DOI: 10.3901/JME.2019.17.185

基于最大公共子图挖掘和装夹性能分析的 夹具耦合设计方法*

秦国华¹ 邱剑鹏¹ 王华敏^{1,2} 吴铁军³

(1. 南昌航空大学航空工程学院 南昌 330063;

2. 南京航空航天大学机电学院 南京 210016;

3. 东莞职业技术学院机电工程系 东莞 523808)

摘要: 大量的夹具案例积累了可重用的经验和知识, 为新零件实现夹具设计提供了理论基础和决策参考, 而准确的夹具检索和合理的装夹判断是成功完成夹具设计的核心与关键。为此, 根据零件 B-rep 表示的实体模型, 通过特征之间的几何拓扑关系构造零件的标号图。结合标号图的线性顺序性, 提出了标定索引值的 DFS 词典序原则, 建立了最小 DFS 编码的遍历方法。其次, 在挖掘出零件最小 DFS 编码的基础上, 依据边编码中特征值、属性值及特征之间的连接关系, 提出零件之间最大公共子图的挖掘方法, 通过定义零件相似度和库相似度两个概念, 构造出相似夹具检索和排序的决策因子。最后, 通过建立定位确定性和装夹稳定性分析算法, 依据决策因子大小逐一评估夹具对新零件的合理装夹, 直至通过装夹性能的检验, 完成夹具的设计过程。由于与零件一一对应的最小 DFS 编码, 其搜索方法为图论中用递归容易实现的经典算法, 而装夹性能评估只涉及点的坐标和法矢量信息, 则属于可编程实现的离散型问题, 故提出的“先检索排序夹具, 后验证装夹性能”的夹具耦合设计方法, 不仅能丰富和拓展计算机辅助夹具设计方法和理论, 而且还将大大促进计算机辅助夹具设计系统的实用性。

关键词: 最小 DFS 编码; 最大公共子图; 挖掘算法; 决策因子; 装夹性能

中图分类号: TG702

Couple Approach to Fixture Design Based on Maximum Common Subgraph Mining and Fixturing Performance Analysis

QIN Guohua¹ QIU Jianpeng¹ WANG Huamin^{1,2} WU Tiejun³

(1. School of Aeronautical Manufacturing Engineering, Nanchang Hangkong University, Nanchang 330063;

2. College of Mechanical & Electrical Engineering,

Nanjing University of Aeronautics and Astronautics, Nanjing 210016;

3. Department of Mechanical and Electrical Engineering, Dongguan Polytechni, Dongguan 523808)

Abstract: A numerous of fixture cases, which accumulate reusable experience and knowledge, can supply the theoretical basis and decision criterion for fixture design of a new workpiece. However, the accurate search of similar fixtures from fixture cases is key to the achievement of fixture design, in addition that the searched fixtures can successfully grasp the new workpiece. Therefore, based on the entity B-Rep (i.e., Boundary Representation) model of a workpiece, the labeled graph of a workpiece is constructed according to the geometrical and topological relationships between features. In combination with the linear sequence of labeled graph, the principles of the DFS dictionary sequence, which can calibrate the index value for an arbitrary feature, are proposed to traversal the minimum DFS code. Again, based on the minimum DFS code, the mining method of maximum common subgraph is suggested for workpieces according to the feature value, performance value and link relationship between features. Furthermore, two concepts on the workpiece similarity and library similarity are defined to formulize the decision factor for the sequence of the searched fixtures. The final step is to formulize the locating determination and the fixturing stability so that the searched fixtures are one by one evaluated the fixturing performance for a new workpiece. the design process of fixture is over when and only when a fixture achieves

* 国家自然科学基金(51465045, 51765047)、江西省主要学科科学技术带头人资助计划(20172BCB22013)、航空科学基金(2016ZE56011)、广东省教育厅特色创新类项目(2017GKTSCX10 2)、江西省教育厅科技项目(GJJ170572)和南昌市科技重大(2018-ZDKJGG-010)资助项目。20180910收到初稿, 20190531收到修改稿

the good fixturing performance. In fact, the minimum DFS code and a workpiece can hold a unique one to one relationship. The search method of DFS code, which is classic algorithm in graph theory, can be easily realized by the recursion method. On other hand, the fixturing performance evaluation only concern on the coordinates and the corresponding normal vectors. It belongs to the programmed discretization problem. So the presented "it is crucial for the searched fixture to satisfy the fixturing performance" method is not only well enrich the theory of computer aided fixture design, but greatly accelerate its practicability.

Key words: minimum DFS code; maximum common subgraph; mining algorithm; decision factor; fixturing performance

0 前言

除了定位工件,让工件获得相对于机床和刀具的正确位置外,夹具的主要作用旨在提供合适的夹紧力,抵抗零件在加工过程中受到的切削力和切削扭矩等外力的影响,以便保证零件的可靠装夹、加工精度和生产安全^[1-2]。在整个生产过程中,生产准备时间要占生产制造周期的 50%~70%,其中夹具的设计与制造占生产准备时间的 70%左右^[3]。由此可见,夹具的设计与制造不仅极大地影响着零件的加工质量,而且还是决定产品开发周期的关键因素。

大量的夹具案例为夹具设计提供了参考和经验,这是提高夹具设计质量和效率的基础。为此,夹具的可重用性极大地吸引着专家、学者、工程技术人员的注意和兴趣。事实上,早在 20 世纪 70 年代,就开始了夹具 CAD 的研究工作,由于计算机硬件和软件的限制,该工作仅仅是夹具的计算机绘图。随着计算机和信息技术的快速发展,国内外专家学者结合各种先进设计方法,形成了交互式、检索式、生成式等计算机辅助夹具设计(Computer aided fixture design, CAFD)方法。无论哪种夹具设计方法,其核心部分就是工件或夹具 CAD 模型的检索。迄今为止,企业中 CAD 模型的检索依然停留在基于编码^[4-7]、知识^[8-12]的传统检索方式上。

由于受到定义规则等一些主观因素以及有些复杂 CAD 模型难以用文字充分描述的影响^[13-15],使得这种传统的检索方法虽然简单、易于实现,但可靠性不高,为此一些学者提出了基于形状的 CAD 模型检索方法。CICIRELLO、ELINSON 等^[16-17]依据从基于 CSG 结构的 CAD 模型中识别出来的加工特征,通过构建加工特征及其相互之间依赖关系的模型依赖图(Model dependency graph, MDG),得到图的邻接矩阵。然后根据父图顶点的度不小于子图顶点的度的包容关系,提出子、父之间映射矩阵的深度优先搜索方法。最后通过映射父图的邻接矩阵获得匹配矩阵,根据匹配矩阵不小于子图的邻接矩阵这一条件,提出子图同构的贪婪子图同构检索方法。事实上,基于 CGS 结构的 CAD 模型表示不唯

一,无法支持相似性比较^[18]。张欣等^[19]提取 CAD 模型的 B-Rep 信息后,将其转化为属性邻接图(Attributed adjacency graph, AAG),依据各顶点之间的拓扑关系构建邻接矩阵。然后通过求解矩阵的最大特征矢量,确定图顶点集的序列,并依据序列之间的编辑距离构造了图的距离。然后以图的距离为自变量构建近似概率,通过最大近似概率为目标实现公共子图的匹配问题。由于该匹配算法主要是通过比较模型的拓扑结构,对 CAD 模型的几何信息考虑较少,造成模型相似性比较的程度相对粗糙,而且利用相似概率进行匹配亦将导致匹配结果不具有唯一性,而降低准确性。

为此,本文在 B-rep 表示的实体模型基础上构造零件的标号图,通过研究索引值的标定方法,提出与零件一一对应的最小深度优先搜索(Depth first search, DFS)编码的确定算法。根据公共子图的挖掘规则,建立最大公共子图挖掘的夹具排序的决策因子方法。通过研究装夹性能的分析与评估方法,实现所检索出来的夹具案例满足新零件加工要求的验证,最终完成夹具设计的目的。

1 标号图

通过提取 CAD 模型中的 B-rep 信息,可将一个零件等价地表示为一个标号图。若将标号图中的边 $E_{ij} = (V_i, V_j)$ 表示为 $(V_i, V_j, F_i, F_j, R_{ij})$, 那么一个零件的标号图可用一个 5 元组集合来表示,其中 V_i 、 V_j 为特征标识,表示第 i 、 j 个顶点, F_i 、 F_j 为特征标号,表示第 i 、 j 个顶点相对应的表面特征,而 R_{ij} 为边标号,用以表示第 i 个与第 j 个顶点之间的特征关系。

1.1 标号图的 DFS 编码

一般地,零件的表面特征以 P 表示平面, C 表示柱面, H 表示孔面, S 表示球面, ...。为了能够清晰地描述 DFS 的遍历过程,这里仅考虑凹(在实体外所成的角度小于 180°)与凸(在实体外所成的角度大于 180°)的关系,记 0 为表面与表面之间为凸关系, 1 为表面与表面之间为凹关系。

在将标号图映射为 DFS 编码的过程中,应注意遍历的线性顺序性^[20]。若先遍历顶点 V_i , 后遍历顶

图 2 无分支标号图

特征值最小原则：若顶点的特征值越小，则索引值越小。

为了便于说明最小 DFS 编码的挖掘过程，利用大写的罗马数字、带圈的阿拉伯数字分别为顶点和边进行编号，如图 2a 所示。从顶点 I 沿前向边(即一条边引入一个未遍历过的顶点，则称该边为前向边)①、③进行第 1 次挖掘，能够同时挖到 II、IV 两个顶点。由于特征值和属性值均相等，故不能确定顶点 II、IV 的索引值。继续沿前向边②、④进行第 2 次挖掘，发现顶点 III 比顶点 V 具有更小的特征值，故可优先确定顶点 III 的索引值，即 $III=i+2$ ，遍历过程如表 2 所示。由此可以看出，表 2 中遍历方案 1 获得的 DFS 编码是最小的。

表 2 特征值最小的 DFS 编码遍历过程

遍历方案 1			遍历方案 2		
边	深度	DFS 编码	边	深度	DFS 编码
①	1	$(V_i, V_{II}, P, P, 0)$	③	1	$(V_i, V_{IV}, P, P, 0)$
②	2	$(V_{II}, V_{III}, P, P, 0)$	④	2	$(V_{IV}, V_V, P, C, 0)$
③	1	$(V_i, V_{IV}, P, P, 0)$	①	1	$(V_i, V_{II}, P, P, 0)$
④	2	$(V_{IV}, V_V, P, C, 0)$	②	2	$(V_{II}, V_{III}, P, P, 0)$

属性值最小原则：若顶点属性值越小，则索引值越小。

如图 2b 所示，由顶点 I 沿第 2 条前向边①、③挖掘到顶点 II、IV，特征值不但相等，而且属性值也相等，至此依然不能确定顶点 II、IV 的索引值。继续分别挖掘前向边②、④的顶点 III 和顶点 V。由于顶点 III 的属性值为 1 而顶点 V 为 0，因此优先确定顶点 V 的索引值为 $V=i+2$ 。随后其他顶点的索引值亦可确定，如表 3 的遍历方案 2，由此挖掘而形成的 DFS 编码即为最小的 DFS 编码。

表 3 属性值最小的 DFS 编码遍历过程

遍历方案 1			遍历方案 2		
边	深度	DFS 编码	边	深度	DFS 编码
①	1	$(V_i, V_{II}, P, P, 0)$	③	1	$(V_i, V_{IV}, P, P, 0)$
②	2	$(V_{II}, V_{III}, P, P, 1)$	④	2	$(V_{IV}, V_V, P, P, 0)$
③	1	$(V_i, V_{IV}, P, P, 0)$	①	1	$(V_i, V_{II}, P, C, 0)$
④	2	$(V_{IV}, V_V, P, P, 0)$	②	2	$(V_{II}, V_{III}, P, P, 1)$

长度最短原则：若顶点的分支长度越小，则索引值越小。

如图 3a 所示，由顶点 I 开始挖掘，分别沿分支“①、②”和分支“③、④”挖掘至顶点 IV、V，此时遍历到的 DFS 编码分别为 $(V_i, V_{II}, P, P, 0)$ 、 $(V_{II}, V_{IV}, P, P, 0)$ 和 $(V_i, V_{III}, P, P, 0)$ 、 $(V_{III}, V_V, P, P, 0)$ ，其中无论是特征值，还是属性值，均相同，至此无法确定当前顶点 IV、V 的索引值。然而，由于顶点 IV

的分支长度为 2，而顶点 V 的分支长度为 3，故可优先确定顶点 IV 的索引值为 $IV=i+2$ 。

(a) 具有后向边的顶点

(b) 具有前向边的顶点

图 3 具有最短长度的标号图

这是因为，顶点 IV 为终点，既不存在前向边也不存在后向边(如果一条边引入一个已遍历过的顶点，则称其为后向边)，只能从顶点 I 沿着前向边③继续遍历，获得的 DFS 编码为 $(V_i, V_{II}, P, P, 0)$ 、 $(V_{II}, V_{IV}, P, P, 0)$ 、 $(V_i, V_{III}, P, P, 0)$ ，而顶点 V 存在后向边，可继续沿着后向边⑤进行遍历，获得的 DFS 编码则为 $(V_i, V_{III}, P, P, 0)$ 、 $(V_{III}, V_V, P, P, 0)$ 、 $(V_V, V_i, P, P, 0)$ 。根据线性顺序性可知 $(V_i, V_{III}, P, P, 0) < (V_V, V_i, P, P, 0)$ ，显然由 $(V_i, V_{II}, P, P, 0)$ 、 $(V_{II}, V_{IV}, P, P, 0)$ 、 $(V_i, V_{III}, P, P, 0)$ 形成的 DFS 编码最小，应为 $I=i$ ，因此 $II=i+1$ 、 $IV=i+2$ 、 $III=i+3$ 、 $V=i+4$ ，如表 4 中的遍历方案 1。

表 4 长度最短的 DFS 编码遍历过程

遍历方案 1			遍历方案 2		
边	深度	DFS 编码	边	深度	DFS 编码
①	1	$(V_i, V_{II}, P, P, 0)$	③	1	$(V_i, V_{III}, P, P, 0)$
②	2	$(V_{II}, V_{IV}, P, P, 0)$	④	2	$(V_{III}, V_V, P, P, 0)$
③	1	$(V_i, V_{III}, P, P, 0)$	⑤	0	$(V_V, V_i, P, P, 0)$
④	2	$(V_{III}, V_V, P, P, 0)$	①	1	$(V_i, V_{II}, P, P, 0)$
⑤	0	$(V_V, V_i, P, P, 0)$	②	2	$(V_{II}, V_{IV}, P, P, 0)$

而标号图 3b 中，由于顶点 V 引存在前向边⑤，使得顶点 IV 的分支长度小于顶点 VI 的分支长度，故 $IV=i+2$ ，其他顶点的索引值亦随之可确定下来，即有 $II=i+1$ 、 $III=i+3$ 、 $V=i+4$ 、 $VI=i+5$ 。

深度最小原则：若顶点的深度越小，则索引值越小。

标号图 4a 的挖掘过程见表 5，从顶点 I 分别沿分支“①、②、③”和分支“④、⑤、⑥”挖掘至当前顶点 VI、I 时，不仅特征值和属性值相同，而且分支长度也均为 4。但由于前向边③上前向顶点 VI 的深度序号为 3，而后向边上后向顶点 I 的深度序号为 0，故应优先确定分支“④、⑤、⑥”上顶点 I 的索引值。由于顶点 I 的索引值为 i ，故接下来确定顶点 V 的索引值。由于顶点 V 的深度序号为 2，

则顶点 V 的索引值应为 $V=i+2$ 。

表 5 即有前向边又有后向边的 DFS 编码比较

遍历方案 1			遍历方案 2		
边	深度	DFS 编码	边	深度	DFS 编码
①	1	$(V_i, V_{II}, P, P, 0)$	④	1	$(V_i, V_{III}, P, P, 0)$
②	2	$(V_{II}, V_{IV}, P, P, 0)$	⑤	2	$(V_{III}, V_V, P, P, 0)$
③	3	$(V_{IV}, V_{VI}, P, P, 0)$	⑥	0	$(V_V, V_I, P, P, 0)$
④	1	$(V_i, V_{III}, P, P, 0)$	①	1	$(V_i, V_{II}, P, P, 0)$
⑤	2	$(V_{III}, V_V, P, P, 0)$	②	2	$(V_{II}, V_{IV}, P, P, 0)$
⑥	0	$(V_V, V_I, P, P, 0)$	③	3	$(V_{IV}, V_{VI}, P, P, 0)$

同样，表 6 为标号图 4b 映射为 DFS 编码的所有搜索方案，由于分支“①、②、③、④”和分支“⑤、⑥、⑦、⑧”上顶点 II、I 的深度序号分别为 1 和 0，故应优先确定分支“⑤、⑥、⑦、⑧”的索引值，由于当前顶点 I 的索引值为 i ，顶点 VII 的深度序号为 3，故 VII 的索引值为 $V_{II}=i+3$ 。

图 4 具有最小深度的标号图

表 6 仅有后向边的 DFS 编码比较

遍历方案 1			遍历方案 2		
边	深度	编码 1	边	深度	编码 2
①	1	$(V_i, V_{II}, P, P, 0)$	⑤	1	$(V_i, V_{III}, P, P, 0)$
②	2	$(V_{II}, V_{IV}, P, P, 0)$	⑥	2	$(V_{III}, V_V, P, P, 0)$
③	3	$(V_{IV}, V_{VI}, P, P, 0)$	⑦	3	$(V_V, V_{VII}, P, P, 0)$
④	1	$(V_{VI}, V_{II}, P, P, 0)$	⑧	0	$(V_{VII}, V_I, P, P, 0)$
⑤	1	$(V_i, V_{III}, P, P, 0)$	①	1	$(V_i, V_{II}, P, P, 0)$
⑥	2	$(V_{III}, V_V, P, P, 0)$	②	2	$(V_{II}, V_{IV}, P, P, 0)$
⑦	3	$(V_V, V_{VII}, P, P, 0)$	③	3	$(V_{IV}, V_{VI}, P, P, 0)$

分支最早原则：若当前顶点出现分支越早，则当前顶点的索引值越小。

自顶点 I 开始挖掘，直至顶点 VII、VIII、IX、X 时，由于顶点 VIII 的特征值和顶点 X 的属性值大于顶点 VII、X，故优先遍历分支“①、②、③”和分支“⑤、⑥、⑦”，如图 5 所示。然而，挖掘至当前顶点 VII、X 时，特征值、属性值，以及分支长度、深度序号，均相同，显然无法确定当前顶点的索引值。但是，分支“①、②、③”出现分支的顶点为 IV，深度为 2，而分支“⑤、⑥、⑦”出现分支的顶点为 III，其深度为 1，故应优先确定分支“⑤、

⑥、⑦”的索引值，由于当前顶点 IX 的深度序号为 3，故索引值为 $IX=i+3$ ，由此挖掘到的最小 DFS 编码为表 7 中的遍历方案 1。

图 5 最早出现分支的标号图

表 7 分支最早的 DFS 编码比较

遍历方案 1			遍历方案 2		
边	深度	编码 1	边	深度	编码 2
①	1	$(V_i, V_{II}, P, P, 0)$	⑤	1	$(V_i, V_{III}, P, P, 0)$
②	2	$(V_{II}, V_{IV}, P, P, 0)$	⑥	2	$(V_{III}, V_V, P, P, 0)$
③	3	$(V_{IV}, V_{VI}, P, P, 0)$	⑦	3	$(V_V, V_{IX}, P, P, 0)$
④	3	$(V_{IV}, V_{VIII}, P, C, 0)$	⑧	2	$(V_{III}, V_{VI}, P, P, 0)$
⑤	2	$(V_i, V_{III}, P, P, 0)$	⑨	3	$(V_{VI}, V_X, P, P, 1)$
⑥	3	$(V_{III}, V_V, P, P, 0)$	①	1	$(V_i, V_{II}, P, P, 0)$
⑦	4	$(V_V, V_{IX}, P, P, 0)$	②	2	$(V_{II}, V_{IV}, P, P, 0)$
⑧	3	$(V_{III}, V_{VI}, P, P, 0)$	③	3	$(V_{IV}, V_{VI}, P, P, 0)$
⑨	4	$(V_{VI}, V_X, P, P, 1)$	④	3	$(V_{IV}, V_{VIII}, P, C, 0)$

分支最少原则：若分支中当前顶点的出度越小，则该条分支中当前顶点的索引值越小。

如图 6a 所示，由顶点 I 开始，分别沿着分支“①、②、③”和分支“④、⑤、⑥”挖掘至当前顶点 VI、

(a) 存在无分支的情况

(b) 均存在分支的情况

图 6 具有最少分支的标号图

VII 时, 不论是特征值、属性值, 还是分支长度、深度序号均相同, 显然无法确定当前顶点的索引值。但是, 分支“①、②、③”中顶点 IV 的出度为 1, 分支“④、⑤、⑥”中顶点 V 的出度为 2, 故应优先确定分支“①、②、③”的索引值, 由于当前顶点 VI 的深度序号为 3, 故索引值为 $VI=i+3$, 由此挖掘到的最小 DFS 编码为表 8 中的遍历方案 1。

表 8 存在无分支的 DFS 编码比较

遍历方案 1				遍历方案 2			
边	深度	DFS 编码	出度	边	深度	DFS 编码	出度
①	1	$(V_I, V_{II}, P, P, 0)$	1	④	1	$(V_I, V_{III}, P, P, 0)$	1
②	2	$(V_{II}, V_{IV}, P, P, 0)$	1	⑤	2	$(V_{III}, V_V, P, P, 0)$	1
③	3	$(V_{IV}, V_{VI}, P, P, 0)$	1	⑥	3	$(V_V, V_{VII}, P, P, 0)$	2
④	1	$(V_I, V_{III}, P, P, 0)$	1	⑦	3	$(V_V, V_{VIII}, P, C, 0)$	2
⑤	2	$(V_{III}, V_V, P, P, 0)$	1	①	1	$(V_I, V_{II}, P, P, 0)$	1
⑥	3	$(V_V, V_{VII}, P, P, 0)$	1	②	2	$(V_{II}, V_{IV}, P, P, 0)$	1
⑦	3	$(V_V, V_{VIII}, P, C, 0)$	2	③	3	$(V_{IV}, V_{VI}, P, P, 0)$	1

图 6b 中, 由于顶点 IX、X、XI、XII、XIII 中 IX 和 XII 具有最小特征值, 故优先遍历。但分支“①、②、③”和分支“④、⑤、⑥”自顶点 I 开始挖掘至当前顶点 IX、XII 时, 不论是特征值、属性值, 还是分支长度、深度序号均相同, 显然无法确定当前顶点的索引值。但是, 分支“①、②、③”中顶点 II 的出度为 3, 分支“④、⑤、⑥”顶点 III 的出度为 2, 故应优先确定分支“④、⑤、⑥”的索引值, 由于当前顶点 XII 的深度序号为 4, 故索引值为 $VI=i+4$, 由此挖掘到的最小 DFS 编码为表 9 中的遍历方案 2。

表 9 均存在分支的 DFS 编码比较

遍历方案 1				遍历方案 2			
边	深度	编码 1	出度	边	深度	编码 2	出度
①	1	$(V_I, V_{II}, P, P, 0)$	1	⑧	1	$(V_I, V_{III}, P, P, 0)$	1
②	2	$(V_{II}, V_{IV}, P, P, 0)$	3	⑨	2	$(V_{III}, V_{VII}, P, P, 0)$	2
③	3	$(V_{IV}, V_{IX}, P, P, 0)$	1	⑩	3	$(V_{VII}, V_{XII}, P, P, 0)$	1
④	2	$(V_{II}, V_V, P, P, 0)$	3	⑪	2	$(V_{III}, V_{VIII}, P, P, 0)$	2
⑤	3	$(V_V, V_X, P, C, 0)$	1	⑫	3	$(V_{VIII}, V_{XIII}, P, P, 0)$	1
⑥	2	$(V_{II}, V_{VI}, P, P, 0)$	3	①	1	$(V_I, V_{II}, P, P, 0)$	1
⑦	3	$(V_{VI}, V_{XI}, P, H, 0)$	1	②	2	$(V_{II}, V_{IV}, P, P, 0)$	3
⑧	1	$(V_I, V_{III}, P, P, 0)$	1	③	3	$(V_{IV}, V_{IX}, P, P, 0)$	1
⑨	2	$(V_{III}, V_{VII}, P, P, 0)$	2	④	2	$(V_{II}, V_V, P, P, 0)$	3
⑩	3	$(V_{VII}, V_{XII}, P, P, 0)$	1	⑤	3	$(V_V, V_X, P, C, 0)$	1
⑪	2	$(V_{III}, V_{VIII}, P, P, 0)$	2	⑥	2	$(V_{II}, V_{VI}, P, P, 0)$	3
⑫	3	$(V_{VIII}, V_{XIII}, P, P, 0)$	1	⑦	3	$(V_{VI}, V_{XI}, P, H, 0)$	1

綜上述 6 条原则, 可得如图 7 所示的特征标识确定流程, 图中 i 表示特征值最小原则, ii 表示属性值最小原则, iii 表示深度最小原则, iv 表示长度

最短原则, v 表示分支最少原则, vi 表示分支最少原则。

图 7 索引值的确定流程

2 公共子图

假定夹具库中存储有 n 个夹具案例, 各夹具对应的零件图集为 $G_{obj}=\{G_i \mid G_i=(V_{st}^i, V_{ed}^i, F_{st}^i, F_{ed}^i, R^i), 1 \leq i \leq n\}$ (其中, 对于第 i 个零件的标号图来说, V_{st}^i 、 V_{ed}^i 为顶点集合, F_{st}^i 、 F_{ed}^i 为特征标号集合, R^i 为边标号集合)。再假定一个新零件, 其标号图表示为 $G_{new}=(V_{st}, V_{ed}, F_{st}, F_{ed}, R)$ 。那么, 所谓图 G_{new} 的子图 $g=(v_{st}, v_{ed}, f_{st}, f_{ed}, r)$, 指的是其所有的顶点和边都属于图 G_{new} 的图, 即存在 $f_{st} \subseteq F_{st}$, $f_{ed} \subseteq F_{ed}$ 和 $r \subseteq R$ 。若 g 同时属于图 G_i 和图 G_{new} 的子图, 则称之为 G_i 与 G_{new} 的公共子图。

2.1 公共子图的挖掘

挖掘公共子图的本质, 就是在新零件 G_{new} 的最小 DFS 编码中, 从第一条边编码 $(V_1, V_2, F_1, F_2, R_{12})$ 开始, 与图集 G_{obj} 中第 i 个零件 G_i 进行编码匹配。当执行到任意条边编码 $(V_\alpha, V_\beta, F_\alpha, F_\beta, R_{\alpha\beta})$ 时, 与图集 G_{obj} 中第 i 个零件 G_i 进行匹配编码, 若 G_i 的最小 DFS 编码含有 $(u, v, F_\alpha^i, F_\beta^i, R_{\alpha\beta}^i)$, 则分别在 G_{new} 、 G_i

中保留边($V_\alpha, V_\beta, F_\alpha, F_\beta, R_{\alpha\beta}$)、边($u, v, F_\alpha^i, F_\beta^i, R_{\alpha\beta}^i$), 否则去除, 直至 G_{new} 最小 DFS 编码的最后一条边, 其中 u 和 v 表示任意值, 而保留的所有边组成的子图即为 G_{new} 和 G_i 的公共子图。

假定图集包含有 3 个零件, 如图 8 所示, 按照图 7 的流程可以确定出各边的索引值。将特征标号用数字替换(即 $P=1, C=2, H=3, \dots$)后, 各零件的标号图即可获得。再依据最小 DFS 编码的挖掘算

法与原则, 可遍历出图集中各零件的最小 DFS 编码, 如表 10 所示。类似地, 可获得图 9 中新零件的最小 DFS 编码。为了在图 8 的图集中找出图 9 的公共子图, 应按照最小 DFS 编码, 由第一条边开始匹配, 随后逐边添加进行比较。那么, 匹配过程中如何确定第一条相似边, 又如何添加下一条边呢? 通过注意下列三个关键的挖掘规则, 即可挖掘出公共子图。

(a) 连杆

(b) 直筒

(c) 支架

图 8 零件图集

图 9 套筒

边编码相似规则：若边编码的特征值和属性值相同，则挖掘出公共子图的第一条相似边。

表 10 为图(8,9)零件的最小 DFS 编码，其中套筒的第 1 条边编码为(0, 1, 1, 2, 0)，而连杆的第 6 条边编码为(4, 5, 1, 2, 0)、直筒的第 1 条边编码为(0, 1, 1, 2, 0)、支架的第 9 条边编码为(4, 5, 1, 2, 0)，这四条边边编码中特征值和特征关系值均相同，故认为它们是相似的边。

表 10 套筒与库中零件的最小 DFS 编码

边	零件图集			套筒
	连杆	直筒	支架	
①	(0,1,1,1,0)	(0,1,1,2,0)	(0,1,1,1,0)	(0,1,1,2,0)
②	(1,2,1,1,0)	(1,2,2,1,0)	(1,2,1,1,0)	(1,2,2,1,0)
③	(2,3,1,1,0)	(2,3,1,3,0)	(2,0,1,1,0)	(2,3,1,2,1)
④	(3,0,1,1,0)	(3,0,3,1,0)	(2,3,1,1,0)	(3,4,2,1,0)
⑤	(3,4,1,1,0)	(3,4,3,3,0)	(3,0,1,1,0)	(4,5,1,3,0)
⑥	(4,5,1,2,0)	(4,1,3,2,0)	(3,4,1,1,0)	(5,0,3,1,0)
⑦	(5,0,2,1,1)		(4,0,1,1,0)	(5,6,3,3,0)
⑧	(5,1,2,1,1)		(4,1,1,1,0)	(6,3,3,2,0)
⑨	(5,2,2,1,1)		(4,5,1,2,0)	
⑩	(5,6,2,1,0)		(5,1,2,1,0)	
⑪	(6,7,1,3,0)		(5,2,2,1,0)	
⑫	(7,4,3,1,0)		(5,3,2,1,0)	
⑬	(7,8,3,3,2)		(3,6,1,3,0)	
⑭	(8,9,3,1,0)		(6,1,3,1,0)	
⑮	(9,10,1,2,0)		(0,7,1,2,1)	
⑯	(10,0,2,1,1)		(7,8,2,1,0)	
⑰	(10,1,2,1,1)			
⑱	(10,2,2,1,1)			
⑲	(10,3,2,1,1)			
⑳	(10,11,2,1,0)			
㉑	(11,8,1,3,0)			

边编码连接规则：若相邻两条边编码的连接关系相同，则挖掘出公共子图的下一条相似边。

若相邻两条边编码相似，除了它们各自的特征值和属性值相同外，下一条边编码与上一条边编码还必须存在相同的连接关系。比如，支架的通孔与两端平面连接部分编码为 $(3, 6, 1, 3, 0) \rightarrow (6, 1, 3, 1, 0)$ ，套筒的通孔与两端平面连接部分边编码为 $(4, 5, 1, 3, 0) \rightarrow (5, 0, 3, 1, 0)$ ，由于第一条边编码均含有相同的特征值和特征关系值 1、3、0，第二条边编码均含有相同的特征值和特征关系值 3、1、0，而且它们之间存在相同的连接关系(即上一条边编码中第 2 个特征标识与第二条边编码中第 1 个特征标识相同，这种连接关系称之为串联关系)，那么可以认为它们是相似的。又如，连杆的第 5 条边编码 $(3, 4, 1, 1, 0)$ 和第 6 条边编码 $(4, 5, 1, 2, 0)$ 、支架的第 8 条边编码 $(4, 1, 1, 1, 0)$ 和第 9 条边编码 $(4, 5, 1, 2, 0)$ ，它们的第 1 条边编码特征值和属性值相同，都是 1、1、0，第 2 条边编码特征值和特征关系值也相同都是 1、2、0，而且它们第 1 条边和第 2 条边也各自有连接关系。但是，连杆的第 3 条边编码中第 2 个特征标识和第 9 条边编码中第 1 个特征标识相同，属于串联关系，而图 8c 的第 6 条编码中第 1 个特征标识和第 7 条边编码中第 1 个特征标识相同，属于并联关系。所以它们不是相似的。再如，连杆的第 6 条边编码 $(4, 5, 1, 2, 0)$ 、第 20 条边编码 $(10, 11, 2, 1, 0)$ 分别和套筒的第 1 条边编码 $(0, 1, 1, 2, 0)$ 、第 2 条边编码 $(1, 2, 2, 1, 0)$ 的特征值和属性值一样，但连杆的两条边编码不相连，而套筒的两条边编码形成串联，由此可知，这两个图的两条边编码也不相似。

边编码顺序规则：若边编码中第一个特征值大于第二个特征值，则两个特征值及其相应的索引值更换位置。

挖掘图形的最小 DFS 编码时，是按照顶点的线性顺序性进行遍历完成的。因此，不同图形的相似部分，可能由于遍历顺序的不同，导致相似部分出现不同的 DFS 编码，进而无法实现 DFS 编码的正确匹配。

图 10 中的两个零件，其最小 DFS 编码如表 11 所示，球台与球轴的相似部分，由于遍历顺序的不

图 10 具有不同遍历顺序的图形

同，造成特征值的顺序不同，分别为 $(0, 1, 1, 2, 0) \rightarrow (1, 2, 2, 4, 1)$ 、 $(3, 4, 4, 2, 1) \rightarrow (4, 5, 2, 1, 0)$ ，显然编码无法进行比较和匹配。

表 11 不同遍历方向的最小 DFS 编码

边	更改前		更改后	
	球台	球轴	球台	球轴
1	(0,1,1,2,0)	(0,1,1,2,0)	(0,1,1,2,0)	(0,1,1,2,0)
2	(1,2,2,4,1)	(1,2,2,1,1)	(1,2,2,4,1)	(2,1,1,2,1)
3	(2,3,4,3,0)	(2,3,1,4,0)	(3,2,3,4,0)	(2,3,1,4,0)
4	(3,4,3,1,1)	(3,4,4,2,1)	(3,4,3,1,1)	(4,3,2,4,1)
5		(4,5,2,1,0)		(5,4,1,2,0)

为此，应对最小 DFS 编码进行转换，把所有图形的边编码按照特征值由小到大的顺序进行重新排列。比如球轴中编码 $(3, 4, 4, 2, 1)$ ，由于第一个特征值比第二个特征值大，应将这两个特征值的位置进行交换，同时各自对应的索引值也应交换位置，最后编码改为 $(4, 3, 2, 4, 1)$ 。其他边编码进行相应转换后，图 10 中两个零件的编码见表 11 所示，转换后即可挖掘公共子图。

2.2 最大公共子图的挖掘

在一个图形编码中，由于起始的边编码不同，可能挖掘出多个公共子图。

如图 11 所示，依据图形的最小 DFS 编码，以①为第 1 条边开始遍历，挖掘至顶点 4 时获得一个具有 4 条边的子图，如图 11a 所示，当添加下一条编码为 $(0, 5, 1, 2, 1)$ 的⑤时，无法匹配到具有 5 条边的子图。若以编码 $(0, 5, 1, 2, 1)$ 为第 1 条边开始遍历，可获得一个具有 1 条边的子图，如图 11b 所示，继续添加下一条编码 $(5, 6, 2, 1, 1)$ 时，无法匹配到具有 2 条边的子图。而以编码为 $(5, 6, 2, 1, 1)$ 为第 1 条边开始遍历，获得一个具有 1 条边的子图，如图 11c 所示。

再如图 12，按照最小 DFS 编码从第 1 条边编码 $(0, 1, 1, 2, 0)$ 顺序进行挖掘，可匹配到一个具有 5 条边的子图，如图 12a 所示，继续添加下一条编码 $(5, 6, 2, 2, 1)$ 时，无法匹配到具有 6 条边的子图。若以 $(5, 6, 2, 2, 1)$ 为第一条编码进行匹配，匹配不到任何子图。而以编码为 $(6, 7, 2, 1, 1)$ 为第 1 条边开始遍历，能够获得一个具有 8 条边的子图，如图 12b 所示。

(a) 球台及其最小 DFS 编码

(a) 子图有 4 条边

(b) 子图有 1 条边

(c) 子图有 1 条边

图 11 不相连的编码

由于零件图集的最小 DFS 编码已经存储在计算机中,那么最大公共子图的挖掘算法可构建如下。

步骤 1, 挖掘新零件 G_{new} 的最小 DFS 编码。

步骤 2, 依据边编码顺序原则更改编码。

步骤 3, 依据特征值和属性值, 对比新零件 G_{new} 和库中零件 G_i 的边编码, 排除只存在于一个图形中的边编码(即独立边)。

步骤 4, 以剩余边数少的图形(即源图形)为基础, 读取第一条边的边编码。

步骤 5, 在剩余边数多的零件编码(目标图形)中寻找出所有相似的边编码。

步骤 6, 判断源图形中, 当前边是否存在连接关系(属于串联还是并联关系)的下一条边。

(a) 子图有 5 条边

(b) 子图有 8 条边

图 12 相连的编码

步骤 7, 存在则添加下一条边, 然后在目标图形中, 按照相同的连接关系, 给第一条相似边(即与源图形的当前边相似)添加所有相似的下一条边编码, 直至所有相似边均添加完毕。不存在则获得与目标图形的第一个公共子图。

步骤 8, 判断当前边是否属于源图形的第一条边。

步骤 9, 属于则获得与目标图形的所有公共子图。不属于则以当前边为第一条边, 返回步骤 6。

步骤 10, 根据决策因子对库中零件进行排序。

依据上述的挖掘方法, 可在钻模库(见图 8)中挖掘图 9 中套筒的最大公共子图如表 12 所示。

由表 12 可知, 直筒与连杆、直筒、支架的最大公共子图中, 边编码数分别为 4、5、3。

2.3 决策因子

挖掘出公共子图后, 若记 N_{imax} 为新零件 G_{new} 与图集 G_{obj} 中第 i ($1 \leq i \leq n$) 个零件 G_i 之间公共子图的最大边数, 则可定义零件相似度 δ_i 和库相似度 A_i 分别为

$$\delta_i = \frac{N_{imax}}{N_{new}} \quad (1)$$

$$A_i = \frac{N_{imax}}{N_i} \quad (2)$$

式中 N_{new} ——新零件 G_{new} 最小 DFS 编码的边数;

N_i ——库中第 i 个零件 G_i 最小 DFS 编码的边数。

表 12 套筒与库中零件之间的最大公共子图

边	连杆(目标图形)	套筒(源图形)	直筒(源图形)	套筒(目标图形)	支架(目标图形)	套筒(源图形)
1	(0,1,1,0)	(0,1,1,2,0)	(0,1,1,2,0)	(0,1,1,2,0)	(0,1,1,1,0)	(0,1,1,2,0)
2	(1,2,1,1,0)	(2,1,1,2,0)	I (2,1,1,2,0)	(2,1,1,2,0)	(1,2,1,1,0)	(2,1,1,2,0)
3	(2,3,1,1,0)	I (2,3,1,2,1)	II (2,3,1,3,0)	(2,3,1,2,1)	(2,0,1,1,0)	(2,3,1,2,1)
4	(3,0,1,1,0)	II (4,3,1,2,0)	III (0,3,1,3,0)	I (4,3,1,2,0)	(2,3,1,1,0)	I (4,3,1,2,0)
5	(3,4,1,1,0)	III (4,5,1,3,0)	IV (3,4,3,3,0)	II (4,5,1,3,0)	(3,0,1,1,0)	II (4,5,1,3,0)
6	(4,5,1,2,0)	IV (0,5,1,3,0)	V (1,4,2,3,0)	III (0,5,1,3,0)	(3,4,1,1,0)	III (0,5,1,3,0)
7	(0,5,1,2,1)	(5,6,3,3,0)		IV (5,6,3,3,0)	(4,0,1,1,0)	(5,6,3,3,0)
8	(1,5,1,2,1)	(3,6,2,3,0)		V (3,6,2,3,0)	(4,1,1,1,0)	(3,6,2,3,0)
9	(2,5,1,2,1)				(4,5,1,2,0)	
10	(6,5,1,2,0)				(1,5,1,2,0)	
11	(6,7,1,3,0)				(2,5,1,2,0)	
12	(4,7,1,3,0)				I (3,5,1,2,0)	
13	(7,8,3,3,2)				II (3,6,1,3,0)	
14	III (9,8,1,3,0)				III (1,6,1,3,0)	
15	II (9,10,1,2,0)				(0,7,1,2,1)	
16	I (0,10,1,2,1)				(8,7,1,2,0)	
17	(1,10,1,2,1)					
18	(2,10,1,2,1)					
19	(3,10,1,2,1)					
20	(11,10,1,2,0)					
21	IV (11,8,1,3,0)					

注：表中罗马数字 I、II、...、V 表示边编码的挖掘顺序。

这里，定义零件相似度 δ_i 为第一决策因子，库相似度 Δ_i 为第二决策因子。按照“第一决策因子越大，夹具越相似”的原则，对夹具进行排序；若第一决策因子相同，则按照“第二决策因子越大，夹具越相似”的原则，对夹具进行排序；再若第二决策因子也相同，则夹具相似度也相同。

3 装夹性能

装夹性能的本质就是装夹布局与加工要求之间的关系，包括定位确定性和工件稳定性两个方面。

3.1 定位确定性

假定工件的装夹布局由 k 个定位点和 n 个夹紧力组成，则定位确定性指的是理论约束与工件在该装夹布局中实际约束之间的关系。若实际约束包含理论约束，装夹布局具有好的定位确定性，加工要求才能获得保证。

这里，记 $\mathbf{e}_x=[1,0,0]^T$ 、 $\mathbf{e}_y=[0,1,0]^T$ 、 $\mathbf{e}_z=[0,0,1]^T$ 分别为 X 、 Y 、 Z 的方向矢量，若工序基准上任意点的坐标为 $\mathbf{r}_p=[x_p, y_p, z_p]^T$ ，那么理论约束 $\mathbf{x}_0$ (即满足加工要求所必须限制的自由度) 为式(3)的解^[21-23]

$$\mathbf{e}^T \mathbf{E} \mathbf{x}_0 = 0 \quad (3)$$

式中 $\mathbf{E}$ ——位置矩阵，若加工要求为尺寸和定位位置，则 $\mathbf{E}=[\mathbf{I}, -\mathbf{\Omega}]$ ，若加工要求为

定向位置，则 $\mathbf{E}=[\mathbf{O}, -\mathbf{\Omega}]$ ；

$\mathbf{I}$ ——单位矩阵；

$\mathbf{O}$ ——零矩阵；

$\mathbf{\Omega}$ ——斜对称矩阵；

$$\mathbf{\Omega} = \begin{bmatrix} 0 & -z_p & y_p \\ z_p & 0 & -x_p \\ -y_p & x_p & 0 \end{bmatrix};$$

$\mathbf{e}$ ——加工要求方向。

而实际约束^[21-23]指的就是定位点布局所限制的自由度，这样，实际约束 $\mathbf{x}$ 应为式(4)的解

$$\mathbf{J} \mathbf{x} = \mathbf{0} \quad (4)$$

式中 $\mathbf{J}$ ——定位雅可比矩阵；

$$\mathbf{J} = \begin{bmatrix} \mathbf{n}_1 & \dots & \mathbf{n}_i & \dots & \mathbf{n}_k \\ \mathbf{r}_1 \times \mathbf{n}_1 & \dots & \mathbf{r}_i \times \mathbf{n}_i & \dots & \mathbf{r}_k \times \mathbf{n}_k \end{bmatrix}^T;$$

$\mathbf{n}_i$ ——工件在第 i 个定位点 $\mathbf{r}_i=[x_i, y_i, z_i]^T$ 处的单位法向量且 $\mathbf{n}_i=[n_{ix}, n_{iy}, n_{iz}]^T$ 。

为了保证工件相对于刀具的正确位置，理论约束 $\mathbf{x}_0$ 不能超过实际约束 $\mathbf{x}$ 。只有这样，才能保证工件的加工要求。因此，根据式(4)中齐次线性方程组解的存在性，可得定位确定性的理论条件^[21-23]为

$$\text{rank}(\text{diag}(\mathbf{x}_0)) + \text{rank}(\mathbf{J}) - \text{rank}(\mathbf{J} \mathbf{x}_0) = 6 \quad (5)$$

式中 rank ——求秩函数；

diag——创建对角阵函数。

另外, 为了减少夹具的制造成本和重量, 提高刀具对工件的可达性, 应使得定位方案中的定位元件越少越好, 故得定位确定性的工艺条件^[21-23]为

$$\text{rank}(\mathbf{J}) = k \quad (6)$$

3.2 装夹稳定性

装夹稳定性指的是通过施加 n 个合适的夹紧力, 抵抗切削力与切削扭矩对定位位置的破坏作用, 使其在整个加工过程中保持平衡状态^[24]。若记工件在切削过程中受到的外力包括工件重力 $\mathbf{F}_{\text{grav}}$ 、夹紧力 $\mathbf{F}_j (1 \leq j \leq n)$ 、切削力 $\mathbf{F}_{\text{cut}}$ 以及切削扭矩 $\mathbf{M}_{\text{cut}}$, 此时第 i 个 ($1 \leq i \leq k$) 定位点处的支撑反力为 $\mathbf{F}_i$ 。再记 $\mathbf{r}_i = [x_i, y_i, z_i]^T$ 表示第 i ($1 \leq i \leq k+n$) 个装夹点的位置, $\mathbf{n}_i = [n_{ix}, n_{iy}, n_{iz}]^T$ 为工件在 $\mathbf{r}_i$ 处的单位内法矢量, $\mathbf{r}_{\text{grav}} = [x_{\text{grav}}, y_{\text{grav}}, z_{\text{grav}}]^T$ 、 $\mathbf{r}_{\text{cut}} = [x_{\text{cut}}, y_{\text{cut}}, z_{\text{cut}}]^T$ 分别表示工件重心和切削力的位置。由于夹紧力属于压力, 如果忽略工件与装夹元件(即定位元件和夹紧元件)之间摩擦, 那么根据工件的静力平衡条件可得装夹稳定性模型为

$$\begin{aligned} \mathbf{G}\mathbf{y} &= \mathbf{w} \\ \text{s.t. } \mathbf{y} &\geq \mathbf{0} \end{aligned} \quad (7)$$

式中 $\mathbf{G}$ ——装夹点的方位矩阵;

$$\mathbf{G} = \begin{bmatrix} \mathbf{n}_1 & \cdots & \mathbf{n}_k & \mathbf{n}_{k+1} & \cdots & \mathbf{n}_{k+n} \\ \mathbf{r}_1 \times \mathbf{n}_1 & \cdots & \mathbf{r}_k \times \mathbf{n}_k & \mathbf{r}_{k+1} \times \mathbf{n}_{k+1} & \cdots & \mathbf{r}_{k+n} \times \mathbf{n}_{k+n} \end{bmatrix};$$

$\mathbf{y}$ ——接触力(包括定位点处的支撑反力和夹紧力)矢量, $\mathbf{y} = [y_1, \cdots, y_k, y_{k+1}, \cdots, y_{k+n}]^T$;

$\mathbf{w}$ ——外力旋量;

$$\mathbf{w} = - \begin{bmatrix} \mathbf{F}_{\text{grav}} + \mathbf{F}_{\text{cut}} \\ \mathbf{r}_{\text{grav}} \times \mathbf{F}_{\text{grav}} + \mathbf{r}_{\text{cut}} \times \mathbf{F}_{\text{cut}} + \mathbf{M}_{\text{cut}} \end{bmatrix}。$$

显然, 如果式(7)有解, 说明工件处于稳定状态, 此时可在给定的夹紧点上施加夹紧力。对于式(7)的联立线性方程组, 若矢量 $\mathbf{w}$ 中第 i 个元素 $w_i \geq 0 (1 \leq i \leq 6)$, 那么工件稳定性可根据求解下列具有收敛性的线性规划问题^[25-26]进行检验

$$\begin{aligned} \max \quad & S = g_1 y_1 + g_2 y_2 + \cdots + g_{k+1} y_{k+n} \\ \text{s.t.} \quad & \begin{cases} g_{11} y_1 + g_{12} y_2 + \cdots + g_{1(k+n)} y_{k+n} \leq w_1 \\ g_{21} y_1 + g_{22} y_2 + \cdots + g_{2(k+n)} y_{k+n} \leq w_2 \\ \vdots \\ g_{61} y_1 + g_{62} y_2 + \cdots + g_{6(k+n)} y_{k+n} \leq w_6 \\ y_1, y_2, \cdots, y_{k+n} \geq 0 \end{cases} \end{aligned} \quad (8)$$

式中 g_{ij} ——矩阵 $\mathbf{G}$ 中第 i 行第 j 列元素 ($1 \leq j \leq k+n$);

g_j ——矩阵 $\mathbf{G}$ 第 j 列元素之和, $g_j = \sum_{i=1}^6 g_{ij}$ 。

则当且仅当

$$\max(S) = \sum_{i=1}^6 w_i \quad (9)$$

方程式(7)有解。

结合最大公共子图的挖掘和装夹性能的分析方法, 可得新零件的夹具设计方法和流程如图 13 所示。为了合理地检索出新零件所能用的夹具, 应首先通过最大公共子图确定库中夹具的优先顺序, 然后逐一分析夹具的装夹性能, 直至装夹性能满足加工要求, 否则说明库中匹配不到合适的夹具。

图 13 耦合夹具设计流程

4 应用实例

为了能够在图 9 中的套筒上成功钻孔, 必须设计出一套合理的钻孔夹具, 以保证孔径 $\Phi 8_0^{+0.01}$ mm、轴线位置尺寸 (25 ± 0.2) mm 及其平行度 0.01 mm 等方面的加工要求。夹具库中存储的夹具有连杆、直筒、支架钻孔夹具, 如图 8 所示。这样, 套筒的夹

具设计过程详细如下。

(1) 挖掘套筒的最小 DFS 编码, 如表 10 所示。

(2) 挖掘套筒与连杆的最大公共子图, 如表 12 所示。

(3) 计算套筒与连杆之间的决策因子, 如表 13 所示。

表 13 套筒与库中零件的决策因子及夹具排序

决策因子	第一决策因子	第二决策因子	夹具排序
连杆	4/8	4/12	2
直筒	5/8	5/6	1
支架	3/8	3/16	3

(4) 按照决策因子的大小, 对夹具进行排序, 如表 13 所示。

(5) 依据决策因子的优先顺序, 验证夹具的装夹性能。

在直筒上钻孔时, 由重力、切削力、切削扭矩等组成的外力旋量为 $w = [-0, -2000 \text{ N}, -10 \text{ N}, -50\ 860 \text{ Nmm}, 1\ 000 \text{ Nmm}, 0]^T$ 。直筒定位后, 在夹紧表面 M 上施加均布的夹紧力(可等效为 M 中心处的集中力 F), 装夹布局如表 14 所列。这样, 套筒的装夹布局同于直筒, 如图 14 所示。

表 14 直筒的装夹布局

装夹表面	装夹元件	装夹点	坐标/mm	单位法向量
定位表面	支	1	(-46, 0, 0)	(0, 0, 1)
	承	2	(23, $-23\sqrt{3}$, 0)	(0, 0, 1)
	板	3	(23, $23\sqrt{3}$, 0)	(0, 0, 1)
孔 D	短圆	4	(0, 40, 5)	(0, 1, 0)
	柱销	5	(40, 0, 5)	(1, 0, 0)
夹紧表面	面 M	压板	6	(0, 0, -1)

由于套筒上钻孔的加工要求为 X 、 Z 向上孔中心的位置尺寸 $(25 \pm 0.2) \text{ mm}$, Z 向上孔中心与 A 面之间的平行度 0.01 mm , 而孔径则为钻头直接保证的尺寸, 故理论约束为 $x_0 = \zeta_{Rz} \lambda_{Rz}$, 其中即有 $\zeta_{Rz} = [0, 0, 0, 0, 0, 1]^T$, λ_{Rz} 为任意数。依据表 14 的各装夹点坐标及其法向量, 可得套筒的定位雅可比矩阵为

$$J = \begin{bmatrix} 0 & 0 & -1 & 0 & 46 & 0 \\ 0 & 0 & -1 & -23\sqrt{3} & -23 & 0 \\ 0 & 0 & -1 & 23\sqrt{3} & -23 & 0 \\ 0 & -1 & 0 & -5 & 0 & 0 \\ -1 & 0 & 0 & 0 & 5 & 0 \end{bmatrix}$$

故有 $\text{rank}(\text{diag}(x_0))=1$ 、 $\text{rank}(J)=5$ 、 $\text{rank}(Jx_0)=0$ 。又, 装夹布局有 5 个定位点, 即 $k=5$, 故可知挖掘出来的夹具不仅满足式(5)的理论条件, 也满足式(6)的工艺条件, 具有定位确定性。

图 14 套筒的装夹布局

此外, 装夹点的方位矩阵为

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 \\ 0 & 39.837\ 2 & -39.837\ 2 & 5 & 0 & 0 \\ -46 & 23 & 23 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

由式(8)可得 $\max(S) = 53\ 870$, 故存在 $\max(S) = \sum_{i=1}^6 w_i$, 因此检索出来的夹具亦有装夹稳定性。

(6) 综合最大公共子图和装夹性能分析可知, 直筒的夹具可用于套筒的装夹, 设计过程结束。

5 结论

(1) 根据零件的 B-rep 表示法, 通过特征的几何拓扑关系建立了零件的标号图。依据遍历标号图的线性顺序性, 提出了标号图的 DFS 词典序原则, 建立了最小 DFS 编码的挖掘方法。

(2) 依据边编码中特征值、属性值、特征之间的连接方式, 提出了通过最小 DFS 编码挖掘零件之间最大公共子图的有效方法。通过定义零件相似度和图集相似度, 提出了夹具相似性的优先排序方法。

(3) 最小 DFS 编码的本质是零件的几何拓扑结构, 合理有效的挖掘算法是实现夹具准确匹配的关键所在。匹配成功后, 再通过装夹性能评估方法依次检验其对新零件的装夹合理性, 最终达到实现夹具设计的目的。该方法为实用的计算机辅助夹具设计系统研发提供了基础理论。

参 考 文 献

- [1] 路冬, 李剑峰, 孙杰, 等. 航空框类零件加工动态夹紧力确定有限元分析[J]. 山东大学学报, 2007, 37(1): 19-22.
LU Dong, LI Jianfeng, SUN Jie, et al. Finite element

- analysis for the verification of the dynamic clamping force during machining of the aero frame shape workpiece [J]. Journal of Shandong University, 2007, 37(1): 19-22.
- [2] 秦国华, 张卫红, 万敏. 基于线性规划的工件稳定性建模及其应用[J]. 机械工程学报, 2005, 41(9): 33-37, 41.
- QIN Guohua, ZHANG Weihong, WAN Min. Modeling and application of workpiece stability based on the linear programming method[J]. Chinese Journal of Mechanical Engineering, 2005, 41(9): 33-37, 41.
- [3] HSIAO S W. Concurrent design method for developing a new product[J]. International Journal of Industrial Ergonomics, 2002, 29(1): 41-55.
- [4] 朱耀祥, 融亦鸣, 朱剑, 等. 计算机辅助组合夹具设计系统的研究[J]. 机械工程学报, 1994, 30(5): 40-46.
- ZHU Yaoxiang, RONG Yiming, ZHU Jian, et al. Study on computer aided modular fixture design system[J]. Chinese Journal of Mechanical Engineering, 1994, 30(5): 40-46.
- [5] 刘永喜, 乔立红. 基于成组技术的夹具快速配置研究[J]. 成组技术与生产现代化, 2007, 24(1): 29-32.
- LU Yongxi, QIAO Lihong. Study of fixture rapid configuration based on group technology[J]. Group Technology & Production Modernization, 2007, 24(1): 29-32.
- [6] 张胜文, 李霞, 吴春桥. 基于范例的推理在管件焊接柔性夹具设计系统中的应用研究[J]. 中国机械工程, 2011, 22(9): 1071-1075.
- ZHANG Shengwen, LI Xia, WU Chunqiao. Application and study of CBR in tube welding flexible fixture design [J]. China Mechanical Engineering, 2011, 22(9): 1071-1075.
- [7] 罗晨, 王欣, 苏春, 等. 基于案例推理的夹具设计案例表示与检索[J]. 机械工程学报, 2015, 51(7): 136-143.
- LUO Chen, WANG Xin, SU Chun, et al. Case representation and retrieval under a case-based reasoning system for fixture design[J]. Chinese Journal of Mechanical Engineering, 2015, 51(7): 136-143.
- [8] NEE A Y C, KUMAR A S. A framework for an object/rule - based automated fixture design system[J]. Annals of CIRP, 1991, 9(5): 3487-3497.
- [9] LIU W J, HUANG Y Q, CHANG W. A study on modular fixture design with intelligent selection of elements and interactive assembly[J]. Journal of Harbin Institute of Technology, 1996, E3(4): 45-48.
- [10] 万能, 苟园捷, 莫蓉. 基于智能元件的柔性夹具设计新方法(1)[J]. 计算机集成制造系统, 2012, 18(11): 2354-2361.
- WAN Neng, GOU Yuanjie, MO Rong. New method for flexible fixture design based on smart fixture unit (1)[J]. Computer Integrated Manufacturing System, 2012, 18(11): 2354-2361.
- [11] 万能, 苟园捷, 莫蓉. 基于智能元件的柔性夹具设计新方法(2)[J]. 计算机集成制造系统, 2012, 18(11): 2362-2369.
- WAN Neng, GOU Yuanjie, MO Rong. New method for flexible fixture design based on smart fixture unit (2)[J]. Computer Integrated Manufacturing System, 2012, 18(11): 2362-2369.
- [12] 高博, 阎艳, 丁洪生, 等. 基于知识组件的夹具元件快速设计研究[J]. 北京理工大学学报, 2013, 33(10): 1021-1026.
- GAO Bo, YAN Yan, DING Hongsheng, et al. Research on rapid design of fixture parts based on knowledge components[J]. Transactions of Beijing Institute of Technology, 2013, 33(10): 1021-1026.
- [13] 朱耀祥, 融亦鸣. 柔性夹具与计算机辅助夹具设计技术的进展[J]. 制造技术与机床, 2000(8): 5-8.
- ZHU Yaoxiang, RONG Yiming. Development of flexible fixture and computer aided fixture design technology[J]. Manufacturing Technology & Machine Tool, 2000(8): 5-8.
- [14] WANG Q, LIAO W H. Application and development of case-based reasoning in fixture design[J]. Journal of Southeast University, 2002, 20(2): 170-175.
- [15] 刘金山, 廖文和, 张素敏, 等. 基于零件特征关注度的夹具实例相似性检索方法[J]. 计算机辅助设计与图形学学报, 2007, 19(10): 1303-1307, 1314.
- LIU Jinshan, LIAO Wenhe, ZHANG Sumin, et al. Research on feature attention-driven fixture case retrieval [J]. Journal of Computer-Aided Design & Computer Graphics, 2007, 19(10): 1303-1307, 1314.
- [16] CICIRELLO V A, REGLI W C. An approach to a feature-based comparison of solid models of machined parts[J]. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 2002, 16(5): 385-399.
- [17] ELINSON A, NAU D S, REGLI W C. Feature-based similarity assessment of solid models [C]. Proceedings of the Fourth ACM/SIGGRAPH Symposium on Solid Modeling & Applications, Atlanta, GA, 1997: 297-310.
- [18] 黎荣, 王金诺. 基于 UG 的 3D 模型相似性评价技术研究[J]. 计算机工程与应用, 2006, 42(36): 86-88, 175.
- LI Rong, WANG Jinnuo. Study of 3D models' similarity assessment based on UG[J]. Computer Engineering and Applications, 2006, 42(36): 86-88, 175.
- [19] 张欣, 莫蓉, 宫中伟, 等. CAD 模型自动语义标注[J]. 计算机辅助设计与图形学学报, 2010, 22(12): 2162-2167.
- ZHANG Xin, MO Rong, GONG Zhongwei, et al. Automatic semantic tagging of cad models for retrieval[J].

- Journal of Computer-Aided Design & Computer Graphics, 2010, 22(12): 2162-2167.
- [20] 李先通, 李建中, 高宏. 一种高效频繁子图挖掘算法[J]. 软件学报, 2007, 18(10): 2469-2480.
- LI Xiantong, LI Jianzhong, GAO Hong. An efficient frequent subgraph mining algorithm[J]. Journal of Software, 2007, 18(10): 2469-2480.
- [21] QIN G H, ZHANG W H, WAN M. A machining dimension-based approach to locating scheme design[J]. ASME Journal of Manufacturing Science and Engineering, 2008, 130(5): 0510101-0510108.
- [22] QIN G H, YE H C, RONG Y M. A unified point-by-point planning algorithm of machining fixture layout for complex workpiece[J]. International Journal of Production Research, 2014, 52(5): 1351-1362.
- [23] 秦国华, 吴志斌, 叶海潮, 等. 基于层次分析法与定位确定性的工件定位方案规划算法[J]. 机械工程学报, 2016, 52(1): 193-203.
- QIN Guohua, WU Zhibin, YE Haichao, et al. Design algorithm of workpiece locating scheme based on analytical hierarchy process and locating determination[J]. Chinese Journal of Mechanical Engineering, 2016, 52(1): 193-203.
- [24] 秦国华, 孙烁, 王华敏, 等. 基于工件稳定性的全区域夹紧力变向迭代规划算法[J]. 兵工学报, 2016, 37(9): 1700-1707.
- QIN Guohua, SUN Shuo, WANG Huamin, et al. A reverse direction iterative planning algorithm of clamping forces in entire active region based on workpiece stability[J]. Acta Armamentarii, 2016, 37(9): 1700-1707.
- [25] 秦国华, 王华敏, 叶海潮, 等. 基于力的存在性与可行性的夹紧力变向增量递减规划算法[J]. 机械工程学报, 2016, 52(11): 72-79.
- QIN Guohua, WANG Huamin, YE Haichao, et al. Planning algorithm of clamping forces for workpiece fixturing scheme based on existence and feasibility of forces[J]. Journal of Mechanical Engineering, 2016, 52(11): 72-79.
- [26] 秦国华, 王子琨, 吴竹溪, 等. 基于表面网格离散化与遗传算法的复杂工件装夹布局规划方法[J]. 机械工程学报, 2016, 52(13): 195-203.
- QIN Guohua, WANG Zikun, WU Zhuxi, et al. A planning method of fixturing layout for complex workpieces based on surface discretization and genetic algorithm[J]. Journal of Mechanical Engineering, 2016, 52(13): 195-203.

作者简介: 秦国华, 男, 1970 年出生, 博士, 教授, 博士后。主要研究方向为工件装夹分析与优化、加工过程力学分析与仿真、残余应力分析与预测、健康管理及故障诊断等。

E-mail: qghwzx@126.com

王华敏(通信作者), 女, 1988 年出生, 博士研究生。主要研究方向为复杂工件装夹布局分析与规划、切削加工变形预测与力学分析等。

E-mail: 910291556@qq.com