

# 一个基于变形映射的约束图有向化与求解方法

金建国 谭建荣

(浙江大学CAD&CG国家重点实验室 杭州 310027)

E-mail: tanjr@mech.zju.edu.cn

**摘要** 提出了变形映射和变形源的概念,给出一种基于变形映射的二维工程图约束图有向化与求解新方法.同时,提出一种基于变形映射的约束图有向化新方法,并在约束图分解的基础上,给出了一个基于复合节点及虚拟约束的约束图变形求解新算法.应用此方法,可提高约束图有向化的可靠性、降低计算复杂度,且能方便地求解带有循环约束的约束图.

**关键词** 变形映射,修正因子,虚拟约束,约束图求解,约束图分解,约束图有向化.

**中图分类号** TP91

参数化设计是一项有着广泛应用的新兴技术,它主要研究约束的满足和约束的维护方法.根据约束对象的不同,约束可分为几何约束和其他类型的约束.几何约束是对变量几何的类型、属性、定形和定位的限制,用于保证几何模型构造和修改的可实现性和有效性.设计人员可以方便地利用几何约束表达其设计意图而不必考虑其具体细节.其他类型的约束则是指对产品功能、性能、工艺、制造和成本等方面提出的要求或应达到的指标.这类约束与几何模型的构成无直接关系,需经过约束的传播与转换变成几何约束形式施加于几何模型.

目前,基于约束的求解方式很多,主要有代数法<sup>[1]</sup>、符号法<sup>[2]</sup>和几何推理法<sup>[3]</sup>.而约束满足主要考虑约束图有向化方法、分解方式和求解算法3个问题.

各国学者<sup>[4~6]</sup>针对这些问题都做了许多工作,并取得了很多成果,但其中带有循环约束的约束图有向化问题始终是一个颇为棘手的问题.也有一些学者提出了这样一种方法,即随意初始化约束图中的约束边方向,然后不断地递归调整各约束边的方向,使满足该调整边所约束的节点不出现过载,并最终趋于稳定.这种方法的优点在于算法较为简便,但其缺点是算法的可靠性难以确切保证.另外,当约束图无效时,特别是有过约束时,算法进入死锁,此时系统难以确切判断该约束图是否有效.针对这些情况,本文提出了一种利用变形技术解决约束图有向化和基于复合节点求解变形约束图的新思路和新方法.该方法首先把约束图有向化地转化为求目标函数(即约束图各元素节点的自由度与进入该节点的约束作用度之和的差的平方和)的极小值问题,然后引入变形映射来求解该极值问题,并利用修正因子判断该约束图是否有效,再利用复合约束图 NewG 来确定推理顺序,用单节点图 OldG 实行具体的细节推理计算.应用此方法,由于转化为一个优化问题,因而能大幅度地提高约束图有向化的可靠性,并且能方便地求解带有循环约束的约束图.实践证明,本文的理论正确,方法有效.

## 1 基本概念

**定义 1(变形映射和变形源).** 在本文中,变形映射有两方面的含义:

- (1) 若我们记原函数为  $f(x)$ ,则称  $g(f(x), x')$  为  $f(x)$  的变形映射,并称  $x'$  为  $g(f(x), x')$  的变形源.
- (2) 若记原图为 OldG,则称  $NewG = Compound(OldG)$  为 OldG 的变形映射,并称 OldG 为映射 Compound

本文研究得到国家 863 高科技项目基金(No. 863 511-707-001)、国家教育部博士点基金(No. 98033506)和浙江省自然科学基金(No. 697010)资助.作者金建国,1970 年生,博士生,主要研究领域为参数化设计造型、曲线曲面造型、真实感图形学.谭建荣,1954 年生,博士,教授,博士生导师,主要研究领域为参数化设计,工程图扫描识别,CAD/CAM.

本文通讯联系人:谭建荣,杭州 310027,浙江大学 CAD&CG 国家重点实验室

本文 1999-02-06 收到原稿,1999-06-22 收到修改稿

的变形源。

定义 2(复合节点). 若在原有向图 OldG 中把满足一定关系的子图合并,作为一个节点,则我们称此新生成的节点为复合节点。

定义 3(虚拟约束). 把原有向约束图 OldG 的满足一定关系的节点合并成复合节点后重新调整,所生成的复合图的约束边称为虚拟约束。虚拟约束本质上是原约束图中的约束复合后的一种表示形式。

定义 4(修正因子). 称有向约束图中某几何元素节点的自由度与进入该节点的有向约束边的作用度(对所约束的几何元素的自由度的降低量)之和的差为该节点的修正因子。若修正因子大于 0,则该节点为欠约束节点。若小于 0,则该节点为过载约束节点。

定义 5(m 维单位立方体顶点的邻域). 对 m 维单位立方体  $[(0,0,\dots,0),(1,1,\dots,1)]$  的顶点 X, 定义其邻域为  $N(X) = \{X \pm e_i \mid X \pm e_i \text{ 仍为此立方体的顶点}, i = \overline{1,m}\}$ 。其中  $e_i = (0,0,\dots,1,0,\dots,0)$ , 表示第 i 个分量为 1 的 m 维单位向量。

定义 6(f(X) 的极小解). m 维单位立方体顶点  $X_0$  称为 f(X) 的局部极小解, 如果  $\forall X \in N(X_0)$ , 则有  $f(X) \geq f(X_0)$ 。另外, 若对任意顶点 X 均有  $f(X) \geq f(X_0)$ , 则称  $X_0$  为 f(X) 的全局极小解。

定义 7(f(X) 的可行下降方向). 任给 m 维单位立方体顶点  $X_0$ , 令  $E = \{e_i^+ , e_i^- \mid e_i^+ = e_i, e_i^- = -e_i, i = \overline{1,m}\}$ , 若  $X_0 + d_i$  仍为此单位立方体的顶点, 且  $f(X_0 + d_i) < f(X_0)$ , 则  $d_i \in E$  称为 f(X) 在  $X_0$  处的可行下降方向。

定义 8(理想极小值点). 函数  $f(X) \geq 0$  在 m 维单位立方体  $[(0,0,\dots,0),(1,1,\dots,1)]$  的顶点上有定义, 若某顶点  $X_0$  满足  $f(X_0) = 0$ , 则称  $X_0$  为函数 f(X) 的理想极小值点。

## 2 约束图的有向化与图分解

### 2.1 约束图的有向化

#### 2.1.1 查找表的建立

对于给定的一张无向约束图, 为了建模方便起见, 我们给出与其对应等价的查找表形式。若记几何元素为  $g_1, g_2, \dots, g_n$ , 约束为  $c_1, c_2, \dots, c_m$ , 则我们建立如下一些查找表。

(1) 元素查找表(见表 1 和表 2)

Table 1 Element searching table

表 1 元素查找表

Element <sup>①</sup>	$g_1$	$g_2$	$g_3$	...
Symbolic <sup>②</sup>	1	2	3	...
Type No. <sup>③</sup>	0	1	2	...

①元素, ②代号, ③类型编号。

Table 2 Basic element table

表 2 基本元素表

Type <sup>①</sup>	Point <sup>②</sup>	line <sup>③</sup>	Circle <sup>④</sup>	Circular arc <sup>⑤</sup>
No. <sup>⑥</sup>	0	1	2	3
Freedom degree <sup>⑦</sup>	2	2	3	5

①类型, ②点, ③线, ④圆, ⑤圆弧, ⑥编号, ⑦自由度。

(1) 约束查找表(见表 3)

Table 3 Constraint searching table

表 3 约束查找表

Constraint <sup>①</sup>	No. <sup>②</sup>	Element <sup>③</sup> 1	Element 2	Constraint property <sup>④</sup>	Dimension <sup>⑤</sup>	Function degree <sup>⑥</sup>	Direction <sup>⑦</sup>	Using order <sup>⑧</sup>
$c_1$	1	$g_6$	$g_8$	Distance <sup>⑨</sup>	Dim	2	0	2
$c_2$	2	$g_2$	$g_5$	Angle <sup>⑩</sup>	Ang	1	1	6
...	...	...	...	...	...	...	...	...

①约束, ②编号, ③元素, ④约束性质, ⑤尺寸值, ⑥作用度, ⑦定向, ⑧使用序号, ⑨距离, ⑩角度。

(III) 元素约束关联表(见表 4)

Table 4 Element and constraint relating table

表 4 元素约束关联表

Element <sup>①</sup>	Relating element <sup>②</sup> 1	Relating constraint <sup>③</sup> 1	Relating element 2	Relating constraint 2	...
$g_1$	$g_3$	$b_{11}$	$g_5$	$b_{12}$	...
$g_2$	$g_5$	$b_{21}$	$g_8$	$b_{22}$	...
...	...	...	...	...	...

①元素,②关联元素,③关联约束.

以上一些查找表的建立,为下面建立约束图有向化数学模型时提供必要的信息打下了基础.

2.1.2 数学模型的建立

约束图的有向化问题实际上等价于这样一个问题,即确立约束图中各个约束边的方向,使满足约束图中各个元素节点的修正因子的平方和为最小.因此,对应于约束查找表中的定向项,我们引入 0-1 型整数变量  $x_i (i=1, \dots, m)$ . 我们作如下约定,对应于每个  $c_i (i=\overline{1, m})$ ,若  $x_i$  为 0,则表示约束边方向是从元素 1 指向元素 2 的;否则,若  $x_i$  为 1,则表示约束边是从元素 2 指向元素 1 的.由此,对应于约束集  $\{c_i | i=\overline{1, m}\}$ ,我们构造  $m$  维空间整数向量点  $X(x_1, x_2, \dots, x_m)$ ,并记  $\{e_i | i=\overline{1, m}\}$  为此  $m$  维线性空间的一组坐标轴基,即  $e_i = (0, 0, \dots, 1, 0, \dots, 0)$ ,其中  $e_i$  表示第  $i$  个分量为 1,其余为 0 的  $m$  维矢量,则我们的约束图有向化问题可转化为求解以下数学模型:

$$\begin{aligned} \min f(X) &= \sum_{j=1}^n (cf_j - d_j)^2, \\ cf_j &= \sum_{k=1}^{N_j} w(b_{p_k}) \text{direction}(b_{p_k}), \\ \text{direction}(b_{p_k}) &= \begin{cases} x_l, & x_l \text{ 为 } 0 \text{ 或 } 1 \\ 1 - x_l, & l \in \overline{1, m} \end{cases} \end{aligned} \quad (1)$$

其中每个节点的修正因子  $Correct_j = d_j - cf_j$ ,即求目标函数  $f(X)$  的极小值,其中  $d_j$  表示元素  $j$  的自由度,  $cf_j$  表示约束图中指向节点  $j$  的约束的作用度之和,  $w(b_{p_k})$  表示指向节点  $j$  的约束  $b_{p_k}$  的作用度.设约束  $b_{p_k}$  即为约束查找表中的约束  $c_l$ ,若节点  $j$  是约束  $c_l$  中的元素 1,则  $\text{direction}(b_{p_k})$  取为  $x_l$ ;否则,若节点  $j$  是约束  $c_l$  中的元素 2,则我们取  $\text{direction}(b_{p_k})$  为  $1 - x_l$ .所有的这些信息均可从相应的查找表中获得.

为了下面求解该问题时方便叙述,我们把式(1)转换成如下等价形式:

$$\min f(X) = f(x_1, x_2, \dots, x_m). \quad (2)$$

它满足:  $X$  为  $m$  维单位立方体  $[(0, 0, \dots, 0), (1, 1, \dots, 1)]$  的顶点.

2.1.3 有向化算法

约束图的有向化已转化为对式(2)的求解.为此,我们引入  $f(X)$  的变形映射  $g(f(X), X_1^*)^{[7,8]}$ .

$$g(X, r, \rho, X_1^*) = \frac{1}{r + f(X)} \exp\left(-\frac{\|X - X_1^*\|^2}{\rho^2}\right), \quad (3)$$

其中  $\|\cdot\|$  表示  $X$  和  $X_1^*$  两点间的距离,  $X_1^*$  为变形源,  $r, \rho$  为参数.此变形映射具有如下良好的性质<sup>[7,8]</sup>: 若  $h \leq \min |f(X_1) - f(X_2)|$ , 其中  $f(X_1) \neq f(X_2)$ ,  $X_1$  和  $X_2$  为  $m$  维单位立方体  $[(0, 0, \dots, 0), (1, 1, \dots, 1)]$  的顶点,参数  $r, \rho \neq 0$ , 且满足  $0 < r + f(X_1^*) < h, \rho^2 \ln \frac{r + \sup(f)}{r + f(X_1^*)} < 1$ , 其中  $\sup(f)$  为  $f$  的一个上界.则有:若  $X_1^*$  是  $f(X)$  的一个极小值的话,  $g(X, r, \rho, X_1^*)$  把原曲面  $f(X)$  在  $X_1^*$  处的盆地变形成为山峰.并且在新曲面  $g(X, r, \rho, X_1^*)$  上,原曲面  $f(X)$  上的比  $X_1^*$  处盆地地势要高的其他所有盆地的极小值点必定不是  $g(X, r, \rho, X_1^*)$  的极小值点(曲面边界点除外).而且若在  $f(X)$  的其他地方存在比  $X_1^*$  处盆地地势更低的盆地的话,则在  $X_1^*$  处沿着某方向在地势更低的盆地内必定存在  $g(X, r, \rho, X_1^*)$  的一个极小值点  $X_2^*$ , 且  $g(X_2^*, r, \rho, X_1^*) < 0$ .利用这个性质与成果,再结合这个问题的实际,下面我们给出约束图的有向化算法.

算法 1. 局部极小值点最速下降搜索算法

(1) 在  $m$  维单位立方体  $[(0,0,\dots,0),(1,1,\dots,1)]$  上任取一初始顶点  $X_0$  (我们一般取约束查找表中初始定向为  $X_0$  的初值).

(2) 若  $X_0$  是  $f(X)$  的局部极小值点, 则返回; 否则, 搜寻  $f(X)$  在  $X_0$  处的可行最速下降方向  $d_0$ .

(3) 令  $X_0 = X_0 + d_0$ , 转步骤(2).

若记  $N$  为迭代次数,  $NMAX$  为允许迭代的最大次数, 则我们有算法 2.

算法 2. 约束图的有向化算法

(1) 取一个  $m$  维单位立方体  $[(0,0,\dots,0),(1,1,\dots,1)]$  顶点  $X_0$  作为初始点, 令  $N=0$ .

(2) 若  $N > NMAX$ , 则返回, 并取  $X_0$  为全局极小值点 (若此时有  $f(X_0) = 0$ , 则提示此约束图为有效约束图, 有向化成功). 否则, 以  $X_0$  为初始点利用算法 1 寻找  $f(X)$  的一个局部极小值点  $X_1^*$ ,  $N = N + 1$ . 若  $X_1^*$  点是  $f(X)$  的理想极小值点, 即  $f(X_1^*) = 0$ , 则返回, 并提示此约束图为有效约束图, 有向化成功.

(3) 否则, 作以  $X_1^*$  为变形源的变形映射:

$$g(X, r, \rho, X_1^*) = \frac{1}{r + f(X)} \exp\left(-\frac{\|X - X_1^*\|^2}{\rho^2}\right),$$

其中根据参数需满足的条件, 可取

$$\begin{cases} r = 0.4 - f(X_1^*) \\ \frac{1}{\rho^2} = 1 + \ln\{1 + 2.5[\sup(f) \cdot f(X_1^*)]\} \end{cases}, \text{ 令 } X_1 = X_1^*.$$

(4) 以  $X_1$  为初始点搜寻  $g(X, r, \rho, X_1)$  的一个局部极小点  $X_2^*$ . 若  $g(X_2^*, r, \rho, X_1) < 0$ , 令  $X_0 = X_2^*$ , 转步骤(2). 否则, 轮换考察  $X_1$  的邻域中除  $X_1^*$  外的顶点列  $\{Y_i\}$ , 若所有以  $\{Y_i\}$  为初始点搜索到的局部极小点  $X_2^*$  均使  $g(X_2^*, r, \rho, X_1) \geq 0$ , 则返回, 并以  $X_1^*$  作为全局极小值; 否则, 令  $X_0$  等于满足  $g(X_2^*, r, \rho, X_1) < 0$  的任一  $X_2^*$ , 转步骤(2).

利用上述算法返回得到的全局极小值点即为所求. 值得注意的是, 除第 1 次计算  $f(X)$  的值以外, 其他时候并不需要对式(1)中的每一项  $j(j = \overline{1, n})$  全部重新计算, 而只要作局部修改计算即可. 另外, 带有循环约束的有效约束图的有向化往往存在着多解, 即在那些循环约束子图内部的约束边方向往往可以改变, 但我们只要找到其中的一个解就可以了.

2.1.4 过欠约束的检查

在过欠约束的检查方面, 可利用上节的结果. 即若算法 2 返回的是理想极小值点, 我们就断定此约束图是有效的, 不存在过欠约束问题. 否则, 我们就检查各元素节点的修正因子, 若节点的修正因子大于 0, 则此节点为欠约束节点; 若小于 0, 则必为过载约束节点, 程序提示用户进行修正.

2.2 约束图分解

在约束图已有向化的基础上, 有向图的强连通子图<sup>[5,9]</sup>在此表示必须整体求解的约束图的子集. 在强连通子图中, 所有节点互为依赖, 因此需要联立求解. 由此, 对有向图 OldG 进行强连通子图复合计算<sup>[6,9]</sup>, 即将图上所有强连通子图分别标记为一个复合节点, 并引入虚拟约束边对原约束图重新进行修正. 此过程我们也可以理解为是对约束图的变形映射过程, 即新得到的复合约束图  $NewG = Compound(OldG)$ . 例如, 考察以下实例:

已知: 给定  $A$  点坐标, 边  $c$  为水平线,  $C$  到  $AB$  的距离为  $h_1$ ,  $\angle C$  为已知角,  $CB$  的长度为  $d_1$ ,  $BD$  和  $BC$  的夹角为  $\alpha$ ,  $D$  到  $BE$  的距离为  $h_2$ ,  $DE$  长度为  $d_2$ ,  $BE$  长度为  $d_3$ .

求:  $\triangle ABC$  和  $\triangle BDE$  (如图 1 所示).

若分别以  $a, b, c$  记  $\triangle ABC$  的 3 条边  $BC, AC, AB$ . 以  $b', d, e$  记  $\triangle BDE$  的 3 条边  $DE, BE, BD$ . 以  $bCaB$  记由元素  $b, C, a, B$  及它们

之间的约束边构成的子图, 以  $DdE$  记由元素  $D, d, E$  及它们之间的约束边构成的子图, 则一个可能的约束图分解方案如图 2 所示, 其中实边集表示需整体求解的约束子集 (实际上这个图的分解是唯一的, 只不过实边集即强

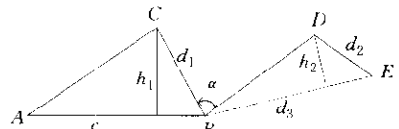


Fig. 1 图1

连通子集内部的约束边方向往往可以改变,所以我们在这里仅说是一个可能的分解方案).

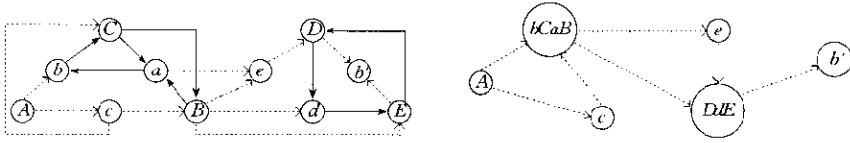


Fig. 2 The old constraint graph OldG (left) and the new compound graph NewG (right) after affected by deformation mapping Compound  
 图2 原约束图OldG(左)与经过变形映射Compound作用后得到的新复合图NewG(右)

### 3 约束图的求解

在过欠约束的检测和 Compound 变形映射完成后,我们利用一个简洁的递归算法来求解获得的有效约束图.该算法能方便地求解带有循环约束的约束图.算法利用复合约束图 NewG 来确定推理顺序,利用原有向图 OldG 来确定求解细节.这里我们若用 CN 表示当前考虑的 NewG 中的节点,则求解算法可描述如下.

**算法 3.** 基于复合节点和虚拟约束的变形约束图求解算法

Compound\_Constraint\_Processing(CN, NewG, OldG)

```

{ if (CN 已知)
  { for(CN 的所有 NewG 中未用虚拟约束约束的复合子节点 CS)
    Compound_Constraint_Processing(CS, NewG, OldG);
  }
else
  { if (CN 未知,其复合自由度(复合节点内部包含的所有几何元素自由度之和)与约束该节点的虚拟约束的作用度(该虚拟约束包含的一系列原约束的作用度之和)之和相等,且这些约束的另一端均为已知复合元素)
    { 计算复合元素 CN;
      if (CN 本身内部只含一个几何元素,即是单节点)
        { if (与 CN 关联的这些已知复合元素本身都是单节点元素)
          { 直接调用相应的计算程序在 NewG 中求解 CN 的值 }
          else (若这些复合元素不都是单节点)
            { 在 OldG 中查找 CN 与这些复合元素内部节点间的约束关系,
              调用相应的计算程序求出 CN 的各个参数值 }
        }
      else (若 CN 是内含多个节点的复合元素)
        { 考察 CN 和与之关联的这些已知复合元素,在 OldG 中查找 CN 内单节点与这些复合节点内部单节点之间的约束关系,并利用这些约束关系,调用相应的计算程序,从已知复合元素内部节点推理出 CN 内部单节点的部分参数值.然后利用 Newton-Raphson 方法解方程组,以整体确定 CN 内部单节点的所有未知参数值 }
        标志此复合元素 CN 为已知,并标志所有这些虚拟约束为已用.
        For(复合节点 CN 的所有未用虚拟约束约束的子节点 CD)
          Compound_Constraint_Processing(CD, NewG, OldG);
    }
  }
}

```

#### 4 实例与结论

本文采用优化方法来解约束图的有向化问题,再把约束集分解成一个个子约束集,然后利用 NewG 进行递归推理,以确定复合节点的求解顺序,用 OldG 实行具体的细节推理计算的思路,这种方法与用递归调整约束边方向的方法相比,大大提高了约束图有向化的可靠性,与约束集整体求解的方法相比大幅度降低了计算复杂度(设约束图中各个元素需确定的参数总数为  $n$ ,经过约束分解并确定求解顺序后共分为  $k$  个约束子集,其中每个子集中可新确定的参数数为  $n_i (i=1, \dots, k)$ , 则  $\sum_{i=1}^k n_i = n$ . 由于用 Newton-Raphson 方法求解方程组的时间复杂度为  $O(m^3)$ , 其中  $m$  为方程组所含变量个数. 因此,若用 Raphson 方法整体求解的话,需耗时间  $O(n^3)$ , 而分解后再求解的时间耗为  $\sum_{i=1}^k O(n_i^3)$ , 显然,  $\sum_{i=1}^k O(n_i^3) \ll O(n^3)$ . 并且所提出的约束图求解算法能方便地求解带有循环约束的约束图,利用修正因子能正确地处理过欠约束问题. 实践证明,该思路是正确的、有效的. 如图3所示为我们用上述方法进行参数化设计的一个实例.

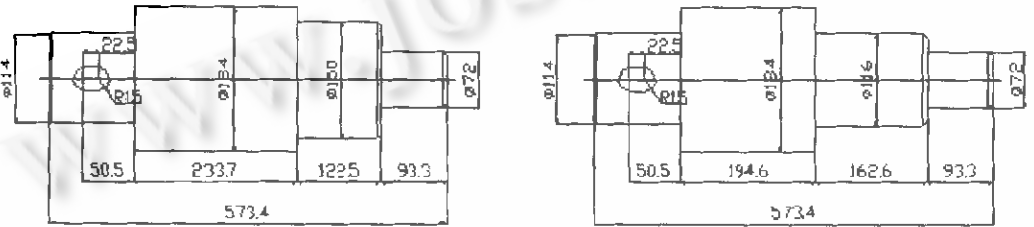


Fig. 3 An example of parametric design using deformation mapping method

图3 用变形思想来进行参数化设计的一个实例

致谢 本文得到了董玉德同学的帮助,在此表示衷心的感谢.

#### 参考文献

- Light R, Gossard D. Modification of geometric model through variation geometry. *Computer Aided Design*, 1982,14(4): 209~214
- Buchanan S A, de Pennington A. Constraint definition system, a computer-algebra based approach to solving geometric-constraint problems. *Computer Aided Design*, 1993,25(12):741~750
- Lee Jee Yeol, Kim K. Geometric reasoning for knowledge-based parametric design using graph representation. *Computer Aided Design*, 1996,28(10):831~841
- Latham R S, Middleditch A E. Connectivity analysis: a tool for processing geometric constraints. *Computer Aided Design*, 1996,28(11):917~928
- Cheng Li-ping, Xiang Wen, Zhang Xin-fang et al. A strategy for geometric constraint satisfaction based on case graphics. *Chinese Journal of Computer Aided Design and Computer Graphics*, 1996,8(5):381~388  
(陈立平, 向文, 张新访等. 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)
- Qian Xiao-pin, Xiang Wen, Huang Tong-jun et al. Rules-Based constraint driven variational geometry system. *Chinese Journal of Computer Aided Design and Computer Graphics*, 1995,7(3):166~171  
(钱小平, 向文, 黄彤军等. 基于规则求解的约束驱动的可变几何造型系统. *计算机辅助设计与图形学学报*, 1995,7(3):166~171)
- Ge Renpu. A filled function method for finding a global minimizer of a function of several variables. *Mathematical Programming*, 1990,46(2):191~204
- Zhu Wen-xing, Zhang Lian-sheng. An approximate algorithm for nonlinear integer programming. *Operation Research Transactions*, 1997,1(1):72~81

(朱文兴,张连生.非线性整数规划的一个近似算法.运筹学学报,1997,1(1):72~81)

9 Duff I E. Direct Method for Sparse Matrix. Oxford: Clarendon Press, 1986

## An Approach Based on Deformation Mapping for Constraint Graph Direction and Solution

JIN Jian-guo TAN Jian-rong

(State Key Laboratory of CAD&CG Zhejiang University Hangzhou 310027)

**Abstract** The concept of deformation mapping and deformation origin is described. A new approach for directing and solving constraint graph is presented, which is deformation mapping based and two-dimension engineering drawing oriented. A directive method for constraint graph, which is based on deformation mapping, is also proposed. Finally, a new approach for solving the deformation constraint graph is given, which is based on compound node and virtual constraint. With this method, the reliability to direct the constraint graph is increased, the computation complexity is reduced, and the cyclic constraint graph can be solved expediently.

**Key words** Deformation mapping, correcting factor, virtual-constraint, constraint-graph-division, constraint-graph-solution, constraint-graph-direction.