中国农业大学学报 2000, 5(2): 100～103
Journal of China Agricultural University

二维图形参数化设计中几何约束模型改进的无向图存储结构

陈忠良[①] **梅树立**
(中国农业大学计算机网络中心)

摘　要　在充分分析现有无向图存储结构优缺点的基础上, 结合二维工程图形的特点, 提出了一种改进的无向图存储结构——分类邻接表存储结构。采用该存储结构可降低程序的时间复杂度, 提高求解效率。

关键词　无向图; 几何约束模型; 分类邻接表
分类号　TP 391.72

An Improved Structure of Undigraph for Geometrical Constrain Model in 2D Parametric Design

Chen Zhongliang　Mei Shuli
(Computer Network Center, CAU)

Abstract　In 2D parametric drawing, undigraph is always used for describing geometrical constraint model. Based on analyzing existed structures of undigraph and characteristics of 2D drawing, Classified Adjacency List which is an improving structure of undigraph is presented. Time Complexity could be reduced and efficiency could be improved if this structure of undigraph is used in algorithm.

Key words　undigraph; geometrical constrain model; Classified Adjacency List

大多数二维工程图形都是由点、直线、圆弧、圆等基本图形元素构成, 这些基本元素之间存在着相切、共线、平行、垂直、共点等复杂的结构约束关系以及距离、角度、直径等尺寸约束关系。图形元素之间既不存在前后顺序关系也不存在明显的层次关系, 任何 2 个图形元素之间都有可能通过某种约束建立起联系, 因此, 用一个无向图存储结构可以很好地描述一个几何约束模型。常用的无向图存储结构有邻接表、邻接多重表和十字链表。笔者在充分研究已有存储结构的基础上, 结合二维工程图形的特点, 提出了一种改进的无向图存储结构——分类邻接表存储结构。采用该存储结构可降低程序的时间复杂度, 提高求解效率。

1　几何约束系统的形式化定义

几何约束系统用无向图结构来表达, 可用式(1)来描述[1]

$$G_{cg}=(G, C) \tag{1}$$

其中: G_{cg} 表示几何约束图; $G=\{g \mid g$ 为组成 G_{cg} 的几何实体$\}$; C 为几何约束集, 描述几何实体 g_i 和 g_j 间的几何约束关系, 如距离、平行、相切等, $C=\{(g_i, g_j) \mid R(g_i, g_j) \cap (g_i, g_j \in G)\}$。

收稿日期: 1999-11-15
①陈忠良, 北京清华东路 17 号 中国农业大学(东校区) 53 信箱, 100083

在以后的 G_{cg} 图中, 规定: 顶点(用圆圈表示) 表示几何实体, 圆圈中的符号 P, L, A, C 分别表示点实体、线实体、圆弧实体和圆实体; 几何约束图中的边表示 2 个顶点之间的一组约束, 为了表达清晰, 在边的旁边用英文缩写来表达 2 个顶点之间的约束关系, 各英文缩写所表示的约束关系如下:

a－tan－a　　// 圆弧和圆弧相切
startpts－con－startpts　　// 起点和起点共点
1－tan－c　　// 直线切于圆
1－parallel－1　　// 直线和直线平行
1－upright－1　　// 直线和直线垂直
1－cross－1　　// 直线和直线相交

图 1 所示为矩形的 G_{cg} 图。用图的结构来表达一个几何约束系统, 不但直观清晰, 而且图的有关算法稳定性好、效率高, 应用十分广泛[2]。

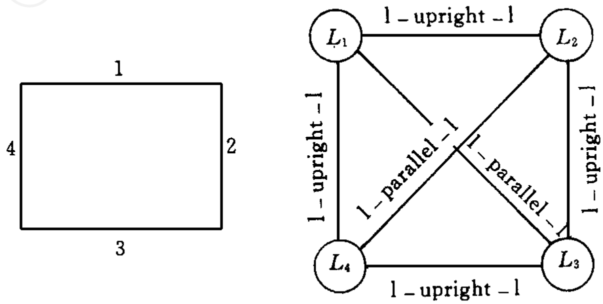


图 1　矩形及其 G_{cg} 图

2　G_{cg} 的存储结构

由于图的结构比较复杂, 任意 2 个顶点之间都可能存在联系; 同时, 图中各个顶点的度数各不相同, 最大个数和最小个数可能相差很多, 所以图的存储结构不能以简单的数组或链表来表示。它要求根据具体图的需要设计恰当的结点结构和表结构, 常用的有邻接矩阵表示法和邻接表表示法。

邻接矩阵是表示顶点之间相邻关系的矩阵。假设 $G_{cg} = (G, C)$ 中具有 n 个图形实体, 即 G_{cg} 有 n 个顶点, 则 G_{cg} 的邻接矩阵是具有如下性质的 n 阶方阵

$$A_{ij} = \begin{cases} w_{ij} & (v_i, v_j) \in C(G) \\ 0 & (v_i, v_j) \notin C(G) \end{cases}$$

其中: A_{ij} 是邻接矩阵的元素; v_i, v_j 表示 G_{cg} 的顶点, w_{ij} 为元素的取值, w_{ij} 的取值和意义如表 1 所示。

表 1　w_{ij} 的取值

w_{ij} 的值	1	2	3	4	…
意义	直线与直线平行	直线与直线垂直	直线与圆(圆弧) 相切	圆弧与圆弧相切	…

按表 1 规定, 图 1 所示无向图用邻接矩阵可表示为

$$A = \begin{pmatrix} 0 & 2 & 1 & 2 \\ 2 & 0 & 2 & 1 \\ 1 & 2 & 0 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix}$$

对一幅复杂的平面图形来说, 图中每个实体往往只与相邻的实体有约束关系, 这样, 图中

边的数目远远小于 n^2, 用邻接矩阵表示必然造成存储空间的浪费; 同时, 在邻接矩阵中求约束的数目必须检测整个矩阵, 所耗费的时间是 $O(n^2)$, 与约束数量的大小无关; 除此之外, 当利用邻接矩阵表示时, 对图进行操作(插入、删除等)也不太方便。

邻接表表示法类似树的孩子链表表示法, 是图的一种链式存储结构。对于无向图中的每个顶点 v_i 该方法把所有邻接于 v_i 的顶点 v_j 链成一个单链表, 这个单链表就成为顶点 v_i 的邻接表。邻接表中每个结点由 3 个域构成, 其中邻接点域(adjvex)指示与顶点 v_i 邻接的点在图中的位置, 链域(nextarc)指示下一条边的结点; 数据域(info)存储和边或弧相关的信息。每个链表上附设 1 个表头结点, 在表头结点中, 除了设有链域(firarc)指向链表中第 1 个结点之外, 还设有存储顶点的名或其他有关信息的数据域(vexdata), 如图 2 所示。

表结点

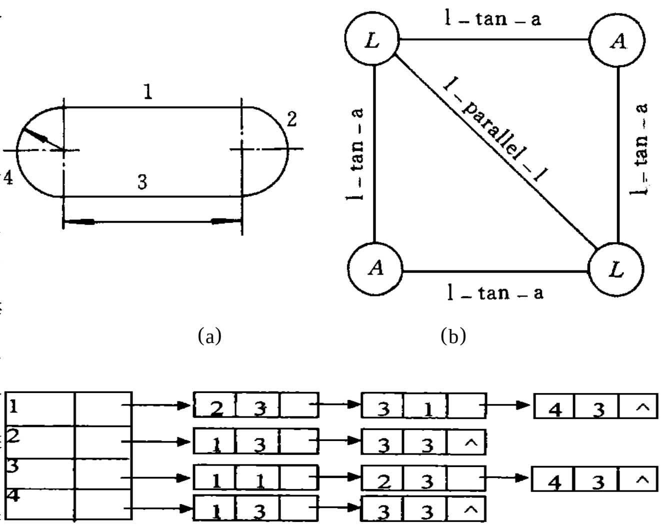
Adjvex	Info	Nextarc

头结点

vexdata	firstarc

图 2　邻接表结点

邻接表也有不足之处, 因为一个顶点 V_i 的邻接表完全可能包含 $O(n)$ 阶数目的顶点, 所以要确定边 $(i \rightarrow j)$ 是否存在也许要花费 $O(n)$ 阶时间, 而这也正是在图形求解过程中经常遇到的情况。如图 3(a)所示键槽由 2 个半圆和 1 对平行直线组成, 直线和半圆相切, 其几何约束系统用邻接表存储(图 3(c))。可以看到, 圆弧的圆心和半径都已完全确定, 要确定直线的 2 个端点, 需利用直线和圆弧相切这一条件, 根据图 3(c)所示邻接表结构, 要由结点 1 查询与之相切的结点 4, 需要依此遍历图中所有结点。



(a)　　(b)

(c) G_{cg} 图的邻接表

图 3　键槽的 G_{cg} 图及邻接表

3　分类邻接表存储结构

考虑到以上 2 种存储结构的缺陷, 笔者采用了一种复合嵌套链表来存储无向图[3], 具体方法是: 将二维图中可能存在的各种结构约束进行分类, 然后对顶点 v_i 和其邻接点 v_j 构成的约束归类, 将和 v_i 构成同一类约束的 v_i 的邻接点链成一个单链表, 定义此单链表为顶点 v_i 的分类邻接表。这种存储结构和邻接表的不同之处仅在于表头结点增加了指向不同类邻接点的指针域; 另外由于这种方法对边的种类进行了区分, 单链表中每个表结点只要有邻接点域(Adjvex)和链域(Nextarc)即可, 即去掉了信息域(Info), 如图 4 所示。

表结点

Adjvex	Nextarc

头结点

vexdata	firstarc	1 parallel	1 tan a	…

图 4　分类邻接表结构

尽管头结点增加了指针域, 但相应表结点也减少了 1 个存储约束种类的信息域, 占用空间并没有增加。当然, 并非所有的顶点都存在各种约束的邻接点(如直线不可能存在圆弧和圆弧相切这种约束), 这必然会造成头结点存储单元的浪费, 为避免这种缺点, 将所有顶点按其类别建立链表, 如建立直线、圆、圆弧 3 个链表, 然后顺序存储图中所有顶点, 即根据图形实体种类建立 3 个顶点表, 这样便可避免只有 1 个顶点所带来的存储空间上的浪费。图 5 是图 3(b)的分类邻接存储形式。现在利用上述存储结构求图 3 键槽图形中直线 1 的端点坐标时就会发现, 由于对约束进行了分类, 程序中便可增加判断语句跳过一些无用的邻接点, 如水平线的端点不可能通过与它平行的直线求出, 因此便可跳过这些邻接点直接遍历与其相切的圆弧。由此可见, 采用分类邻接表存储确实可降低时间复杂度, 提高求解效率。

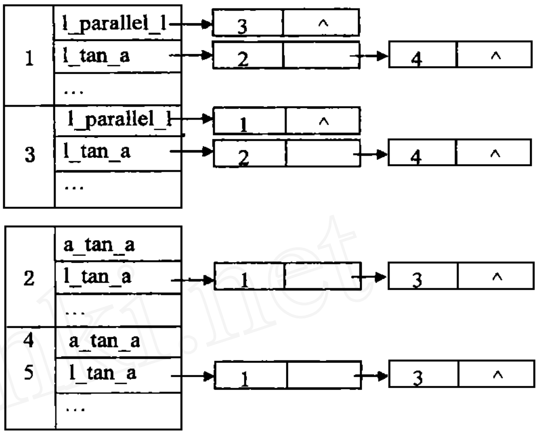


图 5　分类邻接表

4　结　论

采用一种合理的无向图存储结构来描述针对几何模型所建立的约束图是建立图形参数化数学模型的关键。笔者在深入研究现有无向图存储结构的基础上, 结合二维工程图形的特点, 提出的具有参数化特性的分类邻接表存储结构已成功应用在笔者目前正从事的课题“二维零件图和装配图一体化设计的研究”中。实践表明, 采用这种存储结构所建立的多视图参数化模型可大大降低图形求解程序的时间复杂度, 且稳定可靠。

参 考 文 献

1　严蔚敏, 吴伟民. 数据结构. 北京: 清华大学出版社, 1992. 6

2　陈立平. 基于实例图形的几何约束满足策略. 计算机辅助设计与图形学学报, 1996, 8(5): 381～388

3　梅树立. 基于约束分析的工程图参数化设计(视图联动)及装配图 CAD 的研究: [学位论文]. 北京: 中国农业大学, 1999