

文章编号: 1009-3087(2014)04-0067-08

## 微分代数系统中分块快速指标约简的启发式算法

唐 卷<sup>1,2,3</sup> 杨文强<sup>1</sup> 吴文渊<sup>1</sup> 冯 勇<sup>1\*</sup>

(1. 中国科学院 重庆绿色智能技术研究院 自动推理与认知重庆市重点实验室, 重庆 400714;  
2. 中国科学院 成都计算机应用研究所, 四川 成都 610041; 3. 中国科学院大学, 北京 100049)

**摘要:** 为了高效地优化多领域统一建模仿真软件中稀疏微分代数方程的指标约简技术, 提出了一种分块快速指标约简的启发式算法。它的主要思想是将微分代数方程的符号矩阵进行块状上三角化, 然后沿每一独立块状上三角化的子符号矩阵对角线方向, 由上往下依次对其每一对子方阵运用含参数 Pryce 方法进行指标约简。大量数值实验结果表明, 分块快速指标约简方法可以求出微分代数方程的最优偏移向量, 其效率明显优于 Pryce 方法的效率, 效率提高倍数与对角子方阵块数成正相关。进一步以数控冲床为例, 对其微分代数方程采用分块快速指标约简算法进行指标约简, 验证了该方法高效地解决了较大规模高指标微分代数方程的指标约简问题, 便捷地获得了该系统的最优偏移向量等结构信息。

**关键词:** 多领域统一建模; 微分代数方程; 指标约简; Pryce 方法; 分块快速指标约简方法; 数控冲床  
**中图分类号:** TP301 **文献标志码:** A

### A Heuristic Algorithm for Block Fast Index Reduction in Differential Algebraic Systems

TANG Juan<sup>1,2,3</sup>, YANG Wenqiang<sup>1</sup>, WU Wenyuan<sup>1</sup>, FENG Yong<sup>1\*</sup>

(1. Chongqing Key Lab. of Automated Reasoning and Cognition, CIGIT, CAS, Chongqing 400714, China;  
2. Chengdu Inst. of Computer Applications, CAS, Chengdu 610041, China;  
3. Univ. of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** In order to optimize the index reduction technology of sparse differential algebraic equations (DAEs) from multi-domain unified modeling software, a heuristic algorithm for block fast index reduction method (BFIRM) was proposed. The main principle of BFIRM is to do block triangularization of DAEs' signature matrix, then to use Pryce method with parameters to process diagonal sub-square-matrix in block triangulated signature matrix from top to bottom in sequence. A large number of numerical experiment results showed that BFIRM can solve the canonical offsets vectors of sparse DAEs efficiently. And the reduction efficiency positively correlates with the number of the diagonal blocks. Moreover, to take CNC punch machine tool as an example, its DAEs was reduced by the BFIRM. This method was verified effectively in reducing the index of large-scale DAEs with high index and easily got DAEs' structural information, such as canonical offsets vectors.

**Key words:** multi-domain unified modeling; differential algebraic equations; index reduction; Pryce method; block fast index reduction method; CNC punching machine tool

基于工程实践的发展和需要, 复杂产品的建模与仿真技术正朝着多领域统一建模的方向发展。基于对象的建模方式使建模变得相对简单, 建模时间也大为缩短, 代码复用性得以提高, 因此推出了一些基于

对象的建模仿真语言, 比如 Modelica 语言<sup>[1]</sup>。基于 Modelica 语言陈述式非因果建模的特性, 复杂连续仿真模型系统最终转换为一个大规模高指标的微分代数方程(differential algebraic equations, DAEs)系统。

收稿日期: 2013-11-14

基金项目: 国家“973”计划资助项目(2011CB302402); 国家自然科学基金重大研究计划项目(91118001); 重庆市科技攻关计划项目(cstc2012ggB40004)

作者简介: 唐 卷(1987—), 男, 博士生, 研究方向: 微分代数方程指标约简及其数值方法。E-mail: tangjuan0822@gmail.com

\* 通信联系人 E-mail: yongfeng@cigit.ac.cn

http://jsuese.scu.edu.cn

DAEs 的微分指标是为了将其转化为常微分方程组(ordinary differential equation, ODE)而需要对其部分或全部方程求微分的最小次数<sup>[2]</sup>。与数值积分不同,在步长很小时,数值微分的误差与步长的减小成反比,因而无法通过减小步长来控制误差,常常由于不收敛而造成求解失败<sup>[3]</sup>。因此需要将高指标 DAEs 系统转换为低指标系统,避免高指标微分数值运算;从而在高指标系统的数值求解中,指标约简方法变得十分关键。

目前,针对一般 DAEs 系统,结构指标约简方法是指标约简方法中非常重要的一类方法。它主要有 Gear 方法<sup>[4]</sup>、Pantelides 方法<sup>[5]</sup>、Pryce 方法<sup>[6]</sup>、负权二部图方法<sup>[7]</sup>和加权二部图指标转换方法<sup>[8]</sup>。

Gear 方法是一种符号操作的指标约简方法,但对一般非线性系统,因微分了一些不必要的代数方程,它的时间复杂度很高。

Pantelides 方法是一种基于最小奇异结构子集的指标约简方法,由于相对比较低的时间复杂度,被广泛应用于 Dymola 和 MapleSim 等专业软件。但是该方法不能直接处理高阶系统,也不易实现块状化处理。

Pryce 方法是基于系统符号矩阵(signature matrix)的结构指标约简方法。该方法能够直接获取系统最优偏移向量,其时间复杂度也相对较低,但没有严格证明。

针对 Pantelides 方法不能直接处理高阶系统的问题,负权二部图方法和加权二部图指标转换方法都通过赋权二部图实现高阶 DAEs 的指标约简,它们基本上是同一种方法,文献[9]中进一步说明负权二部图方法与 Pantelides 方法具有相同的时间复杂度。在一定意义上,负权二部图方法和加权二部图指标转换方法都可以看成 Pryce 方法在二部图上的表现,但它们也不便于实现块状化处理。

针对存在块状三角化结构的稀疏高阶 DAEs, Pryce 等近期在文献[10]中提出了指标分块处理的技术,该方法作为软件包 DAEsSA 已植入到 Matlab 工具包中。但他们有时只能得到有效偏移向量(valid offsets vectors),而不能获取最优偏移向量(canonical offsets vectors)(见后面例 1)。

为了解决目前结构指标约简方法中存在的问题,作者提出了分块快速指标约简的启发式算法,力求高效优化地解决多物理场建模软件中容易产生大规模稀疏 DAEs(尤其是高指标、高阶 DAEs)系统的指标约简问题。

## 1 分块快速指标约简方法

### 1.1 Pryce 方法

对于任意给定的 DAEs 系统

$$f = (f_1, f_2, \dots, f_n) = 0 \quad (1)$$

式中,  $f_i = f_i(x_1(t), \dots, x_n(t))$ ,  $x_i(t)$  为式(1)中相互独立的  $n$  个函数变量,其符号矩阵(signature matrix)  $\Sigma$  定义为:

$$\Sigma = [\sigma_{ij}] \quad (2)$$

式中,  $\sigma_{ij}$  为  $x_j(t)$  在方程  $f_i$  中的最高微分次数,如果没有出现就为  $-\infty$ 。

定义 1<sup>[6]</sup> 指派问题的原问题为:

$$\begin{aligned} \text{Max } z &= \sum_{(i,j) \in S} \sigma_{ij} \eta_{ij} \\ \text{s. t. } &\begin{cases} \sum_{j \in \{l \mid (i,j) \in S\}} \eta_{ij} = 1, \quad i = 1, 2, \dots, n; \\ \sum_{i \in \{l \mid (i,j) \in S\}} \eta_{ij} = 1, \quad j = 1, 2, \dots, n; \\ \eta_{ij} \geq 0, \quad (i,j) \in S \end{cases} \end{aligned} \quad (3)$$

式中,  $S = \{(i,j) \mid \sigma_{ij} > -\infty, i = 1, \dots, n, j = 1, \dots, n\}$ 。

定义 2 集合  $T$  是  $S$  的一个横截面,如果由  $S$  中  $n$  个不同元素构成的子集  $T$  中每一元素都来自不同行和不同列,即  $T$  中任意 2 个元素既不在同一行也不在同一列。进一步,如果  $\|T\| = \sum_{(i,j) \in T} \sigma_{ij}$  是  $S$  所有横截面中的最大值,则称  $T$  是  $S$  的最大值横截面。

定义 3<sup>[6]</sup> 指派问题的对偶问题为:

$$\begin{aligned} \text{Min } \bar{z} &= \sum_{j=1}^n d_j - \sum_{i=1}^n c_i \\ \text{s. t. } &\begin{cases} d_j - c_i \geq \sigma_{ij}, \quad (i,j) \in S; \\ c_i \geq 0, \quad i = 1, 2, \dots, n \end{cases} \end{aligned} \quad (4)$$

Pryce 方法<sup>[6]</sup>的主要步骤为:

- 1) 生成对应于系统的符号矩阵  $\Sigma$ ;
- 2) 通过求解指派问题找到符号矩阵的最大值横截面  $T$ ;
- 3) 确定 DAE 系统符号矩阵的最优偏移(canonical offsets) 向量问题,即方程偏移向量  $C = (c_i)_{1 \leq i \leq n}$  和变量偏移向量  $D = (d_j)_{1 \leq j \leq n}$  的求解问题:根据最大值横截面  $T$  和符号矩阵  $\Sigma$ ,利用对偶变量的不动点迭代算法<sup>[6]</sup>求解  $C$  和  $D$ ;
- 4) 生成 DAEs 系统对应的 Jacobian 矩阵  $J$ ,当  $f_i$  方程中有  $x_j(t)$  的微分时,

$$J_{ij} = \frac{\partial f_i}{\partial^{(d_j - c_i)} x_j(t)} \quad (5)$$

其余都为 0;

5) 如果矩阵  $J$  非奇异,系统的结构指标定义如下:

$$\nu = \max_i c_i + \begin{cases} 0, & \text{如果所有 } d_j > 0; \\ 1, & \text{如果存在 } d_j = 0 \end{cases} \quad (6)$$

Pryce 方法主要是通过 Pryce 指标约简算法 (pryce index reduction algorithm, PIRA) 来实现。PIRA 是利用不动点迭代的思想来实现指标约简。文献 [6] 中没有严格证明,从而它是一种启发式算法,具体步骤如下。

算法 1 Pryce 指标约简算法 (PIRA)

输入:  $n (> 1)$  阶  $\Sigma$  矩阵  $M$ 。

输出: 偏移向量  $C_1$  和  $D_1$ 。

1) 初始化:  $C_0, C_1, D_0$  和  $D_1$  为  $n$  维零向量,最大值横截面集合  $T$  为  $\emptyset$ , 矩阵  $CM = M$ , 最大迭代次数  $M\_Itera = 100\ 000$ , 实际迭代次数  $A\_Itera = 0$ 。

2) 调整变量偏移值  $D_1$ : 取  $D_1$  为:  $D_1(j) =$

$$\max_{i \in \{1, 2, \dots, n\}} M_{ij}, j = 1, 2, \dots, n。$$

3) 求最大值横截面  $T$ : 通过使用标准算法匈牙利方法,找出  $M$  的最大值横截面集合  $T$ 。

4) 更新方程偏移向量  $C_1$ : 依次取  $i \in \{1, 2, \dots, n\}$ , 必有唯一的  $j$  使得  $(i, j) \in T$ , 更新  $C_1(i) = D_1(j) - M(i, j)$ 。

5) 判断算法是否结束: 如果  $C_0 = C_1$  或  $A\_Itera > M\_Itera$  输出  $C_1$  和  $D_1$ , 算法结束; 否则  $A\_Itera = A\_Itera + 1$ ,  $C_0$  更新为  $C_1$ , 转下一步。

6) 更新变量偏移向量  $D_1$ : 先更新符号矩阵  $CM$ , 对  $i = 1, 2, \dots, n$  如果  $C_1(i) > 0$ , 取  $CM(i, j) = M(i, j) + C_1(i)$ ,  $j = 1, 2, \dots, n$ ; 然后更新  $D_1$  为  $D_1(j) = \max_{i \in \{1, 2, \dots, n\}} CM_{ij}$ ,  $j = 1, 2, \dots, n$ ; 再转 4)。

### 1.2 块状上三角化处理

为了实现非结构奇异的大规模稀疏 DAEs 系统的指标块状化约简技术,首先对它的  $\Sigma$  矩阵  $M$  进行块状化处理。

定义 4 图  $G(M) = (F, Y; E)$  是对应  $M$  的无向二部图,如果  $F$  和  $Y$  分别为方程和变量节点集合,且边集合为:

$$E = \left\{ (F_i, Y_j) \mid \begin{matrix} F_i \in F, Y_j \in Y, M_{ij} \geq 0, \\ i = 1, 2, \dots, n, j = 1, 2, \dots, n \end{matrix} \right\} \quad (7)$$

定义 5 集合  $P$  是二部图  $G(M)$  中  $E$  的任一子集,称  $P$  为  $G(M)$  的匹配集合,如果  $P$  中的任意 2 个元素没有公共的节点属于  $F$  或  $Y$ 。进一步,如果  $P$  中

元素的个数不仅等于  $F$  的个数,而且也等于  $Y$  的个数,则称  $P$  为  $G(M)$  的完美匹配。

$P$  中每一元素称为  $G(M)$  的一条匹配边  $E - P$  中元素称为未匹配边。如果在  $G(M)$  中存在完美匹配  $P$ , 则有向图  $G_d(M)$  按如下规则生成:  $G(M)$  中每一未匹配的边的方向从  $F$  节点指向  $Y$  节点,然后  $P$  中每一条匹配边从  $F$  节点收缩到  $Y$  节点。

稀疏  $\Sigma$  矩阵  $M$  块状三角化处理<sup>[11]</sup> 的具体步骤:

1) 根据  $G(M)$  中的连通分支关系寻找对应于  $M$  中独立的对角块状矩阵。假定该矩阵含有  $l$  个子矩阵  $M_1, M_2, \dots, M_l$ , 可转化为:

$$M \rightarrow \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_l \end{bmatrix}。$$

其中,空白处的元素全为  $-\infty$ 。

2) 对每一个子方阵  $M_i$ , 在  $G(M_i)$  中寻找完美匹配; 根据完美匹配,将  $G(M_i)$  转化为有向图  $G_d(M_i)$ 。

3) 寻找  $G_d(M_i)$  的强连通分支并排序,从而可将方阵  $M_i$  块状上三角化为  $n_i$  块,即

$$M_i \rightarrow \begin{bmatrix} M_{i,1} & X & \cdots & X \\ & M_{i,2} & \cdots & X \\ & & \ddots & \vdots \\ & & & M_{i,n_i} \end{bmatrix}。$$

其中,空白处的元素全为  $-\infty$ , 矩阵  $X$  表示其中可能存在非负元。

### 1.3 分块快速指标约简方法

分块快速指标约简方法通过将 Pryce 方法改进为含参数 Pryce 指标约简方法,并在此方法的基础上提出了启发式的分块快速指标约简算法 (block fast index reduction algorithm, BFIRA)。BFIRA 的基本思想是: 对于耦合的块状上三角结构的  $\Sigma$  矩阵,按对角块的顺序由上往下依次运用含参数 Pryce 指标约简算法 (pryce index reduction algorithm with parameter, PPIRA), 从而逐渐从局部最优偏移向量扩展到  $\Sigma$  矩阵的整体最优偏移向量,它的具体步骤如下。

算法 2 分块快速指标约简算法 (BFIRA)

输入:  $n$  阶  $q$  块上三角结构的  $\Sigma$  矩阵

$$M = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1q} \\ & M_{22} & \cdots & M_{2q} \\ & & \ddots & \vdots \\ & & & M_{qq} \end{bmatrix}_{n \times n}$$

和子方阵阶数的  $q$  维向量  $[n_1 \ n_2 \ \dots \ n_q]_{1 \times q}$  (满足  $n = \sum_{j=1}^q n_j$ );

输出: 偏移值向量  $C$  和  $D$ 。

1) 初始化: 取  $n$  维  $q$  块参数向量为  $MD = [MD_1 \ MD_2 \ \dots \ MD_q]_{1 \times q}$ , 其中子向量  $MD_j = [0 \ \rho \ \dots \ \rho]_{1 \times n_j}$ , 对应的方程偏移值向量  $C = [C_1 \ C_2 \ \dots \ C_q]_{1 \times q}$ ,  $C_j = [0 \ \rho \ \dots \ \rho]_{1 \times n_j}$  和变量偏移值向量  $D = [D_1 \ D_2 \ \dots \ D_q]_{1 \times q}$ ,  $D_j = [0 \ \rho \ \dots \ \rho]_{1 \times n_j}$ ,  $j = 1 \ 2 \ \dots \ q$ ; 并取子块数  $i = 1$ 。

2) 运用 PPIRA, 获取偏移向量  $C_i$  和  $D_i$ :  $[C_i \ D_i] = \text{PPRIA}(M_i \ MD_i)$ , 步骤如算法 3 所示。

3) 判断算法是否结束: 如果  $i = q$ , 输出  $C$  和  $D$ , 结束; 否则  $i = i + 1$ , 转下一步。

4) 非对角矩阵  $[M_{i-1 \ j} \ M_{i-1 \ j+1} \ \dots \ M_{i-1 \ q}]$  更新: 对  $k = 1 \ 2 \ \dots \ n_{i-1}$ , 如果  $C_{i-1}(k) > 0$ , 取

$$M_{i-1 \ j}(k \ l) = M_{i-1 \ j}(k \ l) + C_{i-1}(k)。$$

其中,  $l = 1 \ 2 \ \dots \ n_j$ ,  $j = i \ i + 1 \ \dots \ q$ 。

5) 参数向量  $[MD_i \ MD_{i+1} \ \dots \ MD_q]$  更新: 先获取更新后  $[M_{i-1 \ j} \ M_{i-1 \ j+1} \ \dots \ M_{i-1 \ q}]$  矩阵中各列最大值的向量, 然后取它与原参数向量  $[MD_i \ MD_{i+1} \ \dots \ MD_q]$  中对应元素最大值的向量更新为  $[MD_i \ MD_{i+1} \ \dots \ MD_q]$ , 转步骤 2)。

算法 2 在 2) 中调用了 PPIRA, 它的具体步骤如下。

算法 3 含参数 Pryce 指标约简算法 (PPIRA)

输入:  $n (> 1)$  阶  $\Sigma$  矩阵  $M$  和需要满足限制的  $n$  维参数向量  $MD$ ;

输出: 偏移向量  $C_1$  和  $D_1$ 。

1) 初始化:  $C_0, C_1, D_0$  和  $D_1$  为  $n$  维零向量, 最大值横截面集合  $T$  为  $\emptyset$ , 矩阵  $CM = M$ , 最大迭代次数  $M\_Itera = 100\ 000$ , 实际迭代次数  $A\_Itera = 0$ 。

2) 调整变量偏移值  $D_1$ : 如果没有输入参数向量  $MD$ , 调整  $D_1$  为:

$$D_1(j) = \max_{i \in \{1 \ 2 \ \dots \ n\}} M_{ij}, \quad j = 1 \ 2 \ \dots \ n;$$

否则, 先取  $D_0$  为:

$$D_0(j) = \max_{i \in \{1 \ 2 \ \dots \ n\}} M_{ij}, \quad j = 1 \ 2 \ \dots \ n,$$

再调整  $D_1$  为

$$D_1(j) = \max\{D_0(j), MD(j)\}, \quad j = 1 \ 2 \ \dots \ n。$$

3) 求最大值横截面  $T$ : 通过使用标准算法匈牙利方法, 找出  $M$  的最大值横截面集合  $T$ 。

4) 更新方程偏移向量  $C_1$ : 依次取  $i \in \{1 \ 2 \ \dots \ n\}$ , 必有唯一的  $j$  使得  $(i \ j) \in T$ , 更新  $C_1(i) =$

$D_1(j) - M(i \ j)$ 。

5) 判断算法是否结束: 如果  $C_0 = C_1$  或  $A\_Itera > M\_Itera$ , 输出  $C_1$  和  $D_1$ , 算法结束; 否则  $A\_Itera = A\_Itera + 1$ ,  $C_0$  更新为  $C_1$ , 转下一步。

6) 更新变量偏移向量  $D_1$ : 先更新符号矩阵  $CM$ , 对  $i = 1 \ 2 \ \dots \ n$ , 如果  $C_1(i) > 0$ , 取  $CM(i \ j) = M(i \ j) + C_1(i)$ ,  $j = 1 \ 2 \ \dots \ n$ ; 然后更新  $D_1$  为  $D_1(j) = \max_{i \in \{1 \ 2 \ \dots \ n\}} CM_{ij}$ ,  $j = 1 \ 2 \ \dots \ n$ ; 再转 4)。

例 1 假设微分代数方程组如下:

$$\begin{cases} f_1 = x_1'' + x_3; \\ f_2 = x_2' + x_3; \\ f_3 = x_1 + x_2 + x_6; \\ f_4 = x_4'' + x_6; \\ f_5 = x_5' + x_6; \\ f_6 = x_4 + x_5; \\ f_i = 0 \quad i = 1 \ 2 \ \dots \ 6 \end{cases} \quad (8)$$

式中,  $(x_1 \ x_2 \ \dots \ x_6) = (x_1(t) \ x_2(t) \ \dots \ x_6(t))$ 。

对应于 DAEs 系统的  $\Sigma$  矩阵为:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$\hat{c}_i$	$c_i$
$f_1$	2		0				0	0
$f_2$		1	0				0	0
$f_3$	0	0				1	1	1
$f_4$				2		0	0	1
$f_5$					1	0	0	2
$f_6$				0	0		1	3
$\hat{d}_j$	2	1	0	2	1	0		
$d_j$	2	1	0	3	3	2		

上述  $c_i$  和  $d_j$  是对整个  $\Sigma$  矩阵使用 Pryce 方法得到的最优偏移值;  $\hat{c}_i$  和  $\hat{d}_j$  是对每一子方阵使用 Pryce 方法得到的局部最优偏移值; 进一步可以知道该系统是指标为 4 的高指标系统。BFIRA 的具体过程如下。

首先取第一个对角子方阵, 参数向量  $MD_1$  为  $(0 \ 0 \ 0)$  (如图 1), 运用 PPIRA。在该算法中运用匈牙利算法得到最大值横截面  $T = \{(1 \ 1), (2 \ 3), (3 \ 2)\}$  (图 1 中“•”标记处)。

从图 1 可以看出实际迭代次数  $A\_Itera = 2$ , 第一次运用 PPIRA 结束, 其偏移向量为  $C_1 = (0 \ 0 \ 1)$  和  $D_1 = (2 \ 1 \ 0)$ 。

然后取第 2 个对角子方阵, 参数向量  $MD_2$  为  $(0 \ 0 \ 2)$ , 调用 PPIRA。当  $A\_Itera = 1$  时,  $C_2$  变为  $(0 \ 2 \ 1)$ ,  $D_2$  变为  $(2 \ 1 \ 2)$ ; 当  $A\_Itera = 2$ ,  $C_2$  变为  $(0 \ 2 \ 1)$ ,  $D_2$  变为  $(2 \ 1 \ 2)$ 。

2 3)  $D_2$  变为(2 3 2);当  $A\_Itera = 3$   $C_2$  变为(1, 2 3)  $D_2$  变为(3 3 2);当  $A\_Itera = 4$   $C_2$  和  $D_2$  保持不变,则 PPIRA 结束,其偏移向量  $C_2 = (1 2 3)$  和  $D_2 = (3 3 2)$ 。

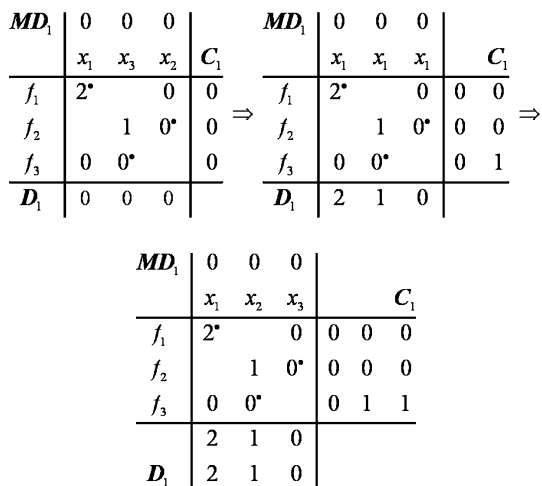


图 1 第一个对角子方阵运算流程图

Fig.1 First diagonal matrix reduction flowchart

因  $\Sigma$  矩阵有 2 个子块,从而分块快速指标约简算法结束,可以得到  $\Sigma$  矩阵的整体偏移值向量  $C = (C_1, C_2) = (0 0, 1, 1 2 3)$  和  $D = (D_1, D_2) = (2, 1 0 3 3 2)$ 。

这个例子中第 2 个子方阵的局部偏移向量  $\hat{C}_2 = (0 0, 1)$  和与之对应整体部分偏移向量  $C_2 = (1 2, 3)$  之间不存在一个固定的常数,与文献[10]中定理 4.4 的结论不一致。如果整体部分偏移向量和局部偏移向量之间如果存在一个常数,则这个常数至少为 2,即  $C_2' = (2 2 3) > C_2$  (其中  $C' > C$  如果  $c_i' \geq c_i, i = 1 2, \dots, n$ ),这说明 Pryce 块状化约简方法<sup>[10]</sup>此时只能获取 DAEs 系统的有效偏移向量,而分块快速指标约简方法可以准确地获取最优偏移向量。

1.4 数值实验及分析

对于大规模稀疏 DAE 系统,为了提高数值计算的效率,一般先对系统进行块状三角化处理,然后再从下往上或从上往下不断地进行数值迭代求解,从而系统方程的块状三角化是进行大规模数值计算必要的预处理技术。在数值实验中假定系统对应的符号矩阵已利用上文中块状三角化技术变换成相互对立的耦合块状上三角符号矩阵。对于问题(4),所有约束方程的系数为 1 或 -1,且符号矩阵中所有元素都为整数,则问题(4)约束区域的所有顶点都为整数向量,从而最优偏移向量必为整数向量。在 Matlab2014a 软件平台上,调用 Optimization 优化包中的 intlinprog 函数(IntLPA),并利用符号矩阵的稀

疏性来求解最优偏移向量  $C$  和  $D$ ;并以它们为标准,从算法成功率和效率的角度,通过数值实验来分析启发式算法 PIRA 和 BFIRA 的性能。

为了便于进行数值实验,不妨设耦合块状上三角符号矩阵  $M$  的块数为  $q$ ,每对角子方阵的阶数均为  $n$ ,则符号矩阵  $M$  的阶数为  $N = q \cdot n$ ,记为  $N = \langle q, n \rangle$ 。对角线上每一子方阵中元素按概率{0.7, 0.15 0.05 0.05 0.05} 随机选取{0, 1, 2, 3, 4} 中对应的元素;对角线下方所有子方阵中的元素全取为 -1 000(表示符号矩阵中的  $-\infty$ );对角线上方每一子方阵中每一元素按概率{0.9, 0.05, 0.025, 0.025} 随机选取{-1 000, 0, 1, 2} 中对应的元素。符号矩阵的稀疏率(sparse rate, SR) 为符号矩阵中所有非负元素的总个数与所有元素的总个数的比值,则耦合块状上三角符号矩阵的稀疏率  $SR = \frac{1 + 0.05(q - 1)}{q}$ 。实验平台为 win7\ Matlab2014a,计

算机配置为 Intel i72600 CPU @ 3.4 GHz, RAM 6.00 GB, 64 位操作系统。

选取表 1 中的 17 种情况,每一种情况做 100 次随机实验,并记录 3 种方法的平均运行时间,PIRA 与 BFIRA 的成功率与平均迭代次数。

从表 1 结果中可以看出,在实验平台可计算的情况下  $N = \{10, 100, 1000\}$  的 11 种情况各 100 次随机实验中,启发式 PIRA 和 BFIRA 的成功率均为 1,即 PIRA、BFIRA 和 IntLPA 都可以求出最优偏移向量。随着规模的扩大,3 种方法的计算时间都明显增加,其中, BFIRA 计算时间的增加率最慢; IntLPA 计算时间的增加率最快,而且需要的计算时间越来越多。这是因为对于  $N = q \cdot n$ ,PIRA 和 BFIRA 的计算时间都是由计算最大横截面的时间  $T_{cost}$  和迭代时间  $Iterations \times E_{cost}$  2 部分组成,其中  $E_{cost}$  为对应规模一次迭代计算时间;当规模  $N$  不断扩大时,两部分时间都增加。

从表 1 结果中可以看出,在同一规模  $N$ (10 除外)中,随着符号矩阵稀疏率的降低(子块数  $q$  增加),PIRA 的迭代次数  $Iterations$  逐渐增加,迭代时间  $Iterations \times E_{cost}(N)$  也逐渐增加,从而 PIRA 的计算时间不断增加。在同一规模  $N$ (10 除外)中,随着稀疏率的降低, BFIRA 的计算时间基本呈现先降低后增加的趋势。这是因为对于固定的  $N = q \cdot n$ ,当稀疏率逐渐降低时  $n$  不断减少, BFIRA 计算最大横截面的时间  $T_{cost}(n)$  不断减少;当子块数  $q$  不断增加  $n$  不断减少,迭代次数  $Iterations$  不断增加,一次

迭代计算时间  $E_{cost}(n)$  减少, 迭代时间可能增加, 从而 BFIRA 的计算时间先减少后增加。

表 1 PIRA 和 BFIRA 的性能比较

Tab. 1 Performance comparison between PIRA and BFIRA

$N$	$\langle q, n \rangle$	SR	计算时间 /s			迭代次数		ER
			PIRA	BFIRA	IntLPA	PIRA	BFIRA	
10	$\langle 2, 5 \rangle$	0.525	0.003 1*	0.001 2*	0.032 8	2	3	2.50
	$\langle 5, 2 \rangle$	0.240	0.001 0*	0.001 5*	0.028 1	2	2	0.71
	$\langle 5, 20 \rangle$	0.240	0.020 8*	0.011 9*	0.266 8	10	25	1.75
100	$\langle 10, 10 \rangle$	0.145	0.025 2*	0.012 7*	0.177 8	11	32	1.98
	$\langle 20, 5 \rangle$	0.098	0.029 4*	0.012 1*	0.159 7	13	46	2.44
	$\langle 5, 200 \rangle$	0.240	0.728 8*	0.224 9*	195.844 0	0	0	3.24
1 000	$\langle 10, 100 \rangle$	0.145	0.761 9*	0.159 0*	132.320 0	3	7	4.79
	$\langle 20, 50 \rangle$	0.098	1.257 7*	0.141 9*	22.932 1	26	146	8.87
	$\langle 50, 20 \rangle$	0.069	2.488 0*	0.132 8*	9.484 9	54	251	18.73
	$\langle 100, 10 \rangle$	0.060	3.549 2*	0.139 7*	24.507 8	78	338	25.40
	$\langle 200, 5 \rangle$	0.055	4.619 3*	0.153 9*	30.825 8	104	478	30.01
	$\langle 5, 400 \rangle$	0.240	4.227 1	1.110 5	NULL	0	0	3.81
	$\langle 10, 200 \rangle$	0.145	4.443 0	0.734 8	NULL	0	0	6.05
2 000	$\langle 20, 100 \rangle$	0.098	7.975 5	0.534 1	NULL	6	18	14.93
	$\langle 50, 40 \rangle$	0.069	13.649 8	0.476 0	NULL	58	360	28.68
	$\langle 100, 20 \rangle$	0.060	20.933 7	0.467 4	NULL	103	511	44.78
	$\langle 200, 10 \rangle$	0.055	28.279 0	0.494 2	NULL	150	684	57.23

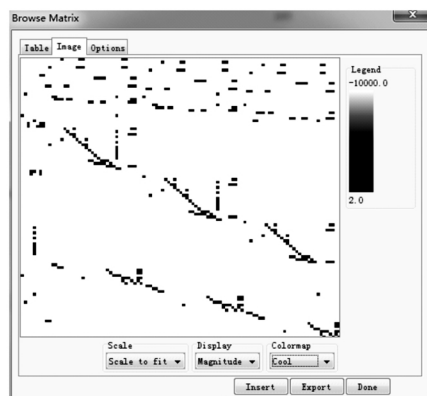
注: \* 表示成功率为 1 ER 表示 PIRA 的计算时间与 BFIRA 的计算时间比值。

最后从表 1 中效率比 ER 可以看出, BFIRA 的效率明显比 PIRA 的效率高, 且其提高倍数 ER 与子块数  $q$  成正相关。这说明对于大规模稀疏系统, 分块快速指标约简方法比 Pryce 方法具有明显的优越性。

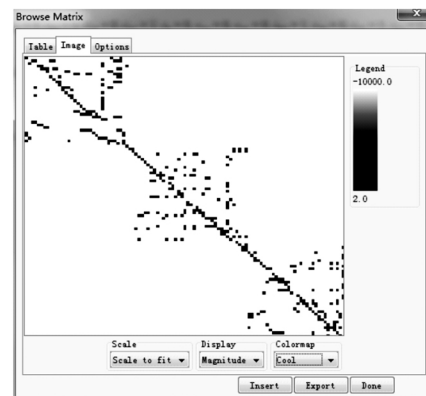
## 2 数控冲床仿真系统 DAEs 指标约简

数控冲床主要运动是由 3 个独立的无刷直流电

机驱动系统分别控制平台和冲头沿 X、Y、Z 轴方向的平移运动, 其仿真模型可分解为丝杠螺母机构、曲柄滑块机构以及电机电路 3 个主要部分<sup>[12]</sup>。该模型通过 Maplesim 软件得到规模为 105 的 DAEs 系统。首先该系统的符号矩阵图(见图 2(a))经过 1.2 节块状上三角化处理后变为图 2(b)。



(a)



(b)

图 2 数控冲床仿真系统的符号矩阵图

Fig. 2 Signature matrix of CNC punching machine tool equations

该系统划分为(34 个方程 + 71 个方程) 2 个相互独立的大块, 其中每一大块又可以继续分解为块状上三角矩阵  $M_i$  为第  $i$  个大块  $M_{jj}$  为对角子方阵,  $i = 1, 2, j = 1, 2, \dots, 9$ 。然后用上述 3 种方法对每一独立块状上三角符号矩阵  $M_i$  进行指标约简, 获取其最优偏移向量  $C_i$  和  $D_i$ , 并记录这 3 种方法的计算时间, 见图 3。

图 3 进一步验证, BFIRA 的效率优于 PIRA 的效率。根据整体方程最优偏移向量  $C = (C_1, C_2)$  确定系统中需要被微分的方程, 见表 2;  $O(i)$  表示原系统中第  $i$  个方程需要被微分一次,  $T(j)$  表示第  $j$  个方程需要被微分 2 次, 则有:

$$\begin{cases} DE_{11} = O(\{18, 19, 20, 55, 56, 57, 58, 59, 61, 95\}); \\ DE_{33} = T(\{37\}) \cup O(\{12, 28, 83, 99\}); \\ DE_{99} = O(\{21, 22, 66, 74, 96, 97, 98, 100\}) \end{cases} \quad (9)$$

表 2 数控钻床系统结构信息

Tab. 2 Structural information of CNC drilling system

	方阵阶数								
	$M_1$ 34 阶		$M_2$ 71 阶						
	$M_{11}$	$M_{22}$	$M_{33}$	$M_{44}$	$M_{55}$	$M_{66}$	$M_{77}$	$M_{88}$	$M_{99}$
需要被微分方程	33	1	1	1	33	1	1	2	32
	$DE_{11}$	$\emptyset$	$\emptyset$	$\emptyset$	$DE_{33}$	$T\{45\}$	$T\{7\}$	$\emptyset$	$DE_{99}$

### 3 结 论

通过本数值实验以及数控冲床为例的实验表明, 启发式分块快速指标约简算法可以求出大规模稀疏微分代数方程系统的最优偏移向量, 其效率明显优于 Pryce 指标约简算法的效率。

虽然分块快速指标约简算法的性能比较好, 但是缺乏严格的理论论证。在以后的工作中, 可以先考虑严格论证 Pryce 指标约简算法的终止性, 然后研究分块快速指标约简算法的终止性, 并进一步分析它们的时间复杂度。

此外, 分块快速指标约简方法和 Pryce 方法等都属于 DAEs 系统的结构分析方法, 如果系统本身是结构奇异的, 那么这 2 个算法都会失效。文献 [13] 中给出一个反例进一步说明含参数非结构奇异 DAEs 系统的结构分析方法仍然会失效。因此, 在以后的工作中, 还可以考虑结合其他指标约简方法 (如 Tractability 指标<sup>[2, 13]</sup>) 来完善 DAEs 系统的指标约简。

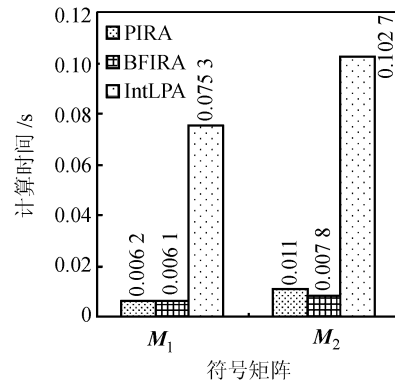


图 3 3 种指标约简算法的比较

Fig. 3 Comparison of three kinds of index reduction algorithm

从表 2 可知, 该系统是结构指标为 3 的高指标 DAEs 系统, 因此指标约简可以提高该系统数值计算方法的稳定性。

### 参考文献:

- [1] Modelica Association. A unified object-oriented language for systems modeling: language specification [EB/OL]. (2012 - 05 - 09 [2014 - 05 - 12]). <https://www.modelica.org/documents/ModelicaSpec33.pdf>.
- [2] Lamour R, März R, Tischendorf C. Differential algebraic equations: A projector based analysis [M]. 2nd ed. Berlin: Springer-Verlag, 2013.
- [3] Hairer E, Wanner G. Solving ordinary differential equations II: Stiff and differential-algebraic problems [M]. 2nd ed. Berlin: Springer-Verlag, 1999.
- [4] Gear W C. Differential-algebraic equations index transformations [J]. SIAM Journal on Scientific and Statistical Computing, 1988, 9(1): 39 - 47.
- [5] Pantelides C C. The consistent initialization of differential-algebraic systems [J]. SIAM Journal on Scientific and Statistical Computing, 1988, 9(2): 213 - 231.

- [6] Pryce J D. A simple structural analysis method for DAEs [J]. BIT Numerical Mathematics 2001 41(2): 364 - 394.
- [7] Ding Jianwan. Research on methods for consistency analysis and reduction of declarative simulation models [D]. Wuhan: Huazhong University of Science & Technology 2006. [丁建完. 陈述式仿真模型相容性分析与约简方法研究 [D]. 武汉: 华中科技大学 2006.]
- [8] Ding Jianwan, Zhang Chenjian, Chen Liping. Index analysis of discontinuous differential-algebraic models for performance simulation [J]. Journal of Computer-Aided Design & Computer Graphics 2008 20(5): 585 - 590. [丁建完, 张诚坚, 陈立平. 面向性能仿真的非连续微分代数模型的指标分析 [J]. 计算机辅助设计与图形学报 2008, 20(5): 585 - 590.]
- [9] Yang Haijun. Index reduction algorithm analysis of algebraic differential equation from Modelica software [J]. Computer Engineering and Design 2010 31(1): 130 - 134. [阳海军. Modelica 软件中微分代数方程指标约简算法分析 [J]. 计算机工程与设计 2010 31(1): 130 - 134.]
- [10] Pryce J D, Nedialkov N S, Tan Guangning. DAESA: A matlab tool for structural analysis of DAEs: Theory [EB/OL]. (2014 - 01 - 04) [2014 - 05 - 12]. [http://www.cas.mcmaster.ca/%7Eenedialk/PAPERS/DAEs/daesa\\_theory/daesaTheory.pdf](http://www.cas.mcmaster.ca/%7Eenedialk/PAPERS/DAEs/daesa_theory/daesaTheory.pdf).
- [11] Saad Y. Iterative methods for sparse linear systems [M]. 2nd ed. New York: SIAM 2003.
- [12] 廖效果, 刘又午. 数控技术 [M]. 武汉: 湖北科学技术出版社 2004: 237 - 242.
- [13] Lamour R, März R. Detecting structures in differential algebraic equations: Computational aspects [J]. Journal of Computational and Applied Mathematics 2012 236: 4055 - 4066.

(编辑 杨 蓓)