

文章编号:1671-6833(2002)02-0005-04

利用矩阵布尔乘 \odot 的图形拆分方法

谭同德¹,童秉枢²,李春雨¹

(1. 郑州大学信息工程学院,河南 郑州 450052; 2. 清华大学精密仪器与机械学系,北京 100084)

摘 要: 为了给基于实例的设计和基于原型的设计等智能 CAD 设计方法提供图形样本,需将已有的图形参数化.采用图论和拓扑学方法分析图形,研究了存储表达图形的邻接矩阵及寻找图形连通路径的道路矩阵后,推导并定义了矩阵布尔乘 \odot ,建立了寻找从 V_i 点到 V_j 点不重复路径的计算方法,解决了图形的拆分问题,提供了参数化的运算基础,并给出了实现图形自动参数化的算法流程.

关键词: 矩阵布尔乘 \odot ;图形拆分;道路矩阵

中图分类号: TP 181.TG 386 **文献标识码:** A

基于实例的设计及原型的设计等智能 CAD 设计方法^[1,2],要使用已有的设计图形作为样本,此时需将图形参数化.为了自动实现图形参数化,要将一幅复杂的图形拆分成易于识别的简单的图形集合.文献[3]曾提出一种自动参数化图形的方法,该方法建立在图论和拓扑学的理论基础上,应用矩阵作为分析工具,利用自定义的矩阵运算规则拆分图形.本文针对图形拆分、识别和参数化中涉及的图论、拓扑学理论和矩阵运算方法进行讨论.

1 图论与拓扑学理论基础

根据图论的子图拆分理论将图形拆分成回路与开链,按由外到内的次序排列回路与开链,使无序的图形变成有序的实体系列;利用约束建立实体间的拓扑关系和几何关系,将实体与子图关联成与原图形同构的图形^[4];再以一组参数与关联的几何实体序列联系,每一个参数代表序列中的一个实体,参数值就是它代表的几何实体值,改变参数值就改变了相应的几何实体值,这样,引起图形同构变换,实现参数化设计.

在图论中利用矩阵分析图形,找出实体间的联系,研究这些联系,从中得到内在规律,以解决需要解决的问题.

邻接矩阵 A 表示图 G 顶点之间的邻接关系,它是一个 $n \times n$ 阶的方阵.邻接矩阵可以表示图 G 中的自环,但不能表示重边.邻接矩阵存储了无重边的图 G 的全部信息,也就是说,如果已知 A 是图 G 的邻接矩阵,那么就很容易画出图 G 本身^[5].

$$A = \begin{matrix} & \begin{matrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{matrix} \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

道路矩阵 P 可由对邻接矩阵进行布尔乘得到:

$$P = A \vee A^2 \vee A^3 \vee \cdots A^n.$$

其中: $A^l = A^{l-1} \wedge A (l = 2, 3, \cdots, n)$; $a_{ij}^l = \vee (a_{ik}^{l-1} \wedge a_{kj}) (k = 1, 2, 3, \cdots, n)$.

道路矩阵的 P_{ij} 表示从顶点 V_i 到顶点 V_j 存在道路,否则 $P_{ij} = 0$.矩阵 A^l 中的 $a_{ij}^l = 1$ 表示从顶点 V_i 经过 l 条边到达顶点 V_j ,矩阵 A^l 的每一行代表一个点到其它点的路径.

例如:图 1 的邻接矩阵中,其道路矩阵中的各组成分量如下

收稿日期:2002-01-10;修订日期:2002-03-20

基金项目:国家自然科学基金资助项目(59875048)

作者简介:谭同德(1950—),男,山东省莱阳市人,郑州大学副教授,博士,主要从事计算机辅助设计、专家系统、人工智能方面的研究.

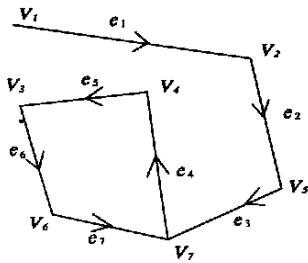


图 1 连通图

Fig. 1 Connected graph

$$A^2 = \begin{matrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} ;$$

$$A^3 = \begin{matrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix} ;$$

$$A^4 = \begin{matrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} ;$$

$$A^5 = \begin{matrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix} ;$$

$$A^6 = \begin{matrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} ;$$

$$A^7 = \begin{matrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix} .$$

矩阵 $A, A^2, A^3, A^4, A^5, A^6, A^7$ 分别表示 V_i 点经过一条边、二条边、三条边、四条边、五条边、六条边、七条边到达 V_j 点, 每一行代表一个点的路径. 例如, 第一行代表 V_1 点的路径, $V_1(0100000)$ 表示 V_1 经过一条边到达 V_2 , 即 $V_1 \rightarrow V_2$; $V_1^2(1000100)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5$, $V_1 \rightarrow V_2 \rightarrow V_1$; $V_1^3(0100001)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2$; $V_1^4(1001110)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1$; $V_1^5(0110001)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4 \rightarrow V_3$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6 \rightarrow V_3$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5 \rightarrow V_7$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1$; $V_1^6(1001110)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4 \rightarrow V_3 \rightarrow V_6$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6 \rightarrow V_3 \rightarrow V_4$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1$; $V_1^7(0110001)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4 \rightarrow V_3 \rightarrow V_6 \rightarrow V_7$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6 \rightarrow V_3 \rightarrow V_4 \rightarrow V_7$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5 \rightarrow V_7$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5 \rightarrow V_7 \rightarrow V_5 \rightarrow V_2$, $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_2$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4 \rightarrow V_3 \rightarrow V_6 \rightarrow V_3$, $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6 \rightarrow V_3 \rightarrow V_4 \rightarrow V_3$.

2 自定义的矩阵的布尔乘 \odot

从以上的分析可知, 可以用邻接矩阵的布尔

乘运算求图形的路径,因为当 V_i 点与 V_j 点相邻时,邻接矩阵 A 的 i 行 j 列元素 $a_{ij}=1$,而 j 行记录 V_j 点的信息;如果 V_j 点与 V_k 点相邻, j 行 k 列的元素 $a_{jk}=1$.当 i 行与 k 列布尔乘时, $a_{ij} \wedge a_{jk}=1$,运算后的 i 行 k 列元素 $a_{ik}=1$,即表示路径 $V_i \rightarrow V_j \rightarrow V_k$.运算时有的点重复相乘,如 V_1^3 中的路径 $V_1 \rightarrow V_2 \rightarrow V_1 \rightarrow V_2$,重复点表示路径的后退,应当排除掉.

定义1 矩阵的布尔乘 \odot , $A \odot A$,
其中: $A^l = A^{l-1} \odot A$ ($l=2,3,\dots,n$);
 $a_{ij}^l = \vee (a_{ik}^{l-1} \odot a_{kj})$ ($k=1,2,3,\dots,n$);
 $a_{ik}^{l-1}=0, a_{kj}=0$ 时, $a_{ik}^{l-1} \odot a_{kj}=0$;
 $a_{ik}^{l-1}=1, a_{kj}=0$ 时, $a_{ik}^{l-1} \odot a_{kj}=0$;
 $a_{ik}^{l-1}=0, a_{kj}=1$ 时, $a_{ik}^{l-1} \odot a_{kj}=0$;
 $a_{ik}^{l-1}=1, a_{kj}=1$ 时, $a_{ik}^{l-1} \odot a_{kj}=0$ (V_j 点是路径的 $L-2$ 点);
 $a_{ik}^{l-1}=1, a_{kj}=1$ 时, $a_{ik}^{l-1} \odot a_{kj}=1$.
这样就能以运算 \odot 求图形的路径,以点 V_i 为

始点的路径可通过 V_i 行与邻接矩阵 A 相 \odot 求得,即 $V_i(a_{i1}a_{i2}\dots a_{in}) \odot A \odot A \dots A$.例如,图1第一行 V_1 点的运算 \odot 如下:
 $V_1(0100000)$ 表示 V_1 经过一条边到达 V_2 ,即 $V_1 \rightarrow V_2$;
 $V_1^2(00000100)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5$;
 $V_1^3(00000001)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7$;
 $V_1^4(0001010)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4$,
 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6$;
 $V_1^5(00100000)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4 \rightarrow V_3$,
 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6 \rightarrow V_3$;
 $V_1^6(0001010)$ 表示 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_4 \rightarrow V_3 \rightarrow V_6 \rightarrow V_7$,
 $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_6 \rightarrow V_3 \rightarrow V_4 \rightarrow V_7$.

3 算法流程

3.1 图形拆分的算法流程

利用矩阵布尔乘 \odot 拆分图形的算法流程如图2所示.图中列出了拆分开链的算法流程和拆分

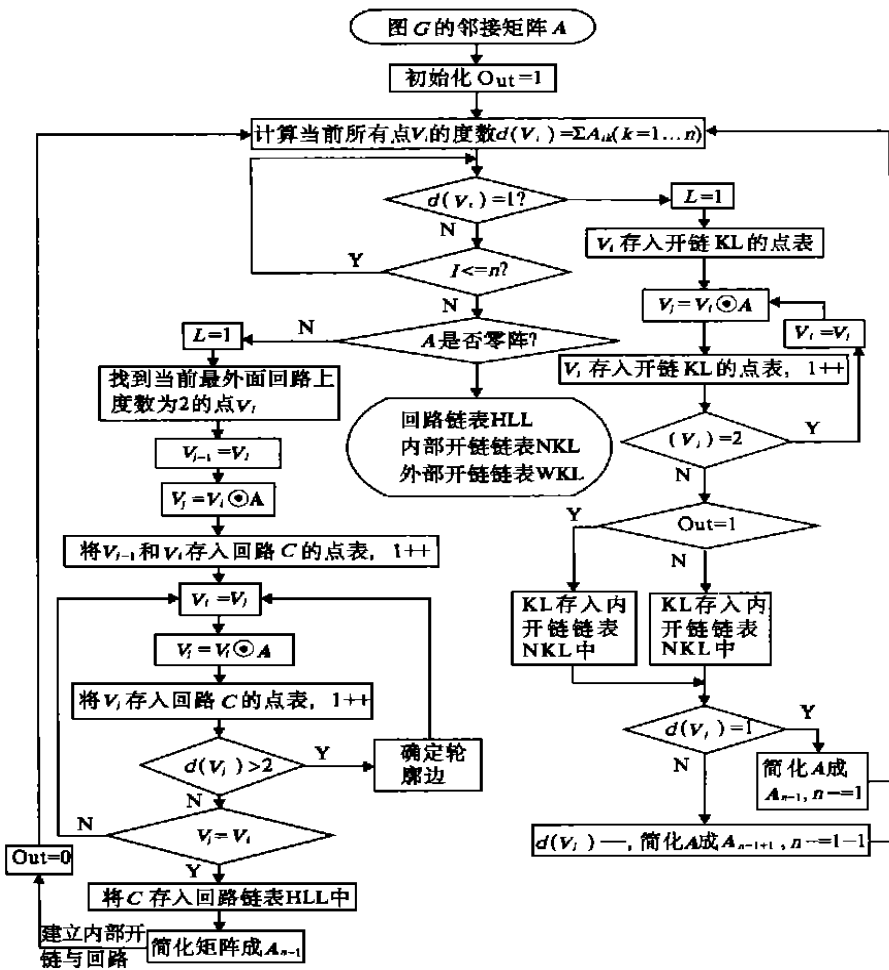


图2 图形拆分的算法流程

Fig.2 Algorithm flow of dsassembling graph

最外面轮廓回路的算法流程,首先计算点的度数,当找到度数满足 $d(V_i)=1$ 的点时,进入拆分开链的分支算法,循环计算,拆分出所有的开链;然后进入拆分最外面轮廓回路的分支算法,拆分出最外面的轮廓回路;简化邻接矩阵 A 后,再从头进行循环,开始拆分内部开链与回路.拆分内部开链与回路的算法流程与它们相同,只在循环控制上略有差异.一个物体的形状必定是回路连通图,它的外部轮廓必对应连通图最外面的那一条回路,因此,首先要寻找、记录的是最外面的回路.采用矩阵运算 \odot 寻找最外面的回路时,难点是遇到度数大于2的点,度数大于2表示至少有三条边经过该点,需要判断哪一条边代表外部轮廓,要找到这条边,沿这条边继续进行矩阵运算 \odot .

3.2 确定通过 $d(V_i)>2$ 的点的路径

从起点开始,沿一条边前进,依照连通域在边的那一面及前进方向,建立右手回路或左手回路:一个人面向前进方向,他的右边是连通域内,即右手回路;他的左边是连通域内,即左手回路.当遇到度数大于2的点,计算前进边与过此点的其它边之间的夹角,右手回路按逆时针方向计算,左手回路按顺时针方向计算,比较夹角的大小,夹角最大的那条边即是回路最外面的边.如图3所示, e_1 是前进边,箭头表示它的方向.它的判定方法如下:

if 是右手回路
then 按逆时针方向计算 $\angle e_1V_e2$ 和 $\angle e_1V_e3$
(逆时针方向为正);
else 按顺时针方向计算 $\angle e_1V_e2$ 和 $\angle e_1V_e3$
(顺时针方向为正);
if $\angle e_1V_e2 < \angle e_1V_e3$
then e_3 是回路最外面的边;
else
 e_2 是回路最外面的边.

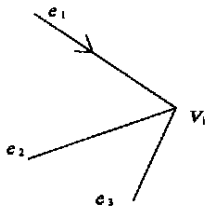


图3 经过度数大于2的点的路径示意图
Fig.3 Path chart of point with $d(V_i)>2$

4 拆分实例

图4是液压支架的底座图,它没有起点度数

为1的开链,所以先拆分最外面的轮廓.拆分步骤如下:

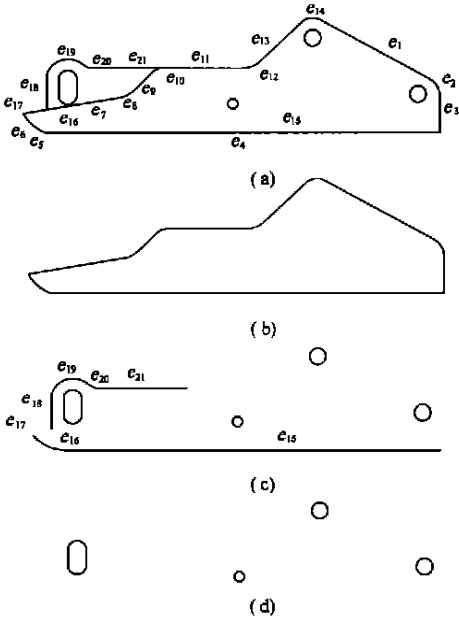


图4 液压支架底座拆分图
Fig.4 Connected graph

- (1) 从实体 e_1 开始,拆分出最外面的轮廓 $e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}$,如图4(b)所示.从实体表中排除图4(b),余下的实体如图4(c)所示.
- (2) 从余下的实体中拆出开链 e_{15}, e_{16}, e_{17} ,从实体表中排除实体 e_{15}, e_{16}, e_{17} .
- (3) 从余下的实体中拆出开链 $e_{18}, e_{19}, e_{20}, e_{21}$,从实体表中排除这些实体.实体表中还余实体如图4(d)所示.
- (4) 从图4(d)所示实体中拆出闭环 e ,从实体表中排除闭环 e ,实体表中还余下3个圆.
- (5) 建立3个圆的序列表,完成图形拆分.

5 结论

本文研究了存储表达图形的邻接矩阵,并寻找图形连通路经的道路矩阵,对道路矩阵的布尔乘进行修改,推导出矩阵布尔乘 \odot ,建立了寻找从 V_i 点到 V_j 点不重复路径的计算方法,使得回路与开链的拆分变为矩阵运算,解决了图形的拆分问题,提供了实现自动图形参数化的运算基础.应用实例表明了该拆分方法的正确性.以此原理为基础开发的图形翻译器,正确地识别图形,实现了样本图形的自动参数化.

(下转第30页)

YANG Jie¹, LUO Ting-liang¹, CHEN Jin-zhong¹, LI Yu¹,
WANG Liang², ZHU Mn-qing³, LIU Guo-ji¹

(1.College of Chemical Engineering ,Zhengzhou University , Zhengzhou 450002, China ;2.Nlon 66 Plant of Engdingshan , Engdingshan 467013, China ;3.Baling Designing Institute of Petrochem. , Yueyang 414003,China)

Abstract : This paper studies the influence of some different carriers of catalysts of preparation of 1,4-naphtho-quinone from naphthalene oxidation on reaction activity .These carriers include pumice ,bergmeal ,arenaceous quartz , titanium white ,bearing silica gel ,Si -Al colloid ,active floridin ,carbon -white .The catalysts are prepared by im-mersion method .It is found that the application of special disposal carriers is suitable for the production of naphtho-quinone .The technological conditions of preparation of two kinds of carriers ,Si -Al colloid and carbon -white is in-vestigated .

Key words : naphthalene ; naphthoquinone ; catalytic oxidation ; carriers of catalysts

(上接第 8 页)

参考文献:

[1] THAM K W , LEE H S , GERO J S et al . Building enve-
lope design using design prototypes[J] . ASHRAE Trans ,
1990, 96(2) : 508—520.

[2] QLAN Lena , GERO John S . Function —behavior —structure
paths and their role in analogy —based design[J] . Arti-fi-
cial Intelligence for Engineering Design , Analysis and
Manufacturing , 1996, (10) : 289—312.

[3] 谭同德, 童秉枢, 王 涛. 图形翻译器的研究[J] .
清华大学学报, 2000, 40(8) : 49—52.

[4] 戴一奇, 胡冠章, 陈 卫. 图论与代数结构[M] . 北
京: 清华大学出版社, 1995.

Disassembling Graph Method with Matrix Boolean Multiplication ⊙

TAN Tong -de¹, TONG Bing -shu², LI Chun -yu¹

(1.College of Electronic & Information , Zhengzhou University , Zhengzhou 450052, China ; 2.Department of Precision Instruments and
Mechanology , Tshinghua University , Beijing 100084, China)

Abstract : Existed graph will be interpreted to parametric graph to offer graphic samples for intelligent CAD design methods , such as case based design and prototype based design , etc . Graph is analyzed by means of graph theory and topology . After studying the adjacent matrix , which saves and represents graph , and the road matrix , which seeks for continue paths , the matrix Boolean multiplication ⊙ is inferred and defined , the algorithm is developed to seek a path from a point V_i to a point V_j , the problem of disassembling graph is solved , the compute base of param-eterization is established , and the flow of algorithm is given to implement automatically parameterization graphs .

Key words : matrix Boolean multiplication ⊙ ; disassembling graph ; road matrix