两两组合定义为元素, 装配单元间的装配逻辑关系定义为多色集合的围道 $F(A)$

$$F(A) = \bigcup_{s=1}^n F_s(A(k, i_1, j), A(k, i_2, j)) \quad (7)$$

式中, $F_s((A(k, i_1, j), A(k, i_2, j)))$ 表示装配单元 $A(k, i_1, j)$ 和 $A(k, i_2, j)$ 之间的第 s 属性, 简称为 F_s 。装配关系及对应符号如表 1 所示, 其中, $F_1 \sim F_{12}$ 表示装配单元间的装配关系。

表 1 装配关系及对应符号

干涉关系						联接和配合关系					
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
+x 向	+y 向	+z 向	-x 向	-y 向	-z 向	x 向	y 向	z 向	x 向	y 向	z 向
平动	平动	平动	平动	平动	平动	平动	平动	平动	转动	转动	转动

传统装配关系矩阵主要描述的是零件间的装配关系, 而由多个零件组成的装配单元(部件或并行零件组)的装配关系矩阵生成的研究并不多见。

本文以零件的装配关系矩阵为基础, 通过多色集合的析取($P \sqcup S$)运算^[12], 逐层推理, 生成部件和并行零件组的装配关系矩阵。

(1) 多色集合的析取运算

多色集合的析取($P \sqcup S$)运算的表达形式如下: 当

$$F_j(A) = \bigvee_{p=1}^m F_j(a_p) = 1, \quad F_j(A) = 1。$$

(2) 矩阵的逻辑推理

通过多色集合的析取运算以及零件间的装配关系矩阵, 由式(8)推理出装配域中非零件装配单元间的装配关系

$$F_s(A(k, i_1, j), A(k, i_2, j)) = \bigvee_{i_1=1}^{m(k+1, i_1)} \left(\bigvee_{i_2=1}^{m(k+1, i_2)} F_s(A(k+1, ii_1, i_1), A(k+1, ii_2, i_2)) \right) \quad (8)$$

式中, $s=1, 2, \dots, 12$; $m(k+1, i_1)$ 表示装配域 $R(k+1, i_1)$ 中包含的装配单元数目, $A(k+1, ii_1, i_1) \in R(k+1, i_1)$; $m(k+1, i_2)$ 表示装配域 $R(k+1, i_2)$ 中包含的装配单元数目, $A(k+1, ii_2, i_2) \in R(k+1, i_2)$ 。

以图 3 的某装配体为例, 其 PS 层次结构树中包含 8 个装配单元分别存在于装配域 $R(1, 1)$, $R(2, 1)$ 和 $R(2, 2)$ 中, 其中, 零件间的装配关系矩阵如图 4(a) 所示; 根据图 4a 中零件间的装配关系获得 $R(1, 1)$ 、 $R(2, 1)$ 和 $R(2, 2)$ 的装配关系矩阵, 如图 4b、4c 和 4d 所示; 其中, $R(2, 1)$ 和 $R(2, 2)$ 为底层装配域, 包含的装配单元是零件, 其装配关系矩阵是从图 4a 直接提取, 而 $R(1, 1)$ 的装配关系矩阵是通过式(8)推理得到。

图 3 某装配体的 PS 层次结构树

图 4 某装配体的 PS 装配关系矩阵

3 异步并行序列规划

定义 2: 异步并行装配序列规划(Asynchronous parallel assembly sequence planning, Ap-aspp)按照装配单元的层次级别关系, 将整体的 ASP 任务分解为若干个及若干层级、彼此独立的 ASP 子任务, 利用蚁群算法的并行性和搜索独立性, 将蚁群算法的搜索策略、搜索函数、禁忌表以及蚂蚁分配等问题进行设计和改进, 对 PS 层次结构树进行逐层规划和整合计算, 实现将传统的线性序列求解方式转化成多个组合的异步并行求解。

3.1 搜索策略

为了实现蚁群算法的异步并行序列规划, 本文采取“全局并行、区域线性”的搜索策略, 如图 5 所示。

(1) 全局并行策略, 是以装配域作为规划对象, 确定一次迭代中装配域参与区域搜索的进度, 是 Ap-aspp 算法的主线程。在 PS 层次结构树中, 由于装配单元的层次隐含着一定的装配先后关系, 即先装配下层装配单元后装配上层装配单元, 因此, 在全局搜索中, 包含下层装配单元的装配域要先于包含上层装配单元的装配域参与区域搜索; 如果包含的装配单元是同层且没有先后约束关系的装配域, 则并行参与区域搜索。

(2) 区域线性策略, 是以装配域作为搜索范围,

以装配单元作为规划对象, 是 Ap-aspp 算法的子线程。在迭代过程中, 主线程实时的将满足规划要求的一个或多个(并行)装配域分配给子线程, 每个子线程根据装配域粒度大小匹配蚂蚁数量, 并采用独立的信息素更新机制对进行区域线性求解。

图 5 一次迭代中装配域的搜索进度

3.2 搜索函数

3.2.1 全局搜索函数

在 Ap-aspp 算法中, 全局规划是从 PS 层次结构树的底层进行遍历, 装配域能否满足参与区域线性规划, 主要决于其装配基础件是否满足搜索条件。

装配基础件是由工艺人员根据装配工艺要求和

实际装夹条件进行确定,作为装配域中首先安装的装配单元,也是蚂蚁搜索的起始节点;如果装配域中装配基础件满足搜索条件,则蚂蚁将装配基础件作为起始节点,对装配域进行搜索了。

装配域的搜索条件:一次迭代中,如果装配域的装配基础件无子装配单元,或者装配基础件有子装配单元,且所有子装配单元在其他装配域中都被蚂蚁走过,判断公式如式(9)所示,则该装配域可以参与区域线性规划。

$$\begin{cases} f(R(k,ir_k)) = \bigvee_{i_{k+1}=m_0(k)+1}^{m(k)} f(A(k+1,i_{k+1},i_k)) \\ A(k,i_k,j_{k-1}) \in R(k,ir_k) \\ A(k,i_k,j_{k-1}) = \bigcup_{i_{k+1}=m_0(k)+1}^{m(k)} A(k+1,i_{k+1},i_k) \end{cases} \quad (9)$$

式中, $A(k+1,i_{k+1},i_k)$ 是 $A(k,i_k,j_{k-1})$ 的子装配单元, $A(k,i_k,j_{k-1})$ 是 $R(k,ir_k)$ 的装配基础件; $f(A(k+1,i_{k+1},i_k))$ 是 $A(k+1,i_{k+1},i_k)$ 的赋值参数,初始化为“1”,若 $A(k+1,i_{k+1},i_k)$ 被蚂蚁走过,则 $f(A(k+1,i_{k+1},i_k))$ 为“0”;如果 $A(k,i_k,j_{k-1})$ 中所有子装配单元都被蚂蚁走过,则 $f(R(k,ir_k))=0$, 否则, $f(R(k,ir_k))=1$ 。

3.2.2 区域搜索函数

在 Ap-asp 算法的区域规划中,参与搜索的蚂蚁有特定的搜索区域 $R(k,ir_k)$ 、独立的 $CM(k,ir_k)$ 以及实时分析几何可行性的 $IM(k,ir_k)$ 等信息,因此,定义 t 时刻蚂蚁由节点 i 转移到目标节点 j 的概率为

$$p_{ij}^{(k,ir_k)}(t) = \begin{cases} \frac{[\tau_{ij}^{(k,ir_k)}(t)]^\alpha [\eta_{ij}^{(k,ir_k)}(t)]^\beta}{\sum_{r \in allow_s(k,ir_k)} [\tau_{ir}^{(k,ir_k)}(t)]^\alpha [\eta_{ir}^{(k,ir_k)}(t)]^\beta} allow_s(k,ir_k) & i, j \in \\ 0 & \text{其他} \end{cases} \quad (10)$$

式中, α , β 分别为信息素和启发因子的重要程度参数;候选集 $allow_s(k,ir_k)$ 表示 $R(k,ir_k)$ 中第 s 只蚂蚁下一步允许走过的装配单元的集合; $\tau_{ij}^{(k,ir_k)}(t)$ 表示 $R(k,ir_k)$ 中蚂蚁由节点 i 转移到目标节点 j 的信息浓度; $\eta_{ij}^{(k,ir_k)}(t)$ 表示 $R(k,ir_k)$ 中蚂蚁由节点 i 转移到目标节点 j 的启发信息。

(1) 候选集 $allow_s(k,ir_k)$: 为了保障每一个装配序列是可行的, Ap-asp 算法在每一次迭代中,依据干涉矩阵 $IM(k,ir_k)$ 中提供的装配干涉信息,以装配域 $R(k,ir_k)$ 为单位对装配单元进行干涉检查,将具有可装配性的装配单元组成了候选集 $allow_s(k,ir_k)$, 保证蚂蚁走过的每个装配单元具有可装配性。

1) 装配单元装配后不干涉后续装配单元的装配,即装配单元具有装配几何可行性:通过干涉矩阵 $IM(k,ir_k)$ 中提供的装配单元间装配干涉信息,多

色集合的析取($P \sqcup S$)运算和合取($P \sqcap S$)运算^[12]建立装配干涉方程,对装配域中未搜索的装配单元进一步判断。

多色集合的析取($P \sqcup S$)运算的表达形式如下:当

$$F_j(A) = \bigwedge_{p=1}^m F_j(a_{ip}) = 1, \quad F_j(A) = 1。$$

定义 $R_s(k,ir_k)$ 为装配域 $R(k,ir_k)$ 中,当第 s 只蚂蚁位于 i 节点(即 $A(k,i,ir_k)$)时未走过的装配单元集合, nr_s 为 $R_s(k,ir_k)$ 中的装配单元数, $A(k,p,ir_k)$, $A(k,q,ir_k)$ 分别为 $R_s(k,ir_k)$ 的第 p , q 个装配单元, $p \neq q$ 且 $p, q \leq nr_s$, 建立装配干涉方程为

$$W_s(k,p,ir_k) = \bigwedge_{p=1}^{nr_s-1} F_j(A(k,p,ir_k), R_s(k,ir_k)) = \bigwedge_{p=1}^{nr_s} F_j(A(k,p,ir_k), \bigvee_{q=1}^{nr_s-1} [A(k,q,ir_k)]_s) \quad (11)$$

式中, $F_j(j=1,2,\dots,6)$ 表示装配单元之间在 $\pm x$, $\pm y$, $\pm z$ 方向的干涉关系属性,若 $W_s(k,p,ir_k)=0$, 则 $R_s(k,ir_k)$ 的第 p 个装配单元不影响后续装配单元的装配,否则,会影响。因此,候选集 $allow_s(k,ir_k)$ 的生成过程可简单表示如下

$$allow_s(k,ir_k) \leftarrow \frac{W_s(k,p,ir_k)=0}{W_s(k,p,ir_k)=0} R_s(k,ir_k) \quad p=1,2,\dots,nr_s$$

从 $R_s(k,ir_k)$ 中选取 $W_s(k,p,ir_k)=0$ 的装配单元构建第 s 只蚂蚁位于 i 节点的候选集 $allow_s(k,ir_k)$ 。

2) 装配单元要能够参与装配:若装配域中有装配单元为非底层装配单元,即该装配单元含有子装配单元,如果其子装配单元未参与装配,则该装配单元是不存在的。

在一次迭代中,定义装配域 $R(k,ir_k)$ 中蚂蚁下一步要走的节点是 $A(k,p,ir_k)$, 即 $A(k,p,ir_k) \in allow_s(k,ir_k)$, 假设装配单元含有子装配单元尚未被蚂蚁没走过,则该装配单元不能参与装配,需待其他装配域的规划结果,直到 $A(k,p,ir_k)$ 所有子装配单元被蚂蚁走过,则装配域 $R(k,ir_k)$ 才能进一步规划,这 and 实际装配过程是一样的,也说明在区域规划过程中,各子线程不是完全独立,彼此之间存在一定的信息传递和约束关系。

(2) 信息素更新机制:针对各个线程以装配域作为搜索区域的并行异步搜索特点,区域规划过程中需要建立区域独立的信息素更新机制,即

$$\begin{cases} \tau_{ij}^{(k,ir_k)}(t+1) = (1-\rho)\tau_{ij}^{(k,ir_k)}(t) + \rho\Delta\tau_{ij}^{(k,ir_k)}(t) \\ \Delta\tau_{ij}^{(k,ir_k)}(t) = \sum_{s=1}^m \Delta\tau_{ij}^{(k,ir_k,s)}(t) \end{cases} \quad (12)$$

式中, ρ 为信息素挥发系数,则 $1-\rho$ 表示信息残留因子, $\rho \in [0,1]$; 本次循环中, $\Delta\tau_{ij}^{(k,ir_k)}(t)$ 表示 $R(k,ir_k)$ 中由节点 i 转移到目标节点 j 的信息素增量,

$\Delta\tau_{ij}^{[(k,ir_k),s]}(t)$ 表示 $R(k,ir_k)$ 的第 s 只蚂蚁由节点 i 转移到目标节点 j 留下的信息量, 并采用精英蚁群策略更新信息素, 即

$$\Delta\tau_{ij}^{[(k,ir_k),s]}(t) = \begin{cases} \frac{Q}{L_s} & (i, j) \in Ant_best_Path \\ 0 & \text{其他} \end{cases} \quad (13)$$

式中, 精英蚂蚁即为第 s 只蚂蚁, Ant_best_Path 为精英蚂蚁路径; Q 为常量; L_s 为第 s 只蚂蚁本次循环中所走路径总长度。

3.3 蚂蚁数

蚂蚁数的选择主要在搜索能力和收敛速度两方面作合理考虑, 先考虑算法的稳定性和所求得最优解的全局性, 其次才是收敛速度。当蚂蚁数量 $m=n$ (问题的规模) 时, 算法可以在较少的迭代次数内找到一个最佳的最优解, 尽管其收敛速度不是最快, 但算法的整体性能最佳^[15]。

在 Ap-asp 算法中, 蚂蚁数包括蚂蚁总数 mc 和区域匹配蚂蚁数 $mc(k,ir_k)$, 其中, mc 根据装配单元总数计算, 基本装配单元总数保持一致; $mc(k,ir_k)$ 则是根据装配域粒度的大小计算, 其对应的是装配域 $R(k,ir_k)$ 匹配的蚂蚁数, 蚂蚁数的计算公式, 如式 (14) 所示

$$\begin{cases} mc = \sum_{k=1}^n n_k \\ mc(k,ir) = n(k,ir) \end{cases} \quad (14)$$

3.4 禁忌表

根据 Ap-asp 算法的搜索策略, 本文对传统蚁群算法的禁忌表进行调整, 定义禁忌表包括全局禁忌表 $tabu$ 和区域禁忌表 $tabu(k,ir_k)$, 其中, 全局禁忌表 $tabu$ 记录每次迭代过程中完成搜索的装配域; 区域禁忌表 $tabu(k,ir_k)$ 记录在装配域 $R(k,ir_k)$ 中蚂蚁当前所走过的装配单元。

在算法执行过程中, 禁忌表随着进化过程作动态调整。其中, 区域禁忌表 $tabu(k,ir_k)$ 实时记录装配域 $R(k,ir_k)$ 中蚂蚁已经走过的装配单元; 而全局禁忌表 $tabu$ 的组成是根据区域禁忌表 $tabu(k,ir_k)$ 以及装配单元与装配域间的递归表示形式 (式 3) 进行推理得到。

3.5 多目标启发函数

多目标启发函数 $\eta_{ij}(k,ir_k)$ 是以为装配域单位, 综合考虑其组成装配单元的装配方向、装配工具以及装配稳定性等指标并对其进行量化, 通过设定权重将各量化指标融合到一起建立的; 装配域 $R(k,ir_k)$ 的多目标启发函数用 $\eta_{ij}(k,ir_k)$ 表示, 其数学表达式如式 (15) 所示

$$\eta_{ij}(k,ir_k) = w_1 d_{ij}(k,ir_k) + w_2 c_{ij}(k,ir_k) + w_3 p_{ij}(k,ir_k) + w_4 s_{ij}(k,ir_k) \quad (15)$$

式中, $w_1 \sim w_4$ 为设定目标权重, 且 $\sum_{i=1}^4 w_i = 1$; $d_{ij}(k,ir_k)$

为方向性指标, $c_{ij}(k,ir_k)$ 为连续性指标, $p_{ij}(k,ir_k)$ 为并行性指标, $s_{ij}(k,ir_k)$ 为稳定性指标, 权重指标要根据装配体的结构特点, 估计不同优化目标的相对重要程度并分配不同的权值。

(1) 方向性指标: 为了减少装配过程中重定向次数, 尽量保持装配方向的一致性, 指标 $d_{ij}(k,ir_k)$ 函数为

$$d_{ij}(k,ir_k) = \begin{cases} 1.0 & d_i(k,ir_k) = d_j(k,ir_k) \\ 0.5 & d_i(k,ir_k) = -d_j(k,ir_k) \\ 0.1 & d_i(k,ir_k) \neq \pm d_j(k,ir_k) \end{cases} \quad (16)$$

式中, $d_i(k,ir_k)$ 是装配域 $R(k,ir_k)$ 中第 i 个装配单元的装配方向。

(2) 连续性指标: 表达连续装配的两个装配单元之间具有配合、连接的物理关系, 提高装配过程的集中性, 指标 $c_{ij}(k,ir_k)$ 函数为

$$c_{ij}(k,ir_k) = \begin{cases} 1.0 & e_{ij}^{CM}(k,ir_k) = 1 \\ 0.1 & e_{ij}^{CM}(k,ir_k) = 0 \end{cases} \quad (17)$$

式中, $e_{ij}^{CM}(k,ir_k)$ 是装配域 $R(k,ir_k)$ 的接触-连接矩阵 $CM(k,ir_k)$ 中装配单元 i 和装配单元 j 的连接-配合关系。

(3) 并行性指标: 指装配过程中相同或相似的操作集中完成的程度, 减少装配工具改变次数越少, 根据装配单元类型或者相同重量 (或体积)^[5] 建立指标 $p_{ij}(k,ir_k)$ 函数为

$$p_{ij}(k,ir_k) = \begin{cases} 1.0 & W_i(k,ir_k) = W_j(k,ir_k) \\ & W_i(k,ir_k) \neq W_j(k,ir_k) \& \\ & T_i(k,ir_k) = T_j(k,ir_k) \\ 0.5 & \\ & W_i(k,ir_k) \neq W_j(k,ir_k) \& \\ & T_i(k,ir_k) \neq T_j(k,ir_k) \\ 0.1 & \end{cases} \quad (18)$$

式中, $W_i(k,ir_k)$ 是装配域 $R(k,ir_k)$ 的第 i 个装配单元的重量; $T_i(k,ir_k)$ 是装配域 $R(k,ir_k)$ 的第 i 个装配单元的类型。

(4) 稳定性指标: 保证装配单元的定位可靠, 即装配过程是一个循序渐进的过程, 倾向于先装配具有较多连接关系的零部件以保持其稳定性^[5], 建立指标 $s_{ij}(k,ir_k)$ 函数为

$$s_{ij}(k,ir_k) =$$

$$\begin{cases} 1.0 & \sum_{j=0}^n \text{IMg}_j(k, ir_k) = 0 \\ \frac{Dcc_{\max}(k, ir_k) - Dcc_j(k, ir_k)}{Dcc_{\max}(k, ir_k)} & \text{其他} \end{cases} \quad (19)$$

式中, $\text{IMg}_j(k, ir_k)$ 是装配域 $R(k, ir_k)$ 的重力方向(默认是 z 方向)的干涉矩阵; $Dcc_{\max}(k, ir_k)$ 是接触-连接矩阵 $\text{CM}(k, ir_k)$ 中某装配单元的最大接触连接度^[16], $Dcc_j(k, ir_k)$ 是装配域 $R(k, ir_k)$ 中当前装配单元的接触连接度。

4 装配序列的表达

装配序列的表达直接影响到装配序列生成与优化以及后续的装配工艺规划的效率, 根据对目前装配序列表达方式及其研究现状的分析, 结合 Ap-asp 算法对非线性装配序列表达方式的需求, 本文建立装配序列树模型及其自相关矩阵 $[(R \times R) \times F(R)]$, 用以表达非线性装配序列。

4.1 装配序列树模型

装配序列树模型中包含两类节点: 装配数据节点和装配关系节点。其中, 装配数据节点为完成规划的装配域节点(R 节点), 由装配单元的线性序列组成; 装配关系节点描述装配数据节点之间常见的装配关系, 即 $S(R(k, i), R(k, j))$ 节点(顺序节点)、

$P(R(k, i), R(k, j))$ 节点(并行节点)以及 $AP(R(k, i), R(k, j))$ 节点(异步并行节点)三种类型。

(1) $S(R(k, i), R(k, j))$ 节点: 在构成装配序列时, $S(R(k, i), R(k, j))$ 节点下的装配域 $R(k, i)$ 和 $R(k, j)$ 必须以自左往右的顺序排列, 即前置装配域 $R(k, i)$ 完成装配, 后置装配域 $R(k, j)$ 才能装配。

(2) $P(R(k, i), R(k, j))$ 节点: 在构成装配序列时, $P(R(k, i), R(k, j))$ 节点下的装配域 $R(k, i)$ 和 $R(k, j)$ 平行排列, 即装配域 $R(k, i)$ 和 $R(k, j)$ 之间能够同时独立执行, 互不影响。

(3) $AP(R(k, i), R(k, j))$ 节点: 在构成装配序列时, $AP(R(k, i), R(k, j))$ 节点下装配域 $R(k, i)$ 和 $R(k, j)$ 异步并行执行, 即后置装配域 $R(k, j)$ 需要前置装配域 $R(k, i)$ 的输出信息, 后置装配域 $R(k, j)$ 才能完成装配。

4.2 树模型的形式化表达

装配序列树模型对非线性装配序列的表达简洁直观, 但不方便计算机进行信息处理和存储, 因此, 通过多色集合理论建立自相关矩阵 $[(R \times R) \times F(R)]$ 对树模型进行形式化表达; 在矩阵中, 定义多色集合的元素为 R 节点的两两组合, 定义多色集合的围道 $F_k(R(k, i), R(k, j))$ 为 R 节点之间可能的装配关系类型, 简记 F_k , 即顺序、并行、异步, $F_k = (F_1, F_2, F_3)$, 数据节点装配关系的形式化映射如表 2 所示, 三种关系节点的运算规则如下所述。

表 2 装配域节点的装配关系及其形式化映射

关系节点	关系模型	PS关系矩阵	过程代数表示
$S(R(k, i), R(k, j))$		$(R(k, i), R(k, j)) \begin{matrix} F_1 & F_2 & F_3 \\ 1 & 0 & 0 \end{matrix}$	$S(R(k, i), R(k, j)) = (R(k, i) // R(k, j))$, $R(k, i)$ 为 $R(k, j)$ 的顺序前置的节点, $R(k, j)$ 为 R_i 的顺序 $R(k, i)$ 后置节点。
$P(R(k, i), R(k, j))$		$(R(k, i), R(k, j)) \begin{matrix} F_1 & F_2 & F_3 \\ 0 & 1 & 0 \\ (R(k, j), R(k, i)) & 0 & 1 & 0 \end{matrix}$	$P(R(k, i), R(k, j)) = (R(k, i) \otimes R(k, j))$, $R(k, i)$ 、 $R(k, j)$ 均为并行结构的分支。
$AP(R(k, i), R(k, j))$		$(R(k, i), R(k, j)) \begin{matrix} F_1 & F_2 & F_3 \\ 0 & 0 & 1 \end{matrix}$	$AP(R(k, i), R(k, j)) = (R(k, i) R(k, j))$, $R(k, i)$ 、 $R(k, j)$ 为异步并行结构的分支, $R(k, i)$ 为 $R(k, j)$ 的前置节点, 即 $R(k, j)$ 需要 $R(k, i)$ 的输出信息作为输入。

5 Ap-asp 算法步骤及流程

(1) 生成 PS 层次结构树。计算结构树每层的节点数为 n_k 和装配域数为 nr_k , $k=1, 2, 3, \dots, n$, 其中, n 为总层数。

(2) 蚂蚁信息初始化。计算蚂蚁的数量 mc , 并根据装配域粒度大小匹配蚂蚁数量 $mc(k, ir_k)$, 根据

装配域地址为匹配蚂蚁分配一套 $\text{TM}(k, ir_k)$ 和 $\text{CM}(k, ir_k)$ 。

(3) 算法数据初始化。通过界面进行人工输入或调整, 如期望因子, 局部更新参数, 最大迭代次数 $N_{\max}$, 各优化目标的相对权重 $w_1 \sim w_4$ 等。

(4) 迭代循环。每次迭代中, 主线程进行全局搜索, 实时输出参与搜索的一个或多个(并行)装配域; 子线程进行区域搜索, 根据各装配域中蚂蚁所

构造序列的适应值, 获得区域迭代最优蚂蚁集。

(5) 结束搜索。全部迭代完成后, 算法即终止, 默认各区域搜索的最后一输出最优序列的非线性

组合为最终解决方案, 其他序列的非线性组合则为次优解, 以静态曲线展现算法的全局收敛性和区域收敛性, 以供比较、分析, 图 6 为算法的流程图。

图 6 蚁群算法的异步并行搜索流程图

6 实例及比较分析

6.1 应用实例

本文以图 7 的蝶阀装配体为例, 蝶阀共 229 个零部件(包含成品件), 从装配的角度将蝶阀按装配体、子装配体、零件进行层次划分, 其中含有较大尺寸的

零件需要多个相同连接件进行固定、连接和支撑, 将这些并行零件组(主要是螺钉、螺母、垫片、肋板等)定义为装配单元, 同时, 19-电动装置(成品)、20-RV 减速器(成品)和 21-连接支架(成品)作为不再分装配体也被定义为装配单元。创建 PS 装配层次结构树如图 8 所示, 包含 3 个装配域共 50 个装配单元, 并以装配域为单位将 PS 装配关系模型自动提取存于数据库中。

图 7 蝶阀装配体实例

1. 后端盖 2. 垫片 3. 4×石棉填料 16×16 4. 2×轴套 5. 2×肋板 1 6. 轴座 1 7. 盖板 8. 肋板 2 9. 轴座 2 10. 压板 11. 密封圈
12. 圈 13. 52×垫圈 16 14. 2×肋板 3 15. 2×肋板 4 16. 上支撑法兰 17. 16×垫圈 30 18. 16×螺母 M30 19. 电动装置(成品)
20. RV 减速器(成品) 21. 连接支架(成品) 22. 2×端法兰 23. 阀板 24. 阀座 25. 5×圆锥销 26. 16×螺栓 M30×100 27. 上支撑接管
28. 52×螺钉 M16×55 29. 大垫片 30. 环筋 31. 阀杆 32. 20×肋板 5 33. 阀筒 34. 2×肋板 6 35. 下支撑接管 36. 4×肋板 7
37. 2×肋板 8 38. 8×螺母 M20 39. 8×垫圈 20 40. 下支撑法兰 41. 8×螺栓 M20×70

图 8 蝶阀的 PS 层次结构树

图 8 中,各符号含义如下: $A(0,0,0)$: 蝶阀; $A(1,1,0)$: 阀体组件; $A(1,2,0)$: 后端盖; $A(1,3,0)$: 垫片; $A(1,4,0)$: 石棉填料 1 16×16 ; $A(1,5,0)$: 石棉填料 2 16×16 ; $A(1,6,0)$: 石棉填料 3 16×16 ; $A(1,7,0)$: 石棉填料 4 16×16 ; $A(1,8,0)$: 轴套 1; $A(1,9,0)$: 轴套 2; $A(1,10,0)$: $16 \times$ 垫圈 30; $A(1,11,0)$: $16 \times$ 螺母 M30; $A(1,12,0)$: 电动装置(成品); $A(1,13,0)$: RV 减速器(成品); $A(1,14,0)$: 连接支架(成品); $A(1,15,0)$: $16 \times$ 螺栓 M30 $\times$ 100; $A(1,16,0)$: 阀杆; $A(1,17,0)$: $8 \times$ 螺母 M20; $A(1,18,0)$: $8 \times$ 垫圈 20; $A(1,19,0)$: $8 \times$ 螺栓 M20 $\times$ 70; $A(1,20,0)$: 阀板组件; $A(1,21,0)$: $2 \times$ 圆锥销; $A(1,22,0)$: $3 \times$ 圆锥销; $A(2,1,1)$: $2 \times$ 肋板 1; $A(2,2,1)$: $2 \times$ 肋板 3; $A(2,3,1)$: $2 \times$ 肋板 4; $A(2,4,1)$: 端法兰 1; $A(2,5,1)$: 端法兰 2; $A(2,6,1)$: 阀座; $A(2,7,1)$: 上支撑接管; $A(2,8,1)$: 上支撑法兰; $A(2,9,1)$: 环筋; $A(2,10,1)$: $20 \times$ 肋板 5; $A(2,11,1)$: 阀筒; $A(2,12,1)$: 下支撑接管; $A(2,13,1)$: $2 \times$ 肋板 8; $A(2,14,1)$: 下支撑法兰; $A(2,15,20)$: 轴座 1; $A(2,16,20)$: 盖板; $A(2,17,20)$: 肋板 2; $A(2,18,20)$: 轴座 2; $A(2,19,20)$: 压板; $A(2,20,20)$: 密封圈; $A(2,21,20)$: 圈; $A(2,22,20)$: $52 \times$ 垫圈 16; $A(2,23,20)$: 阀板; $A(2,24,20)$: $2 \times$ 肋板 6; $A(2,25,20)$: $52 \times$ 螺钉 M16 $\times$ 55; $A(2,26,20)$: 大垫片; $A(2,27,20)$: $4 \times$ 肋板 7。

6.2 优化结果及其分析

运用本文算法,作者在 VC 环境下采用应用编程接口 OpenMP 开发了 Ap-asp System,即“异步并行装配序列规划系统”,如图 9 所示。Ap-asp System 在

运行时,其主线程可以生成一系列的子线程,并将区域规划任务实时分配给单独子线程进行计算。

图 9 Ap-asp System

设置蚂蚁数 $mc=50$, 迭代次数 $N_{cmax}=100$, 期望因子 $\alpha=2.5$, 启发因子 $\beta_1=1.1$, 信息素残留参数 $\rho=0.5$, 由于蝶阀中相同或相似零部件比重较大, 保证装配过程中相同的操作集中完成, 并行性指标权重 $w_3=0.4$, 其他指标权重设为 0.2。

首先, 根据装配工艺确定装配域 $R(1,1)$, $R(2,1)$ 和 $R(2,2)$ 的装配基础件, 分别为装配单元 $A(1,1,0)$, $A(2,11,1)$ 和 $A(2,23,20)$ 。在 100 次迭代过程中, Ap-asp 算法的全局规划与区域规划优化指标如表 3 所示, 算法全局搜索和区域搜索的收敛曲线如图 10 所示, 根据 Ap-asp 算法规划结果如图 11 和图 12 所示: 装配序列树模型如图 11a 所示, 表达了各装配域间的装配关系以及区域线性序列; 装配域间装配关系的

图 10 Ap-asp 算法的全局和局部收敛曲线

图 11 蝶阀的装配序列树模型及其自相关矩阵

图 12 蝶阀的装配甘特图

形式化表达如图 11b 所示；蝶阀的装配甘特图如图 12 所示，表达了装配单元和装配域的装配进程，装配单元进程主要考虑装配零部件数的影响，忽略了

安装复杂程度、装夹时间等因素，装配域进程由组成装配单元的进程决定。Ap-asp 算法输出结果大体上跟实际结果相符，说明解的质量能满足要求。

表 3 Ap-asp 算法规划各项指标

指标	全局搜索	区域搜索		
		$R(1,1)$	$R(2,1)$	$R(2,2)$
迭代次数	31	27	31	26
蚂蚁数	50	23	14	13
适应度值	30.00	13.80	8.40	7.80

6.3 优化方法比较分析

采用蚁群算法的装配序列规划是目前序列规划的主流算法之一，具有一定的代表性。

对蚁群算法来说，很难按常规的并行算法与串行算法的比较方式(加速比)来衡量并行 ACO 与串行 ACO 孰优孰劣，为了验证 Ap-asp 算法的搜索效率，本文以图 7 的蝶阀为例，将蚁群算法分别按照传统蚁群算法的线性规划策略，文献[11]中采用蚁群算法的层次化搜索策略以及本文 Ap-asp 算法的“全局并行、区域线性”的搜索策略，分别对 PS 层次结构树进行规划，规划结果如表 4 所示。

表 4 不同搜索策略蚁群算法的规划结果

算法	搜索策略	最大迭代次数 $C_{\max}$	装配单元 /个	运算时间 /ms	平均时间 /ms
I	线性	200	48	4 493	4 534
				4 617	
				4 492	
II	逐层	200	50	1 071	1 034
				1 008	
				1 024	
III	异步并行	200	50	861	856
				862	
				846	

与算法 I 比较，算法 II 和算法 III 都是逐层规划，并将蚂蚁的搜索范围限制在其对应装配域范围内，蚂蚁的搜索求解空间明显减少，理论的装配序列数量从 $48! = 1.2 \times 10^{61}$ 降为 $23! \times 14! \times 13! = 1.4 \times 10^{43}$ ，有效地降低蚂蚁搜索的时间复杂度，因此，从表 4 中可以看出，算法 II 和算法 III 的运算效率要明显优于算法 I，同时，这两种算法也符合协同设计的需要。

算法 II 与算法 III 之间进行比较，搜索过程基本一致，但是，算法 III 中的蚂蚁采用并行搜索，即在同一时间段，会存在多个蚂蚁在不同装配域参与搜索，搜索效率明显进一步提高，因此，与算法 II 相比，算法 III 的运算效率更具有优势。

上述实例表明：Ap-asp 算法中将蚂蚁的搜索范围限制在对应的装配域中，即每个部分在一个独立线程上完成的，算法的执行效果相当于将所有装配单元以装配域为单位进行划分，减少了序列组合，

降低了算法的时间复杂度；同时，全局实现多个蚂蚁同时参与搜索，进一步提高蚂蚁的搜索效率，使 Ap-asp 算法具有明显的效率优势。

7 结论

基于“分治”原则提出蚁群算法的“全局并行、区域线性”搜索策略，利用 Ap-asp 算法对 PS 层次结构树进行装配序列的异步并行求解，并开发了 Ap-asp System，以蝶阀产品的装配序列规划为例，从解的质量和收敛速度方面验证 Ap-asp 算法实现高效、结果合理，同时，此方法为并行装配序列规划提供了一种新的思路。

参 考 文 献

- [1] TORRES F, PUENTE S, ARACIL R. Disassembly planning based on precedence relations among assemblies[J]. The International Journal of Advanced Manufacturing Technology, 2003, 21(5): 317-327.
- [2] GOT TIPOLU R B, GHOSH K. A simplified and efficient representation for evaluation and selection of assembly sequences[J]. Computers in Industry, 2003, 50(3): 251-264.
- [3] SMITH G C, SMITH S S F. An enhanced genetic algorithm for automated assembly planning[J]. Robotics and Computer Integrated Manufacturing, 2002, 18(5/6): 355-364.
- [4] TSENG, Y J, YU F Y, HUANG F Y. A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method[J]. International Journal of Advanced Manufacturing Technology, 2011, 57(9-12): 1183-1197.
- [5] 于嘉鹏, 王成恩, 王健熙. 基于最大-最小蚁群系统的装配序列规划[J]. 机械工程学报, 2012, 48(23): 152-166.
YU Jiapeng, WANG Chengen, WANG Jianxi. Assembly sequence planning based on max-min ant colony system[J]. Journal of Mechanical Engineering, 2012, 48(23): 152-166.
- [6] 刘亚杰, 古天龙, 徐周波, 等. 基于改进遗传编程的并行装配序列规划[J]. 计算机集成制造系统, 2013, 19(6): 1238-1248.
LIU Yajie, GU Tianlong, XU Zhoubo, et al. Parallel assembly sequence planning based on improved genetic programming[J]. Computer Integrated Manufacturing Systems, 2013, 19(6): 1238-1248.
- [7] 胡小梅, 朱文华, 俞涛. 基于模糊粗糙集的并行装配序

- 列规划方法[J]. 机械工程学报, 2010, 46(15): 130-135.
- HU Xiaomei, ZHU Wenhua, YU Tao. Assembly sequence planning method based on fuzzy rough sets[J]. Journal of Mechanical Engineering, 2010, 46(15): 130-135.
- [8] 杨波, 黄克正, 王慧, 等. 基于约束解除的装配序列并行化优化[J]. 计算机集成制造系统, 2004, 10(7): 832-837.
- YANG Bo, HUANG Kezheng, WANG Hui, et al. Concurrent optimization of assembly sequence based on constraint release strategy[J]. Computer Integrated Manufacturing Systems, 2004, 10(7): 832-837.
- [9] 张秀芬, 蔚刚, 王磊, 等. 支持复杂产品并行拆卸序列规划的遗传算法[J]. 计算机辅助设计与图形学学报, 2015, 27(7): 1327-1333.
- ZHANG Xiufen, YU Gang, WANG Lei, et al. Parallel disassembly sequence planning for complex products based on genetic algorithm[J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(7): 1327-1333.
- [10] 王舒卓. 并行蚁群算法的研究[D]. 沈阳: 东北大学, 2013.
- WANG Shuzhuo. Research on parallel ant colony algorithm[D]. Shenyang: Northeastern University, 2013.
- [11] 于嘉鹏, 王健熙. 基于递归循环的层次化爆炸图自动生成方法[J]. 机械工程学报, 2016, 52(13): 175-188.
- YU Jiapeng, WANG Jianxi. Automatic generation of hierarchical exploded view based on recursive iteration method[J]. Journal of Mechanical Engineering, 2016, 52(13): 175-188.
- [12] PAVLOV V V. Polychromatic sets and graphs for CALS[M]. Moscow: STANKIN Press, 2002.
- [13] 赵姗姗, 李宗斌. 基于多色集合的装配序列形式化推理方法[J]. 计算机集成制造系统, 2008, 14(8): 1579-1616.
- ZHAO Shanshan, LI Zongbin. Formalization reasoning method for assembly sequences based on polychromatic sets[J]. Computer Integrated Manufacturing Systems, 2008, 14(8): 1579-1616.
- [14] 刘翊, 李世其, 王峻峰, 等. 产品分层分级的交互式拆卸装配序列规划[J]. 计算机集成制造系统, 2014, 20(4): 785-792.
- LIU Yi, LI Shiqi, WANG Junfeng, et al. Interactive disassembly approach for assembly sequence planning based on product hierarchical-classification[J]. Computer Integrated Manufacturing Systems, 2014, 20(4): 785-792.
- [15] 杨剑锋. 蚁群算法及其应用研究[D]. 杭州: 浙江大学, 2007.
- YANG Jianfeng. Ant colony algorithm and its application[D]. Hangzhou: Zhejiang University, 2007.
- [16] 于嘉鹏, 王成恩, 张闻雷, 等. 基于优先规则筛选的装配序列规划方法[J]. 东北大学学报, 2009, 30(11): 1636-1640.
- YU Jiapeng, WANG Chengen, ZHANG Wenlei, et al. Assembly sequence planning based on priority rulescreening[J]. Journal of Northeastern University, 2009, 30(11): 1636-1640.
-
- 作者简介: 刘晓阳, 男, 1985 年出生, 讲师。主要研究方向为 CAD/CAM, 制造业信息化等。
E-mail: lxy_841211@163.com
- 刘恩福, 男, 1960 年出生, 教授, 硕士研究生导师。主要研究方向为 CAD/CAM, 数字化设计等。
E-mail: liuef@hebust.edu.cn
- 靳江艳, 女, 1977 年出生, 博士, 讲师。主要研究方向为数字化设计与制造, 装配工艺规划等。
E-mail: 121602570@qq.com