

微分代数系统结构化分析

李光远^{1,2†}, 冯 勇²

(1. 中国科学院大学, 北京 100049; 2. 中国科学院 成都计算机应用研究所, 四川 成都 610041)

摘要: 对工程和科学问题进行建模和仿真的时候, 人们常常很自然地会用微分代数系统对这些问题进行描述. 为了检验微分代数系统的初始相容性并进行求解, 对微分代数系统进行结构化分析非常重要. 本文对经典的微分代数系统结构化分析方法进行了深入的研究; 提出了一种新的结构化分析方法, 可以高效地对大规模、高阶高指标的微分代数系统进行结构化分析, 并快速检验其初始相容性; 证明了该方法的终止性, 分析了其最坏时间复杂度. 该方法的关键在于对最大加权二部子图的使用, 而最大加权二部子图则来源于原始系统的加权二部图. 实验结果显示, 该方法能高效地完成对微分代数系统的结构化分析.

关键词: 微分代数系统; 结构化分析; 初始相容性; 加权二部图; 算法

中图分类号: TP273

文献标识码: A

Structural analysis for differential algebraic systems

LI Guang-yuan^{1,2†}, FENG Yong²

(1. University of China Academy of Science, Beijing 100049, China;

2. Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu Sichuan 610041, China)

Abstract: It is natural to describe physical system with differential algebraic system when modeling and simulating many engineering and scientific problems. Structural analysis is very important for the consistent initialization of differential algebraic system and finally solving it. In this paper, we research the classical methods for structural analysis of differential algebraic systems; and then we propose a new and more time efficient method for structural analysis of large scale differential algebraic systems that have a high index and a high order, and this method can quickly verifies the consistent initialization of the differential algebraic system, find out which equations and how many times they need differentiating as well; we prove the termination of this new method and analyze the worst time complexity of it. The key to this proposed method is the use of maximal weighted bipartite sub-graphs, which are derived from the base of weighted bipartite graphs of the original system. Demonstration and testing show that this method is effective and time efficient for structural analysis of differential algebraic systems.

Key words: differential algebraic system; structural analysis; consistent initialization; weighted bipartite graph; algorithm

1 引言(Introduction)

对工程和科学问题进行建模和仿真的时候, 人们常常很自然地会使用微分代数系统对物理系统进行描述, 微分代数系统也引起了越来越多的关注^[1]. 典型的物理系统如多体动力系统、航空器飞行轨迹追踪、优化控制系统、工艺过程、电子电路仿真和机器人路径规划问题等等, 都可以用微分代数系统进行模拟^[2-4]. 一旦求解出了建模的微分代数系统, 就可以对相应的物理系统进行仿真. 微分代数系统包含了微分方程和代数方程的特性, 以及混合这两种方程所导致的其他特性^[5], 为了求解微分代数系统就需要对其进

行结构化分析, 选择合适的求解方法. 另外, 在求解微分代数系统之前, 为了确保所建立的模型是合理并且可解的, 也需要通过结构化分析对微分代数系统的初始相容性进行验证. 因此, 无论是为了求解微分代数系统还是验证其初始相容性, 都需要对微分代数系统进行结构化分析.

到目前为止, 学界已经提出了很多关于微分代数系统结构化分析的方法, 这些方法几乎都需要将微分代数系统的微分指标降为0(关于微分指标的定义, 可以参考文献[6]和文献[7]), 将系统转化常微分方程. Pantelides在文献[8]中率先提出了微分代数系统的初

收稿日期: 2017-01-26; 录用日期: 2017-04-14.

[†]通信作者. E-mail: li-guang-yuan@sina.com; Tel.: +86 13658305636.

本文责任编辑: 刘允刚.

国家“973”计划项目(NKBRPC-2011CB302402), 国家自然科学基金项目(61402537, 91118001)资助.

Supported by China “973” Project (NKBRPC-2011CB302402) and National Science Foundation of China (61402537, 91118001).

始相容性问题,并给出了一种对微分代数系统进行结构化分析的方法. Pantelides方法利用一种简单二部图算法,找出微分代数系统中的最小结构化奇异子集(minimally structurally singular, MSS),然后对MSS中的方程进行微分,从而修改二部图,重复上述两个步骤直到方程组中的所有MSS子集都被找出来为止(关于结构化奇异以及MSS的定义,可以参考文献[8]).这种方法取得了较好的效果并得到了广泛的运用,但这种方法只能处理一阶的微分代数系统,如果用于处理高阶的微分代数系统,则该方法的处理时间和空间都会成倍地增加.在文献[9]中, Gear提出了另一种方法, Gear方法通过一系列的符号操作找出微分代数系统中的代数方程,并对这些代数方程进行微分,重复这些处理过程直到微分代数系统被转化为常微分方程,这是一种符号方法并能将微分指标完全降低为0.但是对普通的非线性微分代数系统, Gear方法需要进行大量的操作,其中很多操作都比较困难,有时甚至根本就不能完成. Mattsson和Söderlind在文献[10]中提出了一种虚拟方法.在这种虚拟方法中,对原系统中部分或全部方程进行微分,新得到的方程与原方程系统共同构成了一个增广的方程系统.用代数变量替换微分变量,就可以将这个增广的方程系统转化为一个纯粹的代数方程系统并最终对其进行求解.为了实现增广的方程系统向代数方程系统的转化,这种虚拟方法需要对许多方程进行不必要的微分,这就增加了操作的复杂性,极大地降低了处理的效率. Pryce提出了一种签名方法,或者说叫 Σ -方法^[11-12]. Σ -方法定义了一种签名矩阵 $\Sigma = (\sigma_{ij})$, σ_{ij} 是第 i 个方程中第 j 变量导数的最高次阶,如果该方程中没有出现这个变量,则取 σ_{ij} 为 $-\infty$.在得到这个矩阵的最大横截面之后(highest-value transversal, HVT),就可以通过一个不动点迭代算法求这个HVT的对偶,得到的结果就是需要微分的方程与微分的次数 c ,以及微分后各个变量导数的最高次阶 d (关于最大横截面的定义,可以参考文献[11, 13]). Σ -方法的实现有赖于方程系统的稀疏性,并且忽略了这样一个事实,那就是方程的解析形式和稀疏结构可能会改变解的范围,而这在实际应用中是非常重要的.并且 Σ -对方程系统的奇异性验证并不是一开始就进行,而是在最后通过一个Jacobian矩阵来完成的,而这个Jacobian矩阵的构建需要使用先前求得的 c 和 d .因此,不管微分代数系统是否结构化非奇异, Σ -方法都需要完整地执行一遍,这对一些结构化奇异的系统显然是不必要的.对于较大规模的问题,秦小林等人充分利用了其稀疏性,提出了一种三角分块化的方法^[14].借助三角分块化技术,秦方法改进了 Σ -方法,使其在解决大规模的微分代数系统时具有更高的效率,但这种方法的基础依然是签名矩阵和不动点迭代算法.

本文以加权二部图表示微分代数系统,提出了一种高效的微分代数系统结构化分析方法.该方法定义了微分代数系统的最大加权二部图和对应的最大加权二部子图,并通过系统最大加权二部子图进行适当的操作,实现了对微分代数系统,尤其是大规模、高阶高指标的微分代数系统的结构化分析.最后,通过实验对该方法进行了验证.

2 最大加权二部子图(Maximal weighted bipartite sub-graph)

本文以如下形式表示一般的微分代数系统:

$$f_i(t, x, x', \dots, x^{(\sigma)}) = 0, 1 \leq i \leq n, \quad (1)$$

其中: $t \in [t_0, t_f]$, $x(t) \in \mathbb{R}^m$ 且 $\sigma \geq 2$, 在大多数情况下都默认 $n = m$. 一个如方程(1)那样的微分代数系统在 t_0 时刻是初始相容的, 当且仅当存在初始值 $(x_0, x'_0, \dots, x_0^{(\sigma)})$ 在 t_0 时刻使得方程(1)成立,

$$f_i(t_0, x_0, x'_0, \dots, x_0^{(\sigma)}) = 0, 1 \leq i \leq n. \quad (2)$$

这时也可以说初始值 $(x_0, x'_0, \dots, x_0^{(\sigma)})$ 对方程(1)那样的微分代数系统是相容的.

微分代数系统一般都是高指标和高阶的, 实际应用中还常常具有较大的规模, 因此要对微分代数系统进行结构化分析, 效率就成为了关键. 二部图能够很好地展示方程和变量之间的关系, 而加权二部图则可以对高阶高指标微分代数系统进行更详细地描述. 利用加权二部图表示微分代数系统, 可以有效地提高微分代数系统结构化分析的效率.

定义 1 对方程(1)那样的微分代数系统, 在互补的方程 $\{f\}$ 的顶点集 V_e 和变量 $\{x\}$ 的顶点集 V_v 上建立一个二部图 $G = (V_e \cup V_v, E)$, 并以方程 f_i 中出现的变量 x_j 导数的最高次阶 k 作为相应的边 (f_i, x_j) 的权, 这样的二部图 G 称为微分代数系统的加权二部图(weighted bipartite graph, WBG).

例 1 如图1所示的简单物理摆, 其中 L 表示单摆的臂长, λ 表示作用在单摆臂上的力, x 和 y 表示单摆臂一端质量无限小的摆球的笛卡尔坐标^[15].

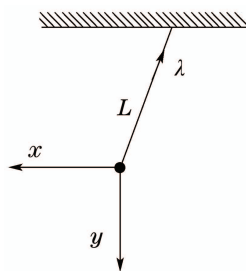


图1 简单的物理摆

Fig. 1 Simple physical pendulum

这样的物理摆可以建模成一个如方程(3)的微分代数系统, 这是一个3-指标的2-阶微分代数系统, 包

含3个方程和3个变量.

$$\begin{cases} f_1: x'' - x\lambda = 0, \\ f_2: y'' - y\lambda + g = 0, \\ f_3: x^2 + y^2 - L^2 = 0. \end{cases} \quad (3)$$

图2表示方程(3)所示微分代数系统的加权二部图, 方程节点集 $\{f_1, f_2, f_3\}$ 与变量节点集 $\{x, y, \lambda\}$ 之间每条边的权用相应的方程 f_i 中出现的变量导数的最高阶来标注.

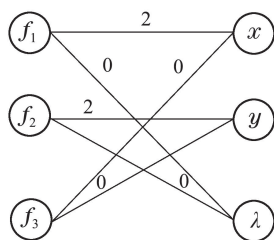


图2 方程(3)的加权二部图

Fig. 2 WBG of equation (3)

加权二部图在计算机中可以用矩阵来表示. 如果将图2表示成如式(4)的矩阵:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

可以发现这个矩阵与文献[11]中的签名矩阵非常相似, 如式(5)所示:

$$\begin{bmatrix} 2 & -\infty & 0 \\ -\infty & 2 & 0 \\ 0 & 0 & -\infty \end{bmatrix}. \quad (5)$$

实际上, 根据定义1, 加权二部图和签名矩阵是一致的, 只是签名矩阵使用了 $-\infty$ 来表示方程 f_i 中不存在变量 x_j . 为了便于编码, 本文在加权二部图的矩阵中用 -1 来表示方程和变量节点不相连, 如式(6)所示.

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (6)$$

定义2 对加权二部图 $G = (V_e \cup V_v, E)$, 保留与变量节点相连的权值最大的边, 删除其余的边, G 中剩余的部分称为 G 关于变量节点的最大加权二部子图(maximal weighted bipartite subgraph, MWBS), 并用 $\bar{G} = (V_e \cup V_v, \bar{E})$ 来表示, 保留下来的边集用 $\bar{E}$ 表示.

如图3(a)所示, 从变量节点出发, 保留与变量节点 $\{x, y, \lambda\}$ 相连的权值最大的边, 删除其余的边, 就得到了图2关于变量节点 $\{x, y, \lambda\}$ 的最大加权二部子图, 如图3(b)所示.

高指标微分代数系统一直存在着确定相容初始值的问题, 为了解决这个问题, 学界提出了很多算法将

微分指标降为0, 通过对其中的一些方程进行微分将微分代数系统转化为常微分方程. 但在将微分代数系统转化为常微分方程的过程中, 除了可能使数值解从约束流行发生漂移外, 不加选择地微分对大规模的系统不但不实际, 而且也不必要^[5], 所以准确地找出系统中需要微分的方程和微分的次数就显得非常重要了. 这个寻找的过程可以通过反复地检测微分代数系统中的MSS子集并对其中的方程进行微分来实现. 利用最大加权二部子图, 可以更快更有效地找出高阶大规模微分代数系统中的MSS子集.

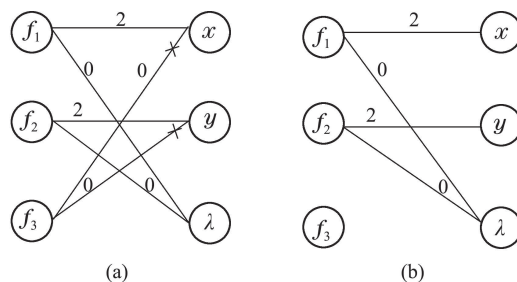


图3 图2的最大加权二部子图

Fig. 3 MWBS of Fig. 2

定义3 对一个微分代数系统的二部图 $G = (V_e, V_v, E)$, 如果 G 的一个匹配 M 包含了 V_e 中的所有方程节点, 则可以将 M 称为关于方程节点集 V_e 的完全匹配, 或者说匹配 M 关于方程节点集 V_e 是完全的, 否则 M 关于方程节点集 V_e 是部分匹配的.

为了实现结构化分析, 首先需要对微分代数系统的结构化非奇异性进行验证. Pantelides在文献[8]中提出, 可以使用文献[15]中的算法通过确定最大横截面来检测微分代数系统的结构化非奇异性. 本文提出了一种更简单、更直接的方法, 那就是通过验证相对于方程节点集的最大匹配的存在性来检测微分代数系统的结构化非奇异性, 在下一节里将证明并使用这种方法.

3 基于MWBS的结构化分析(Structural analysis based on MWBS)

为了实现微分代数系统的结构化分析, 有3个重要的问题需要仔细考虑. 首先, 若微分代数系统是结构化奇异的, 则需要对系统进行调整或重新建模, 使得系统结构化非奇异^[16-17]. 只有当微分代数系统经过验证是结构化非奇异的, 才有必要继续进行结构化分析, 而验证的结果也可以用于后续的分析过程. 其次, 结构化分析也是一个将微分代数系统转化为常微分方程组的过程, 在这个过程中有的方程需要微分一次或多次, 但某些微分代数系统可能没有MSS子集(这样的微分代数系统实际上就是一个常微分方程组), 或者系统中存在需要微分的方程却没能检测出来. 第三, 对方程的微分会增加变量导数的阶并改变方程和变量之间的关系, 因此微分代数系统的加权二部图与对

应的最大加权二部子图都需要多次重构,直到分析过程结束.

根据前面讨论的3个问题,利用微分代数系统的加权二部图以及最大加权二部子图,本文提出一种新的微分代数系统结构化分析方法,算法的伪代码如算法1所示.在算法1中,微分代数系统的加权二部图 $G = (V_e \cup V_v, E)$ 是整个算法的唯一输入; $\bar{G} = (V_e \cup V_v, \bar{E})$ 是 G 的最大加权二部子图; E 是 G 的边集; M 是 G 中关于方程节点的一个匹配; $\bar{M}$ 是 $\bar{G}$ 中关于方程节点的一个匹配; $\text{colored}V_e$ 是一个矢量,用于标记当前访问过的方程节点.若微分代数系统是结构化非奇异的, M 将成为一个完全匹配;若在这个微分代数系统中同时还检测出了MSS子集,则 G , $\bar{G}$ 和 E 都将在while循环中多次重构,最终 $\bar{M}$ 也将成为一个完全匹配,而算法将返回true;否则,算法将返回false.算法1结束后,若返回false,且得到的 M 是一个部分匹配,则表明这个微分代数系统是结构化奇异的;若返回false,而得到的 M 和 $\bar{M}$ 都是完全匹配,则表明这个微分代数系统中没有方程需要微分或没有检测出需要微分的方程;若返回true,对比最后得到的 G 和原始的 G ,就可以得到需要微分的方程以及微分的次数.可以将原始微分代数系统和微分后得到的方程共同构成的系统称为原系统的增广系统,因此最后得到的 G 还表明了原系统的增广系统所包含的方程数和各个变量导数的最高次阶.

算法1 基于MWBS的微分代数系统结构化分析.

输入: $G = (V_e \cup V_v, E)$

输出: $\bar{G}$

步骤1 初始化.

- 1) $n \leftarrow$ 方程节点数
- 2) $m \leftarrow$ 变量节点数
- 3) $k \leftarrow n + 1$
- 4) $M \leftarrow \emptyset$
- 5) 根据原始 G 得到 $\bar{G}$

步骤2 检测系统的结构化非奇异性并得到 M .

- 1) for $i = 1$ to n do
- 2) $\text{colored}V_e(j) \leftarrow \text{false}, j = 1$ to n
- 3) if not Augment($G, M, \text{colored}V_e, i$) then
- 4) return false
- 5) end if
- 6) end for

步骤3 得到 $\bar{M}$.

- 1) 根据 M 和 $\bar{G}$ 得到 $\bar{M}$
- 2) if $\bar{M}$ is complete then
- 3) return false
- 4) end if
- 5) $k \leftarrow$ 第1个不属于 $\bar{M}$ 的方程序列号

步骤4 找出需要微分的方程以及微分的次数.

- 1) while $k \leq n$ do
- 2) $\text{colored}V_e(i) \leftarrow \text{false}, i = 1$ to n
- 3) if Augment($\bar{G}, \bar{M}, \text{colored}V_e, k$) then
- 4) $k \leftarrow k + 1$
- 5) else
- 6) for $i = 1$ to n do
- 7) if $\text{colored}V_e(i)$ then
- 8) for $j = 1$ to m do
- 9) if $E(i, j) \in G$ then
- 10) $E(i, j) \leftarrow E(i, j) + 1$
- 11) end if
- 12) end for
- 13) end if
- 14) end for
- 15) 根据重构 G 的得到 $\bar{G}$
- 16) end if
- 17) end while

步骤5 return true.

在算法1中,作者使用了一个过程Augment,该过程对文献[18]中所提出的算法进行了必要的修改,用于搜索微分代数系统的加权二部图与最大加权二部图子图中的增广路径.除了返回一个布尔值以表明是否能找到增广路径外,过程Augment还会改变匹配 M 或 $\bar{M}$,并用矢量 $\text{colored}V_e$ 标注访问过的方程节点.过程Augment的伪代码如算法2所示.

算法2 Augment.

输入: $G = (V_e \cup V_v, E), M, \text{colored}V_e, k$

输出: M

- 1) $n \leftarrow$ 方程节点数
- 2) $m \leftarrow$ 变量节点数
- 3) $\text{colored}V_e \leftarrow \text{true}$
- 4) for $j = 1$ to m do
- 5) if $E(k, j) \in G$ and $E(k, j) \notin M$ then
- 6) $M \leftarrow M \cup \{E(k, j)\}$
- 7) return true
- 8) end if
- 9) end for
- 10) for $j = 1$ to m do
- 11) if $E(k, j) \in G$ then
- 12) for $i = 1$ to n do
- 13) if $E(i, j) \in M$ and not $\text{colored}V_e(i)$ then
- 14) if Augment($G, M, \text{colored}V_e, i$) then
- 15) $M \leftarrow M \cup \{E(k, j)\}$
- 16) $M \leftarrow M \otimes \{E(i, j)\}$
- 17) return true
- 18) end if
- 19) end if
- 20) end for

- 21) end if
22) end for
23) return false

算法1在第2)步计算原始加权二部图 G 关于方程节点的匹配 M , 从而判断微分代数系统是否结构化非奇异. 微分代数系统的结构化非奇异性可以通过计算其最大横截面来确定^[13], 而微分代数系统最大横截面的存在性, 则与系统关于方程节点的完全匹配的存在性紧密相关.

引理 1 若方程(1)那样的微分代数系统存在最大横截面, 则其加权二部图 $G = (V_e \cup V_v, E)$ 关于方程节点集 V_e 的完全匹配存在; 反之亦然.

证 方程(1)的微分代数系统加权二部图可以用一个关联矩阵来表示, 其形式与文献[11]中的签名矩阵类似. 在这个微分代数系统的加权二部图中, 若存在关于方程节点集 V_e 的完全匹配, 尽管完全匹配可能不唯一, 但总能找到一个总权值最大的完全匹配, 这个完全匹配中所有边的权值正好就构成了关联矩阵中该系统的最大横截面. 所以, 如果微分代数系统加权二部图 G 关于方程节点集 V_e 的完全匹配存在, 则该系统的最大横截面存在.

反之, 若方程(1)那样的微分代数系统存在最大横截面, 而其加权二部图关于方程节点集的完全匹配不存在. 可以假设存在一个部分匹配, 那么至少有一个方程节点不属于这个部分匹配, 因此微分代数系统关联矩阵的横截面中至少包含一个 $-\infty$, 这样该系统就不可能存在最大横截面. 产生的矛盾表明如果微分代数系统的最大横截面存在, 则其加权二部图关于方程节点集 V_e 的完全匹配必然存在. 证毕.

对于方程(1)那样的微分代数系统来说, 最大横截面存在的充分必要条件是其加权二部图存在关于方程节点集 V_e 的完全匹配. 因此, 微分代数系统的结构化非奇异性就可以通过其关于方程节点集 V_e 的完全匹配的存在性来检验. 虽然, 通过计算系统的最大横截面, 或找出其关于方程节点集的完全匹配都可以检验微分代数系统的结构化非奇异性, 从检验的结果上来看2种方法是等价的, 但后者显然更容易实现, 而且效率更高.

算法1的第3)步计算原始加权二部图 G 的最大加权二部子图 $\bar{G}$ 关于方程节点集的匹配 $\bar{M}$. 若匹配 $\bar{M}$ 在第3步结束后成为了完全匹配, 则微分代数系统要么没有MSS子集(或者说没有方程需要微分), 要么算法没能找出需要微分的方程.

引理 2 若方程(1)那样的微分代数系统结构化非奇异, 且系统原始加权二部图 G 的最大加权二部子图 $\bar{G}$ 关于方程节点集 V_e 的完全匹配存在, 则该微分代数系统中找不出MSS子集.

证 对方程(1)那样结构化非奇异的微分代数系统, 假定其原始加权二部图的最大加权二部子图中存在关于方程节点集 V_e 完全匹配 $\bar{M}$, 而且也能在该系统中找出一个MSS子集. 显然, 若微分代数系统中能找到一个MSS子集, 则这个子集中的方程一定满足文献[8]中的准则(10), 也就是 $l < k$, 因此在这样的子集中变量的个数必然要少于方程的个数. 因为在完全匹配 $\bar{M}$ 中, 每一个变量节点都仅仅与一个方程节点相连, 所以在这样的子集中至少有一个方程节点没有变量节点与之相连, 这样矛盾就出现了. 因而, 若 $\bar{G}$ 中存在完全匹配 $\bar{M}$, 则对应的微分代数系统中找不出MSS子集. 证毕.

需要注意的是引理2是一个充分但不必要的条件. 若微分代数系统中没有方程需要微分, 通过适当的行操作(如高斯消元), 可以将其最大加权二部子图 $\bar{G}$ 对应的签名矩阵转化为一个非负的对角矩阵, 对角线上的每个元素都满足 $\Sigma_i \geq 0 (1 \leq i \leq n)$, 如式(7)所示,

$$\begin{bmatrix} \ddots & & & -\infty \\ & \Sigma_i & & \\ & & \ddots & \\ -\infty & & & \ddots \end{bmatrix}. \quad (7)$$

对应的 $\bar{M}$ 将是一个完全匹配, 如图4(a)所示.

若微分代数系统中有方程需要微分却不能检测出来, 同样通过适当的行操作, 可以将最大加权二部子图 $\bar{G}$ 对应的签名矩阵转化为一个对角线上元素非负的矩阵, 对角线上的元素都满足 $\Sigma_i \geq 0$ 与 $\Sigma_j \geq 0 (1 \leq i \leq n, 1 \leq j \leq n)$, 但在某些行上会存在更多的 Σ_i 和 Σ_j , 如式8所示

$$\begin{bmatrix} \ddots & & & -\infty \\ & \Sigma_i & \Sigma_j & \\ & \Sigma_i & \Sigma_j & \\ -\infty & & & \ddots \end{bmatrix}. \quad (8)$$

正是这些行对应了那些需要微分却没能检测出来的方程, $\bar{M}$ 同样也将是一个完全匹配, 如图4(b)所示.

对于前一种情况, 关于微分代数系统的结构化分析至此已经足够并可以结束了; 但对于后一种情况, 则还需要针对特定的问题采取更多具体的措施, 从而完成结构化分析.

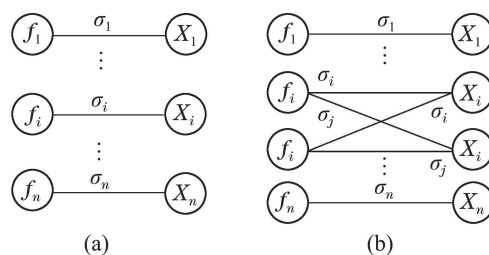


图 4 行操作完成后 $\bar{G}$ 的完全匹配 $\bar{M}$

Fig. 4 $\bar{M}$ of $\bar{G}$ after row operations

定理1 对方程(1)那样的微分代数系统, 算法1总能结束. 如果微分代数系统中找出了需要微分的方程, 则算法1返回true; 否则算法1返回false.

证 为了验证方程(1)表示的微分代数系统的结构化非奇异性, 算法1在第2)步通过循环调用过程Augment来检测系统关于方程节点 V_e 的匹配 M 是否能成为完全匹配. 若微分代数系统是结构化奇异的, 根据引理1, 系统的最大横截面不存在, 匹配 M 将不能成为完全匹配, 总有一次执行过程Augment时会返回false. 一旦过程Augment返回false, 算法1将返回false并立即终止, 此时微分代数系统也被证明是结构化不相容的. 若微分代数系统是结构化非奇异的, 根据引理1, 系统的最大横截面存在, 过程Augment的每一次执行都将返回true. 所有的方程节点都被访问后, M 将成为一个完全匹配, 算法1继续执行.

算法1在第3)步中搜索最大加权二部子图 $\bar{G}$ 关于方程节点的匹配 $\bar{M}$. 若 $\bar{M}$ 在第3)步执行结束后成为了完全匹配, 则算法1返回false并终止执行. 根据引理2, 这时返回false表示微分代数系统没有方程需要微分, 或没有找出需要微分的方程. 否则, $\bar{M}$ 是一个部分匹配, 且算法1继续执行.

对于结构化非奇异且存在MSS子集的微分代数系统, 算法1在第4)步循环地重构系统的加权二部图 $G = (V_e \cup V_v, E)$ 和最大加权二部子图 $\bar{G} = (V_e \cup V_v, \bar{e})$, 使得 $\bar{M}$ 最终能成为一个完全匹配. 不失一般性, 假定算法1在第1次进入while循环时, $\bar{M}$ 是一个部分匹配, 接着执行过程Agument. 若 $\bar{G}$ 中的方程节点 f_k 与变量节点 v_j 相邻, 且 $v_j \notin \bar{M}$, 则增广路径存在, 过程Augment返回true, 并把边 (f_k, v_j) 加入到 $\bar{M}$ 中, 然后 $k \leftarrow k + 1$. 算法1进入到又一次while循环中, 并再次调用过程Augment来搜寻下一个方程节点的增广路径. 若方程节点 f_k 在 $\bar{G}$ 中是孤立的, 则每一次对方程 f_k 的微分都会使其邻接的边权值加1, G 和 $\bar{G}$ 都将重新构造, 最终这个方程节点将至少与一个变量节点 v_j 相连. 若 $v_j \notin \bar{M}$, 则将边 (f_k, v_j) 加入到 $\bar{M}$ 中. 若 $v_j \in \bar{M}$, 则与 v_j 相连且属于 $\bar{M}$ 的另一个方程节点 f_i 将被搜索到. 如果在 $\bar{G}$ 中 f_i 仅仅与 v_j 相连, 则 f_i 与前面访问到的方程节点 f_k 都将被微分, 这就会使得相应的边权值加1. 重构加权二部图 G 之后, 一个新的最大加权二部子图 $\bar{G}$ 也将得到. 如果在接下来的while循环中, 能找到一条途经 f_k, v_j, f_i 和 v_{j+1} 的路径, 则过程Augment返回true, 否则更多的方程节点将会被访问到, 并加以微分以构建增广路径. 可以知道, 如果一个微分代数系统是结构化奇异的, 则该系统中的方程数量多于变量数量. 因此, 对于方程(1)那样结构化非奇异的微分代数系统来说, 其变量节点必然不少于方程节点. 所以, 经过足够多次的while循环, 必然能构建出途经 f_k, v_j, f_i 和 v_{j+1} 以及其他节点的增广路径. 重复上

述过程, $\bar{M}$ 必将成为关于方程节点集 V_e 的完全匹配. 最后, 第5)步将被执行, 算法1也将返回true. 证毕.

使用算法1对方程(1)表示的微分代数系统进行结构化分析时, 最坏时间复杂度主要与算法第4)步有关. 为了在 $\bar{G}$ 中寻找增广路径, 算法1从第 k 个方程节点开始(当前 $f_k \notin \bar{M}, 1 \leq k \leq n$), 访问过的变量节点最多比 $\bar{M}$ 中的方程节点多1个, 即在为第 k 个方程节点搜索匹配的变量节点时, 最多需要访问 $k - 1 + 1 = k$ 个变量节点. 另外, 由于 $\bar{G}$ 中的边都是 G 中与变量节点相邻的权值最大的边, 为了保证 $\bar{G}$ 中的方程节点 f_k 不是孤立的节点, 方程 f_k 中至少有一个变量在整个系统中拥有最高阶的导数. 否则就需要对方程 f_k 进行微分, 微分的次数不能少于方程 f_k 中所有变量导数的最高次阶 $\nu (0 \leq \nu \leq \sigma)$ 与整个微分代数系统中同一个变量(假设是 x_j)导数的最高次阶 $\omega (1 \leq \omega \leq \sigma)$ 之差 $\delta (1 \leq \delta \leq \sigma)$ 的最小值. 在微分 $\delta (\delta = \omega - \nu)$ 次之后, 若方程节点 f_k 任然无法找到匹配的变量节点, 则需要对方程 f_k 与至少一个其他的方程 f_i 进行微分, 此时边 $(f_i, v_j) \in \bar{M}$. 经过最多 σ 次微分后, 在新构建的 $\bar{G}$ 中总能找到一条从 f_k 开始, 途径 x_j, f_i 和 v_{j+1} 等节点的增广路径, 从而将方程节点 f_k 纳入到 $\bar{M}$ 中去. 因此, 为了将方程节点 f_k 纳入 $\bar{M}$, 算法1第4)步最多需要执行 $k\sigma$ 次while循环. 归纳起来, 在最坏情况下算法1第4)步需要执行的循环次数就应该是

$$\sigma + 2\sigma + \cdots + k\sigma + \cdots + n\sigma = \frac{n(n+1)\sigma}{2}.$$

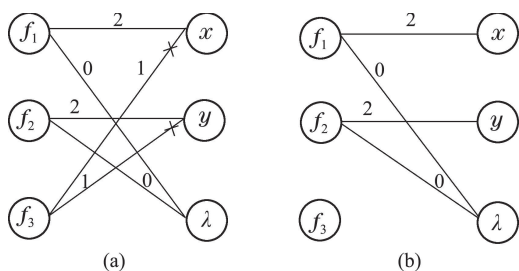
这样, 算法1的最坏时间复杂度就不超过 $n(n+1)\sigma/2$, 也就是 $O(\sigma n^2)$, 其中 n 是原始微分代数系统的方程个数, 而 σ 则是原始系统中变量导数的最高次阶. 通常情况下, σ 是一个远小于 n 的常数, 因此算法1的最坏时间复杂度就应该是 $O(n^2)$.

4 应用举例(Application of Algorithm 1)

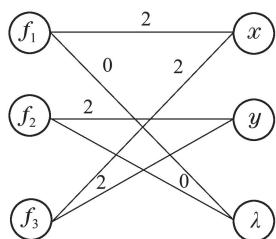
在这一节里用了两个例子来说明算法1的应用过程, 并通过实验验证了算法1的实际应用效果, 最后对实验结果进行了详细的分析.

例2 如图1的简单物理摆, 建模后的微分代数系统为方程(3), 对应的加权二部图如图2所示, 方程节点集 $V_e = \{f_1, f_2, f_3\}$, 变量节点集 $V_v = \{x, y, \lambda\}$, 方程节点数 n 和变量节点数 m 都等于3.

将算法1应用于这个微分代数系统, 在第3)步结束后, $\bar{M} = \{(f_1, x), (f_2, y)\}, k = 3$. 接下来在第1次while循环执行过程中, 算法标记了方程节点 f_3 并微分对应的方程. 边 (f_3, x) 和 (f_3, y) 的权值都增加1, 从0变成了1, 加权二部图 G 转化为图5(a). 算法第4)步的第15行代码执行后, 得到新的最大加权二部子图 $\bar{G}$ 如图5(b)所示. 从形式上看, 图5(b)和图3(b)是一样的, 但图5(b)中的方程 f_3 已经微分了一次. 这时 k 依然等于3, 算法进入第2次while循环.

图5 单摆模型第1次while循环后的 G 和 $\bar{G}$ Fig. 5 G and $\bar{G}$ after the 1st while-loop

在第2次while循环中, 过程Augment依旧返回false, 方程节点 f_3 被标记, 对应的方程需要再进行一次微分, k 也保持为3, 而 G 则转化为图6. 第4)步的第15行代码再次执行后, 因为在获得 $\bar{G}$ 的时候没有边被删除, 所以得到的 $\bar{G}$ 形式与 G 一样如图6所示, 算法进入第3次while循环.

图6 单摆模型第2次while循环后的 G 和 $\bar{G}$ Fig. 6 G and $\bar{G}$ after the 2nd while-loop

在这一次while循环中, 过程Augment返回了true, k 变为了4, 因此循环结束, 进而整个算法返回true并结束. 在本例中, 算法1结束后可以确定需要微分的方程和次数, 用一个序列表示就是(0, 0, 2), 微分后得到的增广系统中各个变量导数的最高次阶则是(2, 2, 0). 若将各个变量不同阶的导数视为不同的变量, 则这个增广系统有7个变量, 分别是 $\{x, x', x'', y, y', y'', \lambda\}$ 和5个方程, 如式(9)所示, 因此需要为两个变量赋给初始值.

$$\begin{cases} x'' - x\lambda = 0, & y'' - y\lambda + g = 0, \\ x^2 + y^2 - L^2 = 0, & xx' + yy' = 0, \\ x'^2 + xx'' + y'^2 + yy'' = 0. \end{cases} \quad (9)$$

例3 如图7所示经过神经突触的脉冲传输模型, 其中 k , A 和 B 表示反应速率, 都是正的常数; c 表示突触一边的输入, C_i 表示中间介质, O 表示突触另一边的输出, 这3者都随时间发生变化; 模型需要确定 $O(t)$ 等于指定的 $g(t)$ 时, $c(t)$ 的变化情况.

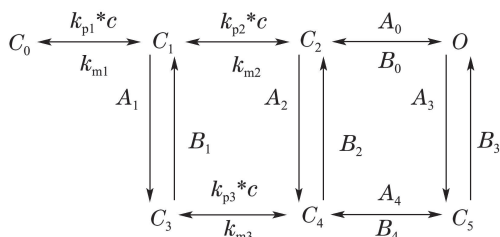


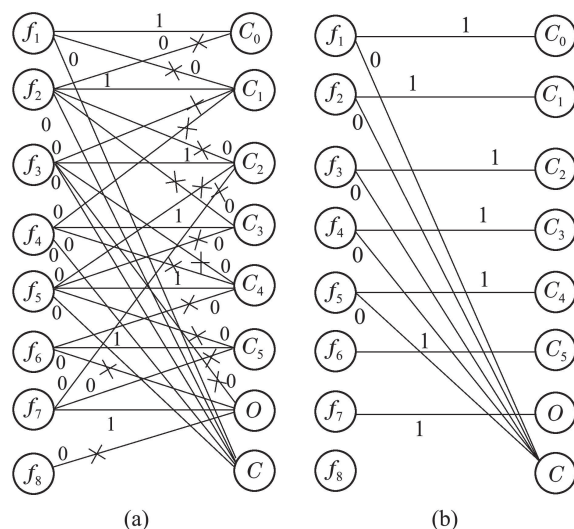
图7 经过神经突触的脉冲传输

Fig. 7 Pulse transmission across a nerve synapse

上述模型可以表示成一个如式(10)所示的一阶微分代数系统, 整个系统有8个方程和8个变量 $\{C_0, C_1, C_2, C_3, C_4, C_5, O, c\}$.

$$\begin{cases} f_1 : C_0' + k_{p1}cC_0 - k_{m1}C_1 = 0, \\ f_2 : C_1' - k_{p1}cC_0 + (k_{m1} + A_1 + k_{p2}c) \times \\ \quad C_1 - k_{m2}C_2 - B_1C_3 = 0, \\ f_3 : C_2' - k_{p2}cC_1 + (k_{m2} + A_2 + A_0) \times \\ \quad C_2 - B_2C_4 - B_0O = 0, \\ f_4 : C_3' - A_1C_1 + (B_1 + k_{p3}c)C_3 - \\ \quad k_{m3}C_4 = 0, \\ f_5 : C_4' - A_2C_2 - k_{p3}cC_3 + (k_{m3} + \\ \quad B_2 + A_4)C_4 - B_4C_5 = 0, \\ f_6 : C_5' - A_4C_4 + (B_3 + B_4)C_5 - \\ \quad A_3O = 0, \\ f_7 : O' - A_0C_2 = \\ \quad B_3C_5 + (A_3 + B_0)O = 0, \\ f_8 : O - g(t) = 0. \end{cases} \quad (10)$$

根据式(10)可以得到如图8(a)的加权二部图 G , 以及对应的最大加权二部子图 $\bar{G}$, 如图8(b)所示. 系统的方程节点数为 $n = 8$, 变量节点数为 $m = 8$.

图8 脉冲传输模型的 G 和 $\bar{G}$ Fig. 8 G and $\bar{G}$ of pulse transmission model

使用算法1分析脉冲传输模型, 在第3步结束后,

$$\bar{M} = \{(f_1, c)\}, \quad k = 2,$$

算法进入while循环. k 从2增加到7的执行中, 过程Augment都返回了true, 使得

$$\bar{M} = \{(f_1, c), (f_2, C_1), (f_3, C_2), (f_4, C_3), \\ (f_5, C_4), (f_6, C_5), (f_7, O)\},$$

如图8(b)所示. 直到 k 取8时, 算法执行过程Augment返回false, 方程节点 f_8 被标记. 对应的方程微分后, G 转化为图9(a), 并得到相应的 $\bar{G}$ 如图9(b)所示.

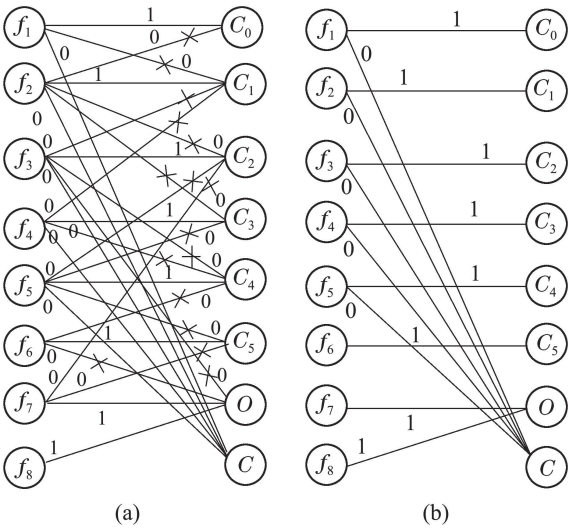


图 9 k 第1次取8后的 G 和 $\bar{G}$

Fig. 9 After k takes value 8 at the first time

算法再次进入while循环, 过程Augment仍然返回false, $k = 8$ 不变, 但方程节点 f_7 和 f_8 被标记, 对应的方程微分后, G 转化为图10(a), 然后得到新的 $\bar{G}$ 如图10(b)所示.

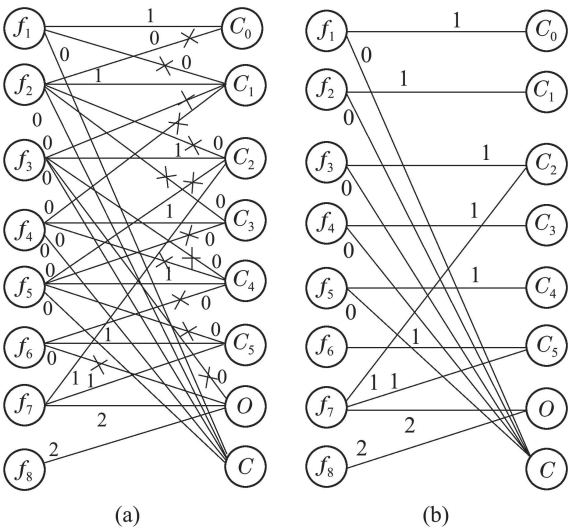


图 10 k 第2次取8后的 G 和 $\bar{G}$

Fig. 10 After k takes value 8 at the second time

算法又一次进入while循环, 过程Augment这一次执行返回了true, k 增加1变为9, $\bar{M} = \{(f_1, C_0), (f_2, C_1), (f_3, c), (f_4, C_3), (f_5, C_4), (f_6, C_5), (f_7, C_2), (f_8, O)\}$ 成为了完全匹配, 然后while循环结束, 算法1返回true并终止. 根据算法1的运行结果可以确定需要微分的方程和次数为(0, 0, 0, 0, 0, 0, 1, 2), 得到的增广系统中各个变量导数的最次阶为(1, 1, 1, 1, 1, 1, 2, 0). 若将各个变量的各阶导数视为不同的变量, 则得到的增广系统有11个方程和16个变量, 因此需要为系统指定5个初始值.

用算法1和Pantelides方法, 以及 Σ -方法对上述两个例子进行了结构化分析, 实验平台配置情况为: Win-

dows 7 x64操作系统, Intel i5-4200M CPU @2.50 GHz, 内存为8.00GB RAM, 最后的实验结果如表1所示.

表 1 上述两例的结构化分析时间

Table 1 Consumed time for above examples			
例子	Pantelides方法	Σ -方法	算法1
简单物理摆/s	0.00655	0.00045	0.00059
脉冲传输/s	0.00800	0.00164	0.00117

根据表1的数据可以对3种方法进行分析和对比. 即使不考虑微分代数系统由高阶转化为一阶需要花费的代价, Pantelides方法完成结构化分析消耗的时间也比算法1多. Pantelides方法只能处理一阶的微分代数系统, 而且转化后的系统规模也将成倍增长, 这不可避免会大大增加结构化分析需要花费的时间. Σ -方法与算法1在处理上述两个例子时的效率相差不大. 相关的其他实验表明, Σ -方法与算法1对小规模的微分代数系统进行结构化分析时执行的效率也很相近, 甚至在一些实验中 Σ -方法执行的效率还稍高一些. 但对于较大规模的问题, 算法1的执行效率更高, 而且随着系统规模的增大, 算法1的优势显得更为明显. Σ -方法在进行微分代数系统的结构化分析时, 首先需要计算签名矩阵的最大横截面, 然后才能根据最大横截面采用固定点迭代的方法来计算 c 和 d , 最后才能利用得到的 c 和 d 构造Jacobian矩阵来验证系统的结构化非奇异性, 这3个步骤分别都需要花费不少的时间, 对大规模的系统更是如此.

为了测试算法对大规模微分代数系统的处理效率, 采用了大量的随机矩阵作为加权二部图来执行算法1. 随机矩阵的规模分别为10, 50, 100, 200直到1000. 矩阵内的元素设置为从-1到3的整数, 其中-1表示方程节点和变量节点不相邻, 也就是方程中不存在某个变量; 整数0到3表示变量在某个方程中导数的最高次阶. 使用最小二乘法对得到的结果进行拟合, 算法1和 Σ -方法消耗的时间如图11所示.

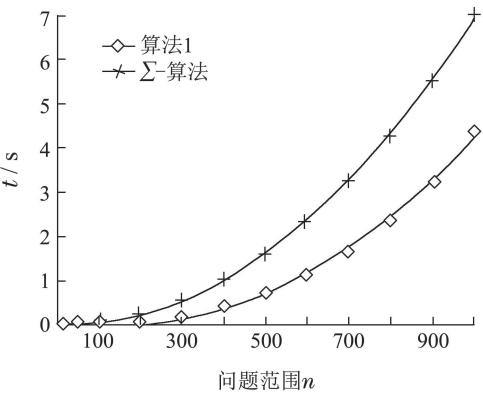


图 11 算法1与 Σ -方法消耗的时间

Fig. 11 The consumed time of structural analysis

图11的实验结果显示, 对大规模的微分代数系统, 算法1的处理效率明显优于 Σ -方法, 而且随着规模的继续增加, 这种优势还会变得更大. 如图12所示, 曲线开始部分表明, 在问题规模较小时, Σ -方法的效率可能会比算法1稍好一些, 这也正好验证了前面的分析.

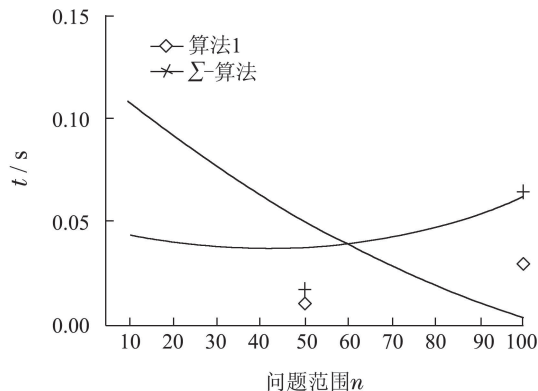


图 12 拟合曲线的开始部分

Fig. 12 The beginning part of the fitted curve

根据第3节的分析, 算法1的时间复杂度是 $O(n^2)$, 而 Σ -方法的时间复杂度为 $O(n^{2.5})$ ^[14], 图11正好揭示了这两者的差别. 秦方法利用三角分块化技术扩展了 Σ -方法, 并取得了较好的效果. 对算法1进行三角分块化处理, 也将成为下一阶段工作的重点.

5 结论(Conclusions)

本文对目前主要的微分代数系统结构化分析方法进行了较为深入的分析 and 比较, 然后定义了微分代数系统的加权二部图和最大加权二部子图, 并证明了微分代数系统加权二部图关于方程节点集的完全匹配与其最大横截面存在性之间的关系. 在此基础上, 本文提出了一种高效的结构化分析方法, 并以实例展示了该方法的实际应用过程, 最后通过实验和对比分析验证了该方法的实际应用效果.

参考文献(References):

- [1] ZHANG Xiuhua, ZHANG Qingling. Passivity for differential algebraic systems [J]. *Control Theory & Applications*, 2005, 22(5): 834 – 836.
(张秀华, 张庆灵. 微分代数系统的无源性 [J]. *控制理论与应用*, 2005, 22(5): 834 – 836.)
- [2] POGORELOV D. Differential – algebraic equations in multibody system modeling [J]. *Numerical Algorithms*, 1998, 19(1): 183 – 194.
- [3] ENGLAND R, GÓMEZ S, LAMOUR R. Expressing optimal control problems as differential algebraic equations [J]. *Computers & Chemical Engineering*, 2005, 29(8): 1720 – 1730.
- [4] BÜSKENS C, GERDTS M. Differentiability of consistency functions for DAE systems [J]. *Journal of Optimization Theory and Applications*, 2005, 125(1): 37 – 61.
- [5] KUNKEL P, MEHRMANN V. Differential-algebraic equations: analysis and numerical solution [J]. *Ems Textbooks in Mathematics*, 2006, 19(8): 1218 – 1228.
- [6] LINDA P. Differential/algebraic equations are not ODE'S [J]. *Siam Journal on Scientific & Statistical Computing*, 1982, 3(3): 367 – 384.
- [7] GEAR C W, PETZOLD L R. ODE methods for the solution of differential/algebraic systems [J]. *Siam Journal on Numerical Analysis*, 1982, 21(4): 716 – 728.
- [8] PANTELIDES C C. The consistent initialization of differential-algebraic systems [J]. *Siam Journal on Scientific & Statistical Computing*, 1988, 9(2): 213 – 231.
- [9] GEAR C W. Differential-algebraic equation index transformations [J]. *Siam Journal on Scientific & Statistical Computing*, 1988, 9(1): 39 – 47.
- [10] MATTSOON S E, DERLIND G. Index reduction in differential-algebraic equations using dummy derivatives [J]. *Siam Journal on Scientific Computing*, 1993, 14(3): 677 – 692.
- [11] PRYCE J D. A simple structural analysis method for DAEs [J]. *Bit Numerical Mathematics*, 2001, 41(2): 364 – 394.
- [12] PRYCE J D, NEDIKOV N S, TAN G. DAESA – A matlab tool for structural analysis of differential-algebraic equations: theory [J]. *Acm Transactions on Mathematical Software*, 2015, 41(2): 1 – 20.
- [13] DUFF I S. On algorithms for obtaining a maximum transversal [J]. *Acm Transactions on Mathematical Software*, 1981, 7(3): 315 – 330.
- [14] QIN X, TANG J, FENG Y, et al. Efficient index reduction algorithm for large scale systems of differential algebraic equations [J]. *Applied Mathematics & Computation*, 2016, 277(C): 10 – 22.
- [15] LÖTSTEDT P, PETZOLD L. Numerical solution of nonlinear differential equations with algebraic constraints i: convergence results for backward differentiation formulas [J]. *Mathematics of Computation*, 1986, 46(174): 491 – 516.
- [16] SCHOLZ L, STEINBRECHER A. Regularization of DAEs based on the signature method [J]. *BIT Numerical Mathematics*, 2016, 56(1): 319 – 340.
- [17] SCHOLZ L, STEINBRECHER A. Structural-algebraic regularization for coupled systems of DAEs [J]. *BIT Numerical Mathematics*, 2016, 56(2): 777 – 804.
- [18] SAIP H A B, LUCCHESI C. Matching algorithm for bipartite graph [R]. Technical Report DCC-03/93. Campinas, Brazil: Universidade Estadual de Campinas, 1993.

作者简介:

李光远 (1973–), 男, 博士研究生, 研究方向为微分代数系统求解以及CPS, E-mail: li_guang_yuan@sina.com;

冯勇 (1965–), 男, 博士, 研究员, 研究方向为计算机应用以及自动推理与机器证明, E-mail: yongfeng@cigit.ac.cn.