

大型有限元程序系统的网格 结点编码优化算法*

华伯浩

(上海计算技术研究所, 1980年3月18日收到)

摘 要

本文介绍了近十年来与大型有限元程序系统研制有关的网格结点编码的优化算法, 讨论了算法应用的若干技巧, 并给出 RCM 算法的一个比较有效的执行程序。

一、引 言

计算力学领域内, 大型有限元程序系统的研制在国际上已日趋完美, 著名的系统有: ASKA, NASTRAN, MARC 等数十个, 国内的研制工作则方兴未艾. 剖析其程序的主要部件不外包括:

- 1° 结构数据库;
- 2° 结构元素库;
- 3° 总刚度矩阵的集成与求解模块 (线性方程组、特征方程组);
- 4° 程序控制及后处理等其它模块.

由数值计算的经验可知, 3° 的求解时间约占总解题时间的 50% 以上.

考察结构平衡线性方程组

$$KX = F \tag{1.1}$$

及特征方程组

$$KX = \lambda cX \tag{1.2}$$

其中 K 为结构总刚度矩阵, F 为外荷载向量, c 为质量矩阵等. K 、 c 均为对称稀疏矩阵, 且有相同的稀疏结构, 如图 1 (a) 或图 1 (b).

通常, K 、 c (包括 g 、 h 区域) 取分区的变带宽存储^[1], 用 LU^{[2]、[3]} 或子结构方法^[4] 求解 (1.1), 而 (1.2) 则采用静态凝缩^[5] 与子空间迭代法^[6].

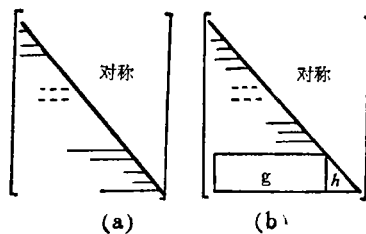


图 1

* 徐次达推荐

分区变带宽技术的引进,使(1.1)、(1.2)的求解运算局限于带区,从而提高了结构分析的效率.因此,若对人工或自动处理所得的结构数据进行网格编码的调整以控制带宽及外形,这对提高大型有限元程序系统的效率是必要的,甚至用波阵法求解的作者也引进了优序的技巧^[7].

从图论导出的一些算法改善了 K 、 c 的带宽及外形.近十年来,始于1968年,先后出现了CM^[8]、RCM^[9]、GPS^[10]、AD^[11]、Rosen^[12]、Grooms^[13]、Rodrigues^[14]、AU^[15]及[2]等有实用价值的算法.

国内由于计算机硬件的限制,有限元大型程序系统的研制仅历时几年,因此,关于网格结点编码优化算法的研究与应用尚欠充分.作者将择述其部份算法,并讨论应用中的有关技术,作为应用,还将给出 RCM 算法的一个有效的执行程序.

二、有关图及层次结构的基本概念

(一) 图

以图 2 (a) 所示的框架结构为例.

设顶点集为 V , 边集 $E \subseteq \{a, b\}$; $a \neq b$ 且 $a, b \in V$, 定义 $G = \langle V, E \rangle$ 为有限无向图.以下简称图. v ($v \in V$) 发出的边数称为该点的分枝数或度. G 标记后称为标记图,如图 2 (b). G 边序列 $\{v_0, v_1\}, \dots, \{v_{i-1}, v_i\}$ 的 i 值称为路径长,而连通图 G 内两个顶点间最短的路径长称为距离,最大的距离称为直径.

标记图的连通性通过相应的邻接矩阵表示,如图 2 (b) 的邻接矩阵为图 2 (c), 记作 B , B 是对称的.其中“ \times ”表示连通,也可标记为“1”,如图 2 (d).

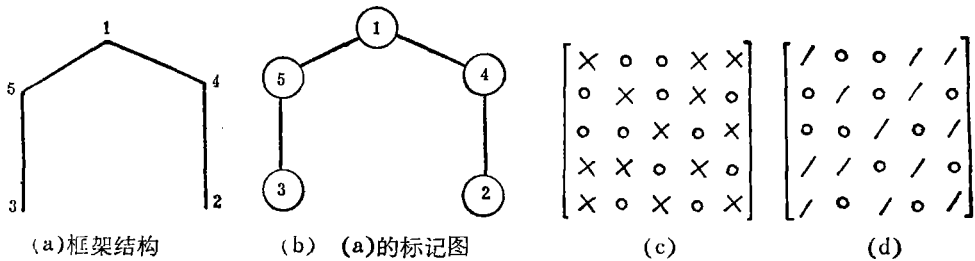


图 2

邻接矩阵的行列交换由初等置换矩阵实现,初等置换矩阵 p 的元素定义为:

$$\begin{cases} p_{kk} = 1, & k \neq i, j \\ p_{ii} = p_{jj} = 1, \\ \text{其余的元素取零,} \end{cases}$$

p 满足以下性质:

$$p = p^T, \quad pp^T = E$$

其中 E 是单位矩阵, T 表示转置.容易验证, K 左乘 p 表示 K 的第 i 行与第 j 行互换, K 右乘 p 表示 K 的第 i 列与第 j 列互换, pKp^T 是同时作这两种变换,对标记图而言,则表示点号 i 与 j 互换.若取一系列初等置换矩阵 p_i , $i=1, 2, \dots, m$, 令 $p = p_1 p_2 \dots p_m$, 且

$$pKp^T = \hat{K}$$

$$pcp^T = c$$

则 (1.1)、(1.2) 等价于

$$\hat{K}(pX) = pF \tag{2.1}$$

$$\hat{K}(pX) = \lambda c(pX) \tag{2.2}$$

令图 2 (a) 的顶点号 2,5 互换, 1,3 互换, 则可得 \hat{G} 及 \hat{B} , 如图 3 (a)、3 (b). 容易算得图 2 的最大带宽 $\beta = 4$, 外形或存储量 $m = 12$, 而图 3 的 $\beta = 1$ 、 $m = 9$. 显见 \hat{B} 是一个好的带阵. 由置换而产生的图 2、图 3 中顶点号的对应关系如下:

原编号 1 2 3 4 5

新编号 3 1 5 2 4

这种将 (2.1)、(2.2) 的 \hat{K} 、 c 排成带形或减缩带宽与外形的方法称为带宽极小化方法, 对应到标记图又称网格结点编码的优化方法.

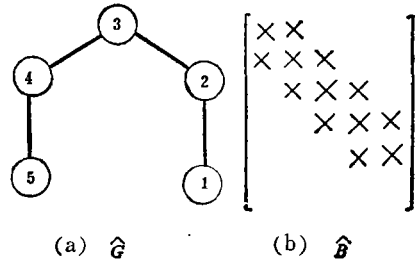


图 3

(二) 层次结构

设图 G 的顶点集 V 按以下原则分割为分层 L_1, L_2, \dots, L_K :

- (a) 相邻于分层 L_1 的所有顶点在 L_1 和 L_2 内;
- (b) 相邻于分层 L_K 的所有顶点在 L_K 或 L_{K-1} 内;
- (c) 对 $1 < i < K$, 相邻于分层 L_i 的所有顶点在 L_{i-1}, L_i 或 L_{i+1} 内.

则 L_i 的全体称为 G 的层次结构, 记作 $L(G)$. 又设顶点 $v \in V$, 且分层定义为:

- (a) $L_1 = \{v\}$;
- (b) 对 $i > 1$, 相邻于分层 L_{i-1} 而未被分配的所有顶点构成分层 L_i .

则所形成的特殊的层次结构 $L_v(G)$ 称基点为 v 的层次结构.

L_i 的元素个数 $W_i(L)$ 称为第 i 分层的宽度, $W(L) = \max\{W_i\}$ 称为 $L(G)$ 的宽度, L 的总层数 K 称为深度.

三、几个优化算法

(一) RCM 算法

Cuthill, Mckee 于 [8] 中提出了一个应用图论概念来重新标记图的顶点以减缩带宽的 CM 算法, 并由 George^[9] 改进成 RCM 算法.

设图 G 连通, 算法步骤为:

(1) 计算图 G 各顶点 i 的度 d_i , 并从中选出低度的顶点 v , 确定低度点 v 有二个直观的方法, 第一个方法取:

$$d_{\min} \leq d_v \leq d_{\min} + \frac{1}{2} d_{\max} \tag{3.1}$$

另一个方法取:

$$d_{\min} \leq d_v \leq \max\{\min\{(d_{\max} + d_{\min})/2, d_{\text{med}} - 1\}, d_{\min}\} \tag{3.2}$$

其中 d_{\min} , d_{\max} , d_{med} 分别表示最小度、最大度、平均度.

(2) 以 v 作基点, 形成 $L_v(G)$.

(3) 用连续正整数标记 $L_v(G)$:

(a) 取基点 v 为第一层.

(b) 对第二层各点按度的增加次序编号为 $2, 3, \dots, i_{n_2}$.

(c) 对第 i 层. 先取 $i=3$, 按度的增加次序将第三层内邻接于点 2 的顶点编号, 号从 $i_{n_2}+1$ 开始, 直至第三层内邻接于 i_{n_2} 的顶点编完号. 以后逐个 i 进行, 待全部顶点编完号.

(4) 对新的标记图 \hat{G} 或相应的邻接矩阵 \hat{B} 计算带宽 β_v .

(5) 比较相邻两次的 β_v 值, 保留 β_v 小的编序.

(6) 按 (3.1)、(3.2) 选择新的 v 作基点, 重复 (2)、(3)、(4)、(5) 直至所有低度的 v 做完.

(1)~(6) 构成了 CM 算法. 一般说来, 经 CM 算法 \hat{G} 的带宽及外形都将得到减缩, 如图 2, 经 CM 算法重新标记后得图 3. 若继续:

(7) 将 CM 算法得到的编序反编号, 即 \hat{G} 的顶点编号 $n-i+1$ 用 i 代换, $i=1, 2, \dots, n$.

(1)~(7) 合起来构成 RCM (Reverse Cuthill-McKee) 算法, 已有人证明在一定条件下 RCM 算法严格地优于 CM 算法^[9].

(二) GPS 算法

Gibbs、Poole 与 Stockmeyer 为了减少 RCM 算法对不同初始点编号及计算带宽的重工作量, 提出了一个改进的新算法, 简称 GPS 算法, 算法由三部份组成:

(1) 寻找图 G 的伪直径端点 u, v , 使 u 与 v 的距离接近直径. 不取直径端点是因为费机时. 直观上看, u, v 的距离较大, 基点为 u, v 的层次结构的深度也较大, 相应的宽度及带宽可能会小. 求取 u, v 的办法是先从度最小的顶点 v 出发建立 L_v , 然后对 L_v 的最后一个分层内的各顶点 s 建立 L_s , 往复迭代, 从中定出深度最大而宽度最小的 L_s , 此 $u(u=s)$ 及 v 即为所求. 一般迭代次数不超过 2, 且有 $K(L_u) = K(L_v)$.

(2) 层宽极小化

算法 (1) 构造了 $L_v = \{L_1, L_2, \dots, L_k\}$ 和 $L_u = \{M_1, M_2, \dots, M_K\}$, 若 L_u 和 L_v 按以下方法重新组合:

(a) 根据 L_v 和 L_u 形成相关层;

(b) 将 G 的全部顶点按相关层的特征分配入新的层次结构 $L = \{N_1, N_2, \dots, N_K\}$.

则 L 能满足:

$$W(L) \leq W(L_v), W(L) \leq W(L_u)$$

(3) 仿照 RCM 算法对 L 逐层编号. 数值效果见^[10].

(三) AD 算法

Akhras, Dhatt 考察了图 4 所示的优序标记图的结点联系表, 归结出表 1 随结点序号 n 增大时有:

(a) 和值增加 (相关点号数相同的),

(b) 平均量增加,

(c) 跨距值增加,

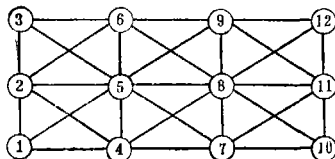


图 4 一个优序标记图

表 1

点 序	相 关 点 号	相	平均量	跨 距
1	1, 2, 4, 5	12	3	6
2	2, 1, 3, 4, 5, 6	21	3.5	7
3	3, 2, 5, 6	16	4	8
4	4, 1, 7, 2, 5, 8	27	4.5	9
5	5, 1, 2, 3, 4, 6, 7, 8, 9	45	5	10
6	6, 2, 3, 5, 8, 9	33	5.5	11
7	7, 4, 5, 8, 10, 11	45	7.5	15
8	8, 4, 5, 6, 7, 9, 10, 11, 12	72	8	16
9	9, 5, 6, 8, 11, 12	51	8.5	17
10	10, 7, 8, 11	36	9	18
11	11, 7, 8, 9, 10, 12	57	9.5	19
12	12, 8, 9, 11	40	10	20

的三条拓扑性质. 若对初始标记图交替地按 (b)、(c) 与 (a)、(b) 调整编号, 则可使 β (或 m) 逐步减小, 由此构成交替迭代的 AD 算法. 由于 AD 算法是一个迭代算法, 一般说来运算时间较 RCM 、 GPS 算法长, 但数值效果有时会超过它们.

(四) Rosen 算法

Rosen 算法用调整标记图的顶点编号, 即交换对应邻接矩阵的行列来减缩带宽. 以图 5 为例, 其基本做法是:

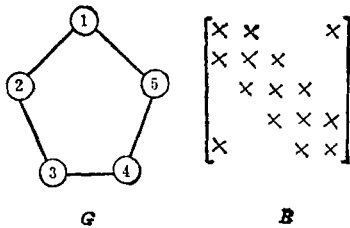


图 5

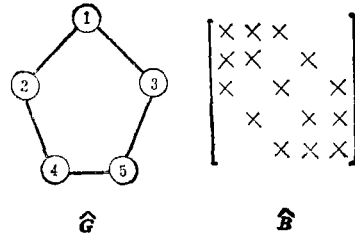


图 6

- (1) 确定 B 的最大带宽及两个达到最大带宽的第一对顶点 (若 $\max \beta_i = \beta_p$, 则 p 及 $p - \beta_p$ 组成这对顶点), 在这对顶点中, 若较高编号的顶点能与一低编号顶点互换以减少带宽, 则转向 (6).
- (2) 若较低编号的顶点能与一高编号顶点互换以减少带宽, 则转向 (6).
- (3) 较高编号的顶点能与一低编号顶点互换, 而带宽保持相同, 则转向 (5).
- (4) 较低编号顶点能与一高编号顶点互换, 而带宽保持相同, 则转向 (5).
- (5) 如果在 (3)、(4) 步已连续执行过规定的经验次数或已交换过的顶点又重新选出来交换, 则算法终止, 否则转向 (6).
- (6) 执行所指出的顶点交换, 并转向 (1).

图 5 经 Rosen 算法得到图 6 所示的优化的标记图 \hat{G} 及邻接矩阵 \hat{B} .

(五) Grooms 算法

Grooms 对 Rosen 算法作了改进，在确定邻接矩阵 B 中达到最大带宽的元素 b_{ij} 后，根据 i 及 j 值，确定：

$$d = [(j-i)/c] \quad (i < j)$$

其中 c 取 2.2， $[\]$ 表示取整。然后将 $(i+1)$ 行到 $(i+d)$ 行各上移一行，第 i 行移至 $(i+d)$ 行位置。同样将 $(j-d)$ 行到 $(j-1)$ 行各下移一行， j 行移至 $(j-d)$ 行位置。然后，各对应列作相应的交换。若 $d=0$ 或交换的遍数超过经验数后终止计算。

(六) Rodrigues 算法

Rodrigues 算法可看作 Rosen 算法的另一种改进。

以图 7 (a) 的框架为例，其邻接矩阵为图 8，并存储其上三角带形，如图 9。 \hat{B} 与 B 的元素间对应的关系为：

$$b_{ij} = \hat{b}_{i, j-i} \quad (j \geq i)$$

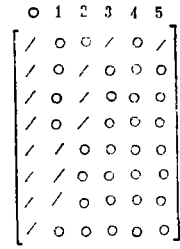
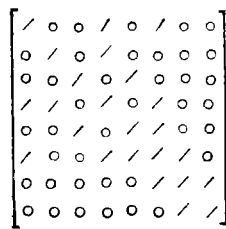
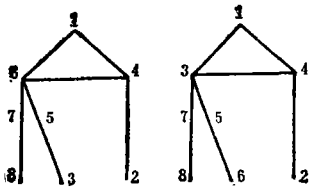


图 7 (a)

图 7 (b)

图 8 图 7(a)的邻接矩阵B

图 9

设 \hat{B} 的 \hat{b}_{ij} 引起最大带宽，容易看出例中顶多有两种不增加带宽的互换使 \hat{b}_{ij} 变成零，第一种交换是行 p 与 $i+j$ 行互换 ($i < p < i+j$)，第二种交换是行 q 与 i 行互换 ($i < q < i+j$)。

- (1) 满足以下条件时执行第一种交换：
 - (a) \hat{b}_{ij} 变为 0；
 - (b) 带宽不增加；
 - (c) \hat{b}_{ij} 经交换尽可能移近第 1 列；
 - (d) 交换后第 j 列不增加新的非零元。

如图 7 (b)。

- (2) 否则，令 $p=0$ ，又若满足条件 (a)~(c) 则实行第二种交换。
- (3) 若 $p=0, q=0$ ，则用 $i-1$ 行与 i 行交换，又若邻接矩阵的 $i-1$ 行带宽缩小，则转 (1)。
- (4) 当带宽或交换的遍数达到规定数值后算法终止。

[14]中给出了数值计算例，表明 Rodrigues 算法优于 Rosen, AU, Grooms 算法。

(七) AU 算法

Akyuz, Utku 用矩阵的行列交换减缩平均带宽 $\bar{\beta}$. 设已知邻接矩阵 B:

(1) 从 B 中选出相邻的两行、列进行交换, 并计算 $\bar{\beta}$, 如果满足下列两个条件之一, 则交换有效:

- (a) $\bar{\beta}$ 减小, 或
- (b) $\bar{\beta}$ 保持相同, 而有较多零元素的行从矩阵中心往外移.

(2) 在一个指定的交换循环内, 按 (1) (a)、(1) (b) 执行行列交换, 其交换顺序规定为: (1,2), (n,n-1), (2,3), (n-1,n-2), ..., 直至中心行.

(3) 如果在一个循环内没有发生交换, 或经过经验的 $3+n/100$ 次循环, 而 $\bar{\beta}$ 不减小时, 交换运算停止, 否则重复执行 (1)、(2).

四、算法的应用技术

分析三的几个算法, CM、RCM、GPS 算法只执行有限次的标记运算, 具有直接法的特点, 而 Rosen、AU、AD、Grooms、Rodrigues 算法则具有迭代法的特点. 从算法的机时来看, 直接法较为有利. 就优化效果而论, 由于目前还没有严密的数学讨论, 因而不能给出准确的评价, 需通过数值检验, 许多作者作过一些工作^{[10]、[13]、[14]}. 通常认为算法对粗糙的初始编序效果明显, 而精致的编序, 效果可能不明显. 目前从应用来看, 八个算法中较受推荐的是 RCM、GPS、AD 算法.

Rosen^[12] 于 1968 年提出了第一个有实用价值的优化算法, 文中给出了 fortran 程序, 同年出现了 AU 算法程序. 国内较早的工作有 [2], 该文附有程序, [16]、[17] 也对几个算法的程序实现作了尝试.

程序设计中, 标记图顶点的交换一般是通过邻接矩阵 B 的行列交换来实现的. 因此, 对高阶的 B 必须有效地解决其存储与行列交换的方法.

(一) B 的存储方法

第一种存储方法. 开设二维数组 $A[1:n, 1:c]$ 存储 B, 以计算机字长的一位作一个存储单位, 存放 B 的一个元素, 其中 n 为 B 的阶, $c = [(n-1)/WL] + 1$, WL 表示机器字长. 以图 2 为例, A 取表 2:

表 2

n \ A	A[1:5, 1:1]			
1	46	00000	00	00000
2	24	00000	00	00000
3	12	00000	00	00000
4	64	00000	00	00000
5	52	00000	00	00000

第二种存储方法. 开设一维数组 $AI[1:R]$, $AJ[1:R]$, 分别存储对角元以外的 B 矩阵

上(下)三角部份非零元的行标与列标, 其中 R 为非零元个数. 以图 5 为例, AI 、 AJ 取表 3:

表 3

AI	1 1 2 3 4
AJ	2 5 3 4 5

第三种存储方法. 开设二维数组 $A[1:n, 1:c]$, 在 A 中按行存放 B 矩阵对角元外的非零元列标, 而空余分量补零, c 表示 B 矩阵各行中除对角元外非零元个数的最大值.

第四种存储方法见 Rodrigues 算法.

除了上述四种存储方法外, 还可设计出其它的存储方法. 应用这些方法时, 需根据算法对运算速度与存储量的要求而加以选择. 前二种方法存储量节省, 但执行行列交换的速度较慢, 而后二种则反之.

(二) B 的行列交换

各优化算法均涉及 B 的行列交换, 交换公式为 (2.1)、(2.2). 设初始标记图 G 的顶点编号用数组 $VL[1:n]$ 的分量表示, VL 称顶点表. 计算中, 某二个顶点编号的交换对应为 VL 及 B 的相应分量的交换, 这种交换对不同的算法应采用不同的技巧, 以提高计算效率. 下一节中的 RCM 算法程序就是一例. 编号过程结束后, VL 的终态就是以初始编序表示的最终编序, 即原编号 i 对应于新编号 $VL[i]$.

五、RCM 算法的程序 BWMIN

作为应用实例, 以下将给出 RCM 算法的程序 BWMIN 及说明:

(一) 功能

本程序用 RCM 算法对已知标记图 G 重新编序, 得到新标记图 \hat{G} , \hat{G} 相应的邻接矩阵 \hat{B} 呈带形, 且带宽减缩. 邻接矩阵采用第三种存储方法, 而 G 的重新标记避开邻接矩阵行列元素的繁复互换, 计算是高效的.

(二) 使用说明

(1) 语句形式

BWMIN($n, no, mb, ms, a, b, d, p$);

其中:

- n 图 G 的顶点数;
- no G 中低度基点的个数, 一般根据 (3.1)、(3.2) 确定, 由于 no 过大会导致运算时间太长, 应用时可视 no 的大小再行调整;
- $a[0:n, 1:m]$ 按行存放 B 矩阵对角元外非零元列标, 其余分量补零, m 表示 B 矩阵各行中对角元外非零元个数的最大值, 且规定 $A[0, 1] = \dots = A[0, m] = 0$;

- b[1:no]** 低度基点的编号;
d[0:n] 规定 $d[0]=1, d[i]$ 存放图 G 中顶点 i 的度;
p[1:n] 过程执行的终态在 $p[i]$ 中给出相应于原编号 i 的新编号;
mb 新标记图的最大带宽;
ms 新标记图的存储量.

(2) 算例

以图 2 为例, BWMIN 的形式参数取值如下:

$n=5, m=2, no=2, b[1]=2, b[2]=3$, 以及表 4.

表 4

i	A	d
0	0, 0	1
1	4, 5	2
2	4, 0	1
3	5, 0	1
4	1, 2	2
5	1, 3	2

计算给出 $p[1:5]$:

3	1	5	2	4
---	---	---	---	---

(3) BWMIN 的计算效率. 以 1600 个顶点的标记图为例, 设最大度为 6, 取二个基点, 用 BWMIN 程序计算约 40 秒 (709 机),

(三) 程序

```

PROCEDURE BWMIN( $n, no, mb, ms, a, b, d, p$ ); VALUE  $n, no$ ;
  REAL  $n, no, mb, ms$ ; ARRAY  $a, b, d, p$ ;
  BEGIN REAL  $i, j, q, r, s, t, ia, id$ ; ARRAY  $pn, qn[1:100]$ ;
    FOR  $i:=1$  STEP 1 UNTIL  $n$  DO
      BEGIN  $id:=d[i]$ ;
        FOR  $j:=1$  STEP 1 UNTIL  $id-1$  DO
          BEGIN  $q:=A[i, j]$ ;  $r:=d[q]$ ;  $s:=j$ ;
            FOR  $ia:=j+1$  STEP 1 UNTIL  $id$  DO
              BEGIN  $t:=d[A[i, ia]]$ ;
                IF  $t<r$  THEN BEGIN  $r:=t$ ;  $s:=ia$  END
              END;
               $A[i, j]:=A[i, s]$ ;  $A[i, s]:=q$ 
            END
          END;
           $A[i, j]:=A[i, s]$ ;  $A[i, s]:=q$ 
        END
      END;
       $mB:=_{10}4$ ; #write drum(0,0,A,d);
      FOR  $id:=1$  STEP 1 UNTIL  $no$  DO
        BEGIN #read drum(0,0,A,d);  $s:=qn[1]:=0$ ;  $ia:=1$ ;  $A[0,1]:=B[id]$ ;
  
```

```

bm1: FOR i:=1 STEP 1 UNTIL ia DO pn[i]:=qn[i]; t:=0;
FOR i:=1 STEP 1 UNTIL ia DO
FOR j:=1 STEP 1 UNTIL d[pn[i]] DO
BEGIN q:=#abs(A[pn[i],j]);
IF A[q,1]<0 THEN GOTO Bm2 ELSE
BEGIN t:=t+1; qn[t]:=q; s:=s+1; p[q]:=s; A[q,1]:=-A[q,1] END;
bm2:END;
ia:=t; IF s<n THEN GOTO bm1; t:=ia:=0; #read, drum(0,0,A);
FOR i:=1 STEP 1 UNTIL n DO
BEGIN s:=10A;
FOR j:=1 STEP 1 UNTIL d[i] DO IF p[A[i,j]]<s THEN s:=p[A[i,j]];
q:=p[i]; r:=IF q<=s THEN 0 ELSE q-s; ia:=ia+r+1;
IF r>t THEN t:=r
END;
IF t<mb THEN
BEGIN mb:=t; ms:=ia; FOR j:=1 STEP 1 UNTIL n DO p[j]:=n+1-p[j];
#write drum(0,4000,p)
END
END;
#read drum(0,4000,p)
END BWMIN;

```

参 考 文 献

1. Jennings, A., A compact storage scheme for the solution of symmetric linear simultaneous equations, *Compt. J.* 9. (1966), 281-285.
2. 曹志浩, 大型线性方程集的直接解法, 复旦学报, (1974.1).
3. 徐次达、华伯浩、王远功, 薄壳有限元混合法与求解程序, 计算技术通讯, (1978.3).
4. J. S. 普齐米尼斯基, 《矩阵结构分析理论》, 国防工业出版社, (1974).
5. 结构和连续力学中的有限单元法, 上海交通大学《有限单元法》翻译组.
6. Wilkinson, J. H., Reinsch, C., *Linear Algebra* (1971).
7. 唐伟枫, 有限元分析的分批波阵法程序, 全国机械学会有限元会议, (1978.12).
8. Cuthill, E. and Mckee, J., Reducing the bandwidth of sparse symmetric matrices, *Proc. 24th Nat Conf. of the ACM, ACM Publ*, 69, *Association for Computing Machinery*, New york, (1969), 157-172.
9. Liu Wai-hung and Sherman, A. H., Comparative analysis of the Cuthill-Mckee and the reverse Cuthill-Mckee ordering algorithms for sparse matrices, *SIAM J. Numer. Anal*, 13,2(1976), 198-213.
10. Gibbs, N. E., Poole, Jr., W. G. and Stockmeyer, Paul K. An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM J. Numer Anal*, 13, 2(1976), 236-250.
11. Akhras, G. and Dhatt, G., An automatic node relabeling scheme for minimizing a matrix or network bandwidth, *Int. J. Num. Math. Engng*, 10, 4,(1976), 787-797.
12. Rosen, R., Matrix bandwidth minimization, *Proc. 23rd Nat. Conf. ACM Publ*, 68, 585-595. *Brandon Systems Press*, Princeton, New Jersey, (1968).
13. Grooms, H. R., Algorithm for matrix bandwidth reduction, *ASCE, J. Struct. Div.* 98, ST1. (1972), 203-214.
14. Rodrigues, J. S., Node numbering optimization in structural analysis, *Proc. ASCE, Jour. Struc. Div.* 101, ST 2, (1975), 361-376.
15. Akyuz, F. A. and Utku, S., An automatic relabeling scheme for bandwidth minimization of stiffness matrices, *J. Amer. Inst. Aeronaut. Astronaut*, 6,(1968),728-730.
16. 弹性薄壳有限单元混合法强度分析, 上海市计算技术研究所, 上海同济大学工程力学研究室等合编 (1976.8).
17. 大型稀疏线性方程组的解法, 上海计算技术研究所, (1978).(待出版)

The Network Node Relabeling Optimum Algorithms for Large Finite Element Program System

Hua Bo-hao

(Shanghai Institute of Computer Technology, Shanghai)

Abstract

This paper describes the network node relabelling optimum algorithms in the recent decade, relating with the large finite elements program system. Some computation technics of applied algorithm are discussed, and a more useful program of algorithm RCM is given for application.