

参数化模型欠、过和完整约束的判定算法*

蒋 鲲^{1,2+}, 高小山², 岳晶岩^{2,3}

¹(黑龙江大学 理学院,黑龙江 哈尔滨 150080)

²(中国科学院 数学机械化重点实验室,北京 100080)

³(吉林大学 数学所,吉林 长春 130012)

A Decision Method for Under-, Over- and Well-Constrainedness of Parametric Model

JIANG Kun^{1,2+}, GAO Xiao-Shan², YUE Jing-Yan^{2,3}

¹(College of Sciences, Heilongjiang University, Harbin 150080, China)

²(Key Laboratory of Mathematics Mechanization, The Chinese Academy of Sciences, Beijing 100080, China)

³(Institute of Mathematics, Jilin University, Changchun 130012, China)

+ Corresponding author: Phn: 86-451-86608282, E-mail: kjiang@etang.com

<http://www.mmrc.iss.ac.cn/~kjiang>

Received 2002-11-04; Accepted 2003-01-28

Jiang K, Gao XS, Yue JY. A decision method for under-, over- and well-constrainedness of parametric model. *Journal of Software*, 2003,14(12):2092~2097.

<http://www.jos.org.cn/1000-9825/14/2092.htm>

Abstract: In parametric CAD design, the designers often encounter the question of judging whether a parametric model is under-, over- or well-constrained. In this paper, a graph-based algorithm for the question is proposed. This algorithm gives not only the decision of under-, over- or well-constrainedness of a parametric model, but also the exact location where under-, over- or well-constraint occurs in the parametric model. This feature provides a more convenient way to the designers in the process of designing.

Key words: parametric CAD; geometric constraint solving; bigraph; DM-decomposition; maximum match

摘 要: 在参数化 CAD 设计中,设计者常常遇到判断一个参数化模型是欠、过和完整约束的问题.针对这个问题,提出了一个判断参数化模型的欠、过和完整约束性的图论算法.该算法不仅能够给出判断,同时还能够对欠和过约束的情形进行定位,即能够给出欠和过约束发生的具体位置.这给设计者在设计过程中提供了很大的方便.

关键词: 参数化 CAD;几何约束求解;偶图;DM 分解;最大匹配

中图法分类号: TP391 文献标识码: A

参数化 CAD 是 CAD 发展史上的第三次技术创新.它是在 1988 年由参数化技术公司(Parametric Technology

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1998030600 (国家重点基础研究发展规划(973)); the Youth Foundation of Tianyuan Mathematics of China under Grant No.10226006 (数学天元青年基金); the President Foundation of the Chinese Academy of Sciences (中国科学院院长基金)

第一作者简介: 蒋鲲(1972—),男,黑龙江哈尔滨人,博士,主要研究领域为几何约束求解,智能 CAD.

Company,简称 PTC)提出的一种崭新的设计理念,是一种基于特征的设计,具有全尺寸约束、全数据相关和尺寸驱动设计修改的特点^[1].基于这个理念,参数化技术公司研制了一个名为 Pro/ENGINEER 的参数化软件.到了 20 世纪 90 年代,参数化设计技术已经变得成熟起来,充分体现出其在许多通用软件、零部件设计上存在着简便、易行的优势.设计者在设计的初始阶段,通常是在草图上不断地添加自己的构想,实际上就是添加一些相关的约束.这往往造成设计的草图不总是完整约束的,而是由欠约束到过约束的设计过程.例如,一个四边形的设计过程可能先是图 1 中的(b),然后是(a),最后可能是(c).设计者在设计的过程中由于专注于自己的构想通常会忽略这个问题,这可能会给设计的修改和实现带来许多麻烦,因此,一个能够判断几何实体的参数化设计模型的欠、过或者完整约束的算法就显得尤为重要.因为有了这个算法以后,就可以实时地提示设计者.使设计者在设计的过程中可以交互地修改草图.实际上,在参数化设计的各个阶段常常要碰到一个问题,就是需要判断一个参数化模型是欠、过还是完整约束的.在许多参数化设计系统的关键算法中也常常碰到这样的判断.文献[3]给出了一个判断参数化模型是否为过约束的符号计算方法.文献[4]中给出的判定算法需要人为地在约束中加上一些权重,而本文提出的判定算法则不用,更有直观的意义.文献[5]提出的判定算法只能对自由度和约束度为 1 的几何实体和几何约束进行判定.本文提出的算法可以处理任意自由度的几何实体.文献[6]也给出了一个判定欠和过约束非常有效的算法,但只是对一些特定约束的判定,而不是所有的约束.文献[7]中给出了一种判定欠和过约束的数值方法.本文提出的算法可以对包括代数形式的约束在内的所有的约束进行判定.本文提出的判断参数化模型的欠、过和完整约束性的算法基于偶图的 DM 分解.这个算法不仅能够给出判断,同时还能够对欠和过约束的情形进行定位,即能够给出欠和过约束发生的具体位置.这给设计者在设计过程中提供了很大的方便.该算法已经由中国科学院数学机械化重点实验室研制的软件 MMP/Geometer^[10]所采用.

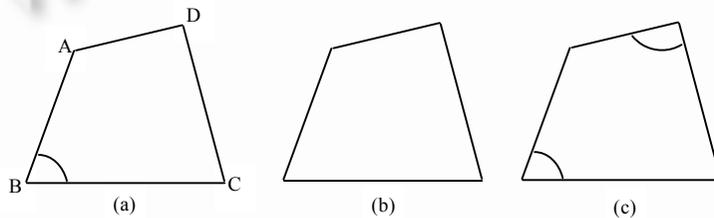


Fig.1 Structural well-, under- and over-constrained parametric design model of geometric object

图 1 结构完整、欠和过约束的几何实体的参数化模型

1 基本概念

本节首先解释在本文中用到的几个基本概念.

1.1 基本概念

基本体素是指一个几何图形中最基本、最具有特征的几何元素,例如,二维中的点、直线、圆和圆弧,三维中的平面、球面和曲面等.

基本体素的自由度是指确定这个基本体素所需要的独立参数的个数,一般用 DOF 来表示.

几何约束是指两个或多个基本体素之间所具有的几何关系,例如,点与点之间的距离、直线与直线之间的角度等等.

几何约束的约束度是指表示这个几何约束所需要的标量方程的个数,一般用 DOC 来表示.

几何实体是指由基本体素按照一定的几何约束关系构成的几何图形.

几何实体的参数化模型是指包括组成这个几何实体的基本体素和几何约束关系以及构造这个几何实体的具体方案在内的一个构造模型.

结构完整(过、欠)约束的参数化模型:对于一个给定的参数化模型 G ,如果满足 $DOC(G)=DOF(G)-3$ (三维是 6),并且对于任何一个子模型 H ,满足 $DOC(H)\leq DOF(H)-3$ (三维是 6),则称这个参数化模型是结构完整约束的;如果存在一个子模型 H 满足, $DOC(H)>DOF(H)-3$ (三维是 6),则称这个参数化模型是结构过约束的;如果 G 不是结

构过约束的,并且满足 $DOC(G) < DOF(G) - 3$ (三维是 6), 则称这个参数化模型是结构欠约束的, 如图 1 所示(已知四边形的 4 个边长, 被标记的角是已知的).

在大部分情形, 结构完整(欠、过)约束与几何完整(欠、过)约束是一致的. 本文主要讨论结构完整(欠、过)约束的判定问题. 参数化设计模型是通过修改定义一个几何实体模型的几何约束来生成一个新的几何实体. 在参数化模型中, 构造几何实体的具体方案中所涉及到的几何约束关系都没有赋予具体的值, 因此我们将这个几何约束关系称为参数, 即这个参数的值是随时可以改变的. 这也是我们为什么将这个模型称为参数化模型的一个原因. 当设计者给这一组参数赋予确定值时, 这个参数化模型就自动生成一个满足这组给定的参数值的几何实体的实例. 这个自动生成的过程被称为参数化模型的实例化. 用于自动生成的方法被称为几何约束求解^[2-8].

1.2 偶图和最大匹配

本文是利用图论中偶图及其 DM 分解来对参数化设计模型的欠、过和完整约束的性质进行判断的, 因此下面我们将给出偶图及其 DM 分解^[9]的一些相关概念.

一个偶图是一个有向图, 记为 $B=(V^+, V^-, E)$. 其中, 结点集合由两个互不相交的子集 V^+ 和 V^- 组成, 边的集合 E 由所有从集合 V^+ 中的结点指向集合 V^- 中结点的边组成. 偶图 B 的一个匹配 M 是集合 E 的一个子集. 其中子集 M 中的任意两条边都不能有相同的始结点. 在一个匹配中所含边的个数最多的时候, 我们称这个匹配是最大匹配. 一个偶图 B 的最大匹配不是惟一的. 我们称图 $G_M=(V^+ \cup V^-, \hat{E}; S^+, S^-)$ 是偶图 $B=(V^+, V^-, E)$ 关于最大匹配的辅助图, 其中 \hat{E} 满足: $(v, w) \in \hat{E} \Leftrightarrow (v, w) \in E$ 或者 $(w, v) \in M, S^+ = V^+ \setminus \delta^+ M$ 和 $S^- = V^- \setminus \delta^- M$ 被分别称为入口和出口. 利用最大匹配的概念, 我们可以将偶图惟一分解成一些不可分的偶子图 $B_i=(V_i^+, V_i^-, E_i)$. 由于这个分解是由 Dulmage-Mendelsohn 首先提出来的, 所以我们将这个分解称为 DM-分解, 将那些不可分的偶子图 $B_i=(V_i^+, V_i^-, E_i)$ 称为 DM-偶子图. 在这些 DM-偶子图中有两个比较特殊, 其中一个 DM-偶子图所包含的 V^+ 中的结点的个数比 V^- 中的结点的个数多, 我们称这个 DM-偶子图为最小不一致偶子图, 记为 V_0 ; 另外一个 DM-偶子图所包含的 V^+ 中的结点的个数比 V^- 中的结点的个数要少, 我们称这个 DM-偶子图为最大不一致偶子图, 记为 V_∞ . 图 2 给出了 DM-分解的一个例子; 其他 DM-偶子图所包含的 V^+ 中的结点的个数与 V^- 中的结点的个数一样多. 偶图的分解不依赖于最大匹配的选择^[9].

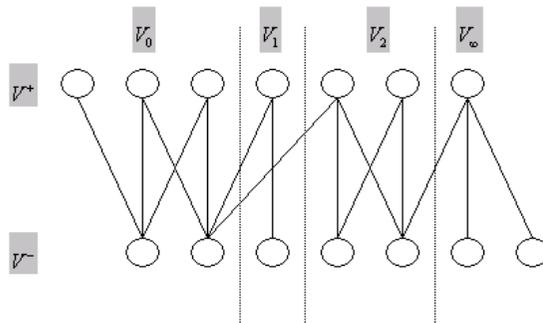


Fig.2 Bigraph and its DM-decomposition

图 2 偶图及其 DM 分解

2 算法

我们首先将参数化设计模型用偶图表示出来. 具体作法如下: 我们用偶图中的 V^+ 表示参数化设计模型中的基本体素. 用 V^- 表示参数化设计模型中的几何约束. 如果一个几何约束与某几个基本体素相关, 则在这个几何约束和与这个几何约束相关联的基本体素之间用一条边相连接, 这样我们就得到一个偶图. 我们称这个偶图为参数化设计模型的表示偶图. 利用偶图的 DM 分解可以得到这个表示偶图的一个分解. 在由分解所得到的最小不一致偶子图 V_0 中, 由于所包含的 V^+ 中结点的个数比 V^- 中结点的个数多, 即表明所对应的基本体素的自由度比几何约束的约束度多, 因此最小不一致偶子图 V_0 对应的是参数化设计模型中的欠约束部分. 同理, 偶图分解中的最

大不一致偶子图 V_{∞} 对应的是参数化设计模型中的过约束的部分.如果 V_0 和 V_{∞} 都是空集,则说明参数化设计模型中所包含的基本体素的自由度等于所包含的几何约束的约束度,即参数化设计模型是完整约束的.通过上机实际编程并进行计算所得到的实验结果与理论分析的结论是一致的.该算法已经由中国科学院数学机械化重点实验室的软件 MMP/Geometer 所实现.通过这个结论,我们可以得到参数化设计模型的欠、过和完整约束的一个判断算法.并且在模型是欠和过约束的情形下,我们还可以通过 V_0 和 V_{∞} 中包含的基本体素和几何约束的具体信息来确定欠和过约束发生的具体位置,这为设计者在交互方式下进行设计提供了很大的方便和修改的理论依据.下面我们就给出这个算法的具体描述.

为了描述算法方便,我们用 $\{(g_1, g_2, \dots, g_m), (c_1, c_2, \dots, c_n)\}$ 来表示一个几何实体 O 的参数化模型 PM , 其中 g_i 是指几何实体中所包含的基本体素, c_i 是指几何实体中的几何约束关系.

算法. 参数化模型完整约束的判断算法.

输入: 一个几何实体的参数化模型 PM .

输出: 一个几何实体的完整约束的参数化模型 WPM .

第 1 步: 给定一个参数化模型 $PM = \{(g_1, g_2, \dots, g_m), (c_1, c_2, \dots, c_n)\}$, 对于每一个基本体素 g_i , 我们用 V^+ 中的结点 $V_{i1}^+ \dots V_{ip}^+$ 来表示, 其中 p 等于基本体素 g_i 的自由度的个数(确定该基本体素所需的自由变量的个数). 对于每一个几何约束 c_j , 我们用 V^- 中的结点 $V_{j1}^- \dots V_{jq}^-$ 来表示, 其中 q 等于几何约束 c_j 的约束度的个数(表示这个几何约束所需方程的个数).

第 2 步: 由于 DM 分解要求参数化模型中所包含的基本体素的自由度的总和等于所有几何约束的约束度总和, 所以我们另外在 V^- 中加入 3 个结点 $V_{\text{ground}1}^-, V_{\text{ground}2}^-, V_{\text{ground}3}^-$ 来表示固定几何实体的绝对位置的 3 个基础约束. 如果 c_j 与 g_i 有关, 则我们在结点 $V_{i1}^+ \dots V_{ip}^+$ 与结点 $V_{j1}^- \dots V_{jq}^-$ 之间都加上一条边 $E_{ik, jl} (k=1, \dots, p, l=1, \dots, q)$. 最后再任意选取一个基本体素 g_i , 将 $V_{\text{ground}1}^-$ 与 V^+ 中表示 g_i 的节点 V_{i1}^+ 相连接, 将 $V_{\text{ground}2}^-$ 与 V_{i2}^+ 相连接, $V_{\text{ground}3}^-$ 与 V_{i3}^+ 相连接, 如果 $p < 3$, 则再任意选取另外一个基本体素 g_j , 依此类推, 将 $V_{\text{ground}i}^-$ 中未被连接的节点与 V^+ 中表示 g_j 的节点相连接, 最终可以将 $V_{\text{ground}i}^-$ 与 V^+ 中的 3 个节点相连接. 这样我们就得到几何实体参数化模型 PM 的一个偶图表示. 我们将这个偶图称为给定的参数化模型 PM 的表示偶图, 记为 $G(PM)$.

第 3 步: 利用偶图的 DM 分解^[9], 对 $G(PM)$ 进行分解. 通过这个分解, 我们得到 $G(PM)$ 中的一些子图: $G_0, G_1, G_2, \dots, G_p, G_{\infty}$.

第 4 步: 如果 $G_0 \neq \emptyset$, 说明给定的参数化模型是欠约束的. 于是就将 G_0 中所包含的基本体素和几何约束输出给设计者, 提示给设计者欠约束发生的位置. 转到第 6 步; 否则进行下一步.

第 5 步: 如果 $G_{\infty} \neq \emptyset$, 说明给定的参数化模型是过约束的. 于是就将 G_{∞} 中所包含的基本体素和几何约束输出给设计者, 提示给设计者过约束发生的位置. 转到第 6 步; 否则进行下一步.

第 6 步: 报告设计者, 这个参数化模型是完整约束的.

第 7 步: 结束.

上述算法只是针对二维情形来描述的. 为了可以对三维的参数化设计模型进行判断, 我们只需对上述算法稍作改动, 将 $V_{\text{ground}i}^- (i=1 \dots 3)$ 改成 $V_{\text{ground}i}^- (i=1 \dots 6)$, 将判断 $p < 3$ 改成 $p < 6$ 即可. 本算法的复杂度是 $O((V^+ + V^-)^3)$.

3 计算实例

最后, 我们给出两个实际计算的例子来说明本文提出的算法的有效性和正确性.

例 1: 图 3 给出一个四边形的参数化设计模型, 其中基本体素包括: 点 A, B, C, D , 直线 L_{AB} 和 L_{BC} . 几何约束包括: d_{AB} 表示点 A 到点 B 的距离. 类似地有 $d_{BC}, d_{CD}, d_{AD}, d_{AC}$. a 表示直线 L_{AB} 和 L_{BC} 之间的角度. $On(A, L_{AB})$ 表示点 A 在直线 L_{AB} 上. 类似地有 $On(B, L_{AB}), On(B, L_{BC}), On(C, L_{BC})$. 图 4 给出这个参数化设计模型的表示偶图. 在表示偶图中, 为了标记方便, 我们将 $On(A, L_{AB})$ 和 $On(B, L_{AB}), On(B, L_{BC}), On(C, L_{BC})$ 都简记为 On , 用 B_{s1}, B_{s2}, B_{s3} 表示 $V_{\text{ground}i}^- (i=1, \dots, 3)$, 这里我们选点 A 和直线 L_{AB} 的方向作为基础约束.

通过算法的计算, 输出的结果是, $G_{\infty} = \{A, B, C, a, d_{AB}, d_{BC}, d_{AC}\} \neq \emptyset, G_0 = \emptyset$. 说明此参数化模型是过约束的, 且过约束就发生在由 A, B, C, L_{AB}, L_{BC} 构成的三角形 ABC 中, 这可以从图 3 中明显看出来. 于是, 若想将此参数化模型成

为完整约束的,只需将 G_{∞} 中所包含的几何约束去掉一个即可,例如可以去掉 d_{AC} .

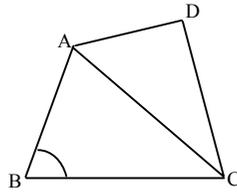


Fig.3 A parametric design model of quadrilateral, length of four sides, length of a diagonal and an angle is given
图3 一个四边形的参数化设计模型,已知4个边长和1个对角形长以及1个角度

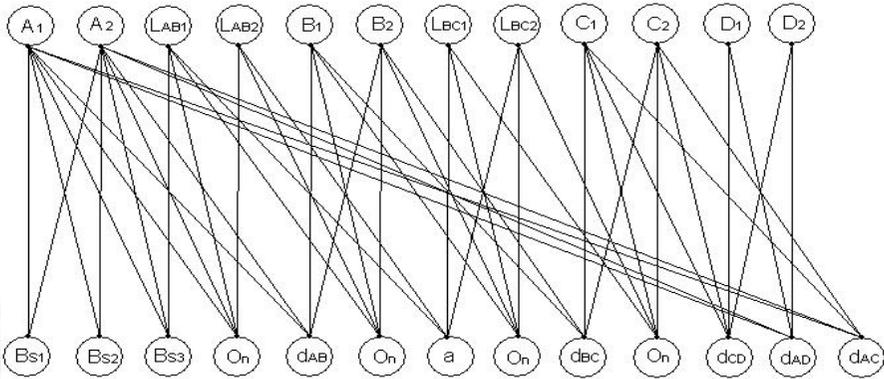


Fig.4 Representation bigraph of parametric design model in Fig.3
图4 图3中的参数化设计模型的代表偶图

例2:这是一个机床的参数化设计模型,如图5所示.其中基本体素包括:点 $p_i, i=1, \dots, 17$, 水平直线 $v_i, i=1, \dots, 6$, 垂直直线 $h_i, i=1, \dots, 7$, 斜直线 $l_i, i=1, \dots, 4$. 几何约束包括:距离约束 $dis(v_1, v_5), dis(v_2, v_5), dis(v_2, v_3), dis(v_4, v_5), dis(v_5, v_6), dis(h_1, h_2), dis(h_2, h_3), dis(h_1, h_5), dis(l_1, l_4), dis(p_{12}, h_4), dis(p_{12}, h_1), dis(p_{16}, p_{17}), dis(h_4, h_6)$, 角度约束 $angle(l_4, h_5), angle(l_1, l_2)$. 在这里,为了方便起见,我们就不列出拓扑约束,即点在直线上的约束,这可以从图中很容易地看出来.

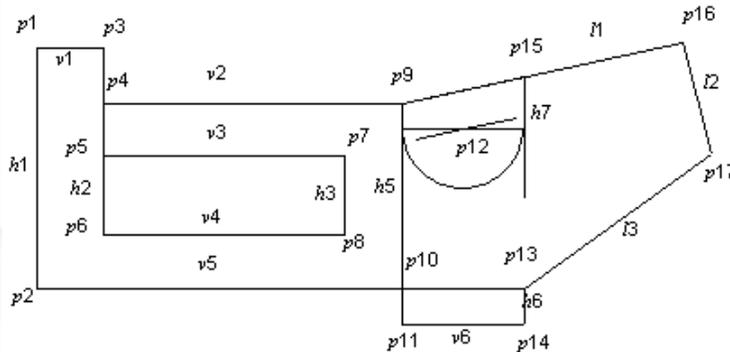


Fig.5 A parametric design model of lathe
图5 一个机床的参数化设计模型

这个参数化设计模型的代表偶图可以仿照例1给出,这里不再给出.通过计算,输出的结果是 $G_0 = \{p_{16}, p_{17}, l_2\} \neq \emptyset, G_{\infty} = \emptyset$. 这说明此参数化模型是欠约束的,且欠约束就发生在由 p_{17}, p_{16}, l_2 组成的直线段上.从给出的几何约束中我们可以看出,由于没有给定点 p_{15} 到 p_{16} 之间的距离,所以导致点 p_{16} 和 p_{17} 这个直线段是无法确定的,我们只需加上一个约束 $dis(p_{15}, p_{16})$ 就可以使原来的参数化设计模型变成完整约束的.

4 结 论

在参数化 CAD 设计中,设计者常常遇到判断一个参数化模型是欠、过和完整约束的问题.针对这个问题,本文提出了一个判断参数化模型的欠、过和完整约束性的图论算法.这个算法不仅能够给出判断,同时还可以对欠和过约束的情形进行定位,即能够给出欠和过约束发生的具体位置.这给设计者在设计过程中提供了很大的方便.在参数化设计模型发生过约束的情形下,有两种情况:一种是一致性(非矛盾)的过约束,即此参数化设计模型仍然可以作出有限个几何实体;另一种情况是不一致性(矛盾)的过约束,即真正意义上的过约束,由此参数化设计模型作不出任何几何实体.本文提出的判断算法还无法判断参数化设计模型在过约束的情况下,究竟是哪一种过约束.这一问题的完全解决还需要借助更强有力且计算复杂度更高的符号计算理论^[3].

References:

- [1] Sunde G. Specification of shape by dimension and other geometric constraints. In: Wozny MJ, McLaughlin HW, Encarnacao JL, eds. Geometric Modeling for CAD Applications. Amsterdam, 1988. 199~213.
- [2] Kramer G. Solving Geometric Constraint Systems. New York: MIT Press, 1992.
- [3] Gao XS, Chou SC. Solving geometric constraint systems II: A symbolic approach and decision of Re-constructibility. Journal of Computer Aided Design, 1998,30(2):115~122.
- [4] Latham RS, Middleditch AE. Connectivity analysis: A tool for processing geometric constraints. Journal of Computer Aided Design, 1996,28(11):917~928.
- [5] Serrano D. Automatic dimensioning in design for manufacturing. In: Rossignac J, Turner J, eds. Proceedings of the Symposium Solid Modelling Foundations and CAD/CAM Applications. New York: ACM Press, 1991. 379~386.
- [6] Owen JC. Algebraic solutions for geometry from dimensional constraints. In: Rossignac J, Turner J, eds. Proceedings of the Symposium Solid Modelling Foundations and CAD/CAM Applications. New York: ACM Press, 1991. 397~407.
- [7] Peng XB, Chen LP, Zhou FL, Zhou J. Singularity analysis of geometric constraint system. Journal of Computer Science and Technology, 2002,17(3):314~323.
- [8] Bouma W, Fudos I, Hoffmann C, Cai J, Paige R. A geometric constraint solver. Journal of Computer Aided Design, 1995, 27(6):487~501.
- [9] Kazuo Murota. Algorithms and Combinatorics 3: Systems Analysis by Graphs and Matroids, Structural Solvability and Controllability. Berlin: Springer-Verlag, 1987.
- [10] <http://www.mmrc.iss.ac.cn/~mmssoft>. 2002.