Structural index reduction algorithms for differential algebraic equations via fixed-point iteration

TANG Juan¹,³, WU WenYuan², QIN XiaoLin¹ & FENG Yong²,*

¹Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, China; ²Chongqing Key Lab of Automated Reasoning and Cognition CIGIT, CAS, Chongqing 400714, China; ³University of Chinese Academy of Sciences, Beijing 100049, China
Email: tj-123123@163.com, wuwenyuan@cigit.ac.cn, qinxl@casit.ac.cn, yongfeng@cigit.ac.cn
Received June 13, 2014; accepted XX XX, 2014

Abstract Motivated by Pryce’s structural index reduction method for differential algebraic equations (DAEs), we give a complete theoretical analysis about existence and uniqueness of the optimal solution of index reduction problem, and then show the termination and complexity of the fixed-point iteration algorithm. Based on block upper triangular structure of system, we propose the block fixed-point iteration method for DAEs with its complex analysis.

Keywords differential algebraic equations, structural analysis, index reduction, linear programming, fixed-point iteration, block triangular forms

MSC(2010) 34A09, 65L80, 65F50, 90C05, 90C27, 90C06

Citation: Tang J, Wu W Y, Qin X L, Feng Y. Structural index reduction algorithms for differential algebraic equations via fixed-point iteration. Sci China Math, 2014, 12 pages, doi: 10.1007/s11425-000-0000-0

1 Introduction

Differential algebraic equations (DAEs) systems arise naturally in modeling many dynamical systems, such as electric circuits, mechanical systems, and spacecraft dynamics. Based on unified multi-domain modeling techniques e.g. Modelica [9], computers can automatically produce thousands of DAEs. The generated DAEs have many interesting characteristics, such as large scale, high index, block structures, which are the major motivations of our work in this paper. It is well known that a direct numerical simulation without index reduction may not be possible or may provide a bad result [5, 12]. Here the index of a DAE system is a key notion in the theory for measuring the distance from the given system with a singular Jacobian to the corresponding ordinary differential equations with a nonsingular Jacobian. Various index concepts exist in the theory of DAEs; and the one related to the structural analysis approach is the “structural index”, which is defined in [23]. For other indices, we refer the interested readers to [1,7,17]. High-index DAE systems usually need differentiations to reveal all the system’s constraints, which are crucial to determine consistent initial conditions. This procedure is the called “index reduction” of DAEs. For applications of high index DAEs, see [16]. Identifying all hidden constraints on formal power series solutions in the neighborhood of a given point is a key step to construct nonsingular Jacobian of a DAE system for numerical integration. Thus, for DAE systems, index reduction is fundamental and unavoidable.

*Corresponding author
In the previous work on DAE index reduction for general DAE systems, Campbell and Gear gave a
derivative-array method to reduce DAEs in [2], which may not be applicable to large-scale nonlinear
systems. Pantelides in [11] introduced a graph-oriented method which gives a systematic way to reduce
high-index systems of differential algebraic equations to lower index, by selectively adding differentiated
forms of the equations already present in the system. Pryce developed structural analysis method in [14].
This approach is based on solving an assignment problem, which can be formulated as an integer linear
programming problem. The idea was generalized to a class of partial differential algebraic equations by
Wu et al [19]. Recently, Pryce et al. in [10, 15] generalized the structural analysis method to the DAE
systems with block triangular forms (BTF), and indicated that the difference between global and local
offsets is constant on each block signature matrix of DAEs with irreducible BTF. Unfortunately, their
method may not obtain the ‘smallest’ optimal as shown in our Example 3.8 mentioned below. Our work
can remedy these drawbacks.

The rest of this paper is organized as follows. Section 2 briefly reviews Pryce’s structural analysis
method, firstly. Then we rigorously prove the existence and uniqueness of smallest optimal solution of
Problem 2.4 and show the termination of fixed-point iteration algorithm which is not given in [14]. In
addition, we also give the time complexity of the algorithm which is $O(n^3 + ||c^*||_1 \cdot n^2)$ due to Theorem
2.7, where $n$ is the size of the system. Section 3 first introduces the block triangular forms (BTF) for
large scale DAE systems. Based on our fixed-point iteration method with parameter, a block fixed-point
iteration algorithm is proposed to find the unique smallest dual-optimal pair of the systems with BTF,
and its time complexity is $O(\ell \sum_{i=1}^{\ell} n_i^3 + ||c^*_i||_1 \cdot n_i^2)$ by Theorem 3.7, where $\ell \sum_{i=1}^{\ell} n_i = n$ and $\ell$
is the number of the blocks on the diagonal. It is usually much better than the cost $O(n^3 + ||c^*||_1 \cdot n^2)$ without taking
the advantages of the structure, when $\ell$ is large. Conclusions are made in the last section.

2 Theoretical foundation for fixed-point iteration method

First we give a brief review about the main steps of Pryce’s structural analysis method [14]. We consider
a DAE system $f = 0$ in $n$ dependent variables $x_j = x_j(t)$ with $t$ a scalar independent variable, of the form

$$f = \{f_i(t, the \ x_j \ and \ derivatives \ of \ them) : 1 \leq i \leq n\}.$$  \hspace{1cm} (2.1)

Step 1. Form the $n \times n$ signature matrix $\Sigma = (\sigma_{ij})$ of the DAE, where

$$\sigma_{ij} = \begin{cases} 
\text{highest differential order of } x_j \text{ appears in equation } f_i, \\
-\infty, \text{ if } x_j \text{ does not occur in } f_i. 
\end{cases}$$

Step 2. Solve an assignment problem (AP) to find a highest value transversal (HVT) $T$, which is a
subset of sparsity pattern $S$ with $n$ finite entries and describes just one element in each row and each
column, such that $\sum \sigma_{ij}$ is maximized and finite. The sparsity pattern $S$ of $\Sigma$ is defined as:

$$S = \text{sparse}(\Sigma) = \{(i, j) : \sigma_{ij} > -\infty\}.$$ \hspace{1cm} (2.2)

This can be formulated as a Linear Programming Problem (LPP), the Primal is:

$$\max_{\xi} \ z = \sum_{(i,j) \in S} \sigma_{ij} \xi_{ij},$$

s.t. $\sum_{j : (i,j) \in S} \xi_{ij} = 1$ for each $i$, $\sum_{i : (i,j) \in S} \xi_{ij} = 1$ for each $j$, $\xi_{ij} \geq 0$ for $(i,j) \in S$. \hspace{1cm} (2.3)

The problem is equivalent to finding a maximum-weight perfect matching in a bipartite graph whose
incidence matrix is the signature matrix, and can be solved by Kuhn-Munkres algorithm [18] whose time
complexity is $O(n^3)$. 

Step 3. Determine the offsets of the problem, which are the vectors \( c = (c_i)_{1 \leq i \leq n} \) and \( d = (d_j)_{1 \leq j \leq n} \) such that \( d_j - c_i \geq \sigma_{ij} \), for all \( 1 \leq i \leq n, 1 \leq j \leq n \), and \( d_j - c_i = \sigma_{ij} \) when \( (i,j) \in T \).

This problem can be formulated as the dual of (2.3) in the variables \( c = (c_1, c_2, \cdots, c_n) \) and \( d = (d_1, d_2, \cdots, d_n) \). The Dual is defined as follows:

\[
\min_{c,d} \quad z = \sum_j d_j - \sum_i c_i, \\
\text{s.t.} \quad d_j - c_i \geq \sigma_{ij} \text{ for all } (i,j), \\
c_i \geq 0 \text{ for all } i. 
\]

(2.4)

To verify the success of the index reduction, we need to check whether the \( n \times n \) system Jacobian matrix \( J \) where

\[
J_{ij} = \begin{cases} 
\frac{\partial f_i}{\partial (d_j - c_i)_{\text{th derivative of } x_j}} & \text{if this derivative is present in } f_i, \\
0 & \text{otherwise}
\end{cases}
\]

is nonsingular.

In this paper the structural index is then defined as:

\[
\nu = \max_i c_i + \begin{cases} 
0 & \text{for all } d_j > 0, \\
1 & \text{for some } d_j = 0.
\end{cases}
\]

(2.5)

In order to determine the canonical (smallest) offsets of DAEs using fixed-point iteration algorithm [14], we introduce some necessary definitions, firstly.

Given \( \Sigma \) and a corresponding transversal \( T \), for \( \forall \ c = (c_i)(\in \mathbb{R}^n) \), we define a mapping

\[
D(c) = (d_j), \text{ where } d_j = \max_i (\sigma_{ij} + c_i),
\]

and for \( \forall \ d = (d_j)(\in \mathbb{R}^n) \), we define a mapping

\[
C_T(d) = (c^*_i), \text{ where } c^*_i = d_j - \sigma_{i,j}, \ (i,j) \in T.
\]

Furthermore, we define the composition mapping \( \phi_T(c) = C_T(D(c)) \) from \( \mathbb{R}^n \) to \( \mathbb{R}^n \). Then we have:

**Algorithm 1** Fixed-point iteration algorithm

**Input:**

- \( \Sigma \) is the signature matrix of a given DAE system
- \( T \) is the HVT of \( \Sigma \) computed by Kuhn-Munkres algorithm

**Output:**

- \( c \) and \( d \)

1: Set \( c' \leftarrow 0 \)
2: Set \( d \leftarrow D(c') \)
3: Set \( c \leftarrow C_T(d) \)
4: while \( c \neq c' \) do
   5: Set \( c' \leftarrow c \)
   6: Set \( d \leftarrow D(c) \)
   7: Set \( c \leftarrow C_T(d) \)
8: end while
9: return \( c, d \)

In order to give a rigorous proof of the existence and uniqueness of smallest offsets in Problem 2.3 and the termination of Algorithm 1 we introduce some definitions and lemmas as follows.
Lemma 2.1. (See [14]).

(i) If the Primal has a feasible solution, then it has a basic feasible solution (BFS). At such a solution the $\xi_{ij}$ are 1 on some transversal $T$ and 0 elsewhere. The corresponding objective function value is $\Sigma_{(i,j)\in T} \sigma_{ij}$, denoted by $||T||$. The optimum is achieved at a BFS.

(ii) The following results are equivalent.

(a) The AP is regular.

(b) The Primal has a feasible solution.

(c) Primal and Dual have a common, finite optimal value given by

$$z = \sum d_j - \sum c_i = ||T||.$$  \hfill (2.6)

(iii) (Principle of complementary slackness) Given a Primal BFS, i.e., a transversal $T$, and a Dual feasible solution $c, d$, the following are equivalent:

(a) $T$ is an HVT, and $c$ and $d$ are optimal for the Dual.

(b) $d_j - c_i = \sigma_{ij}$, for each $(i, j) \in T$.

Lemma 2.2. (See [14]). Assume that $T$ is HVT, then $c$ is optimal of Problem 2.4 if and only if $c$ is the non-negative fix-points of $\phi_T$, that is, $\phi_T(c) = c$.

Definition 2.3. For a given $\Sigma$ and a corresponding $T$, the vector set $VC$ is defined as:

$$VC(\Sigma, T) = \{c \in \mathbb{R}^n | \phi_T(c) = c, c \succ 0\}.$$ \hfill (2.7)

The optimal-dual set of Problem 2.4 is $VC$ by Lemma 2.2. Furthermore, we have:

Lemma 2.4. For a given $\Sigma$ matrix, the optimal-dual set $VC$ is independent of the choice of HVT, that is, $VC(T) = VC(T')$ for any two HVT $T$ and $T'$.

Proof. For any $c \in VC(T)$, define $d = D(c)$. Then $c$ and $d$ are optimal-dual by Lemma 2.2. Note that $T'$ is HVT. By Lemma 2.1(iii), obtain

$$d_j - c_i = \sigma_{ij},$$ \hfill (2.8)

for each $(i, j) \in T'$. According to (2.8), we have

$$c = C_{T'}(d) = C_{T'}(D(c)) = \phi_{T'}(c).$$ \hfill (2.9)

Therefore, $c$ is also the non-negative fix-points of $\phi_{T'}$, that is, $c \in VC(T')$. Conversely, we can easily prove $VC(T') \subseteq VC(T)$ with the similar principle above.

Now, we define a semi-ordering of vectors set. For $\forall a, b \in \mathbb{R}^n$, $a \prec b$ if $a_i \leq b_i$ for each $i$. Using the above results we can prove the existence and uniqueness of the unique smallest optimal solution for the Dual problem.

Lemma 2.5. Assume that the $\Sigma$ matrix of given DAE systems in Problem 2.4 contains a transversal $T$ at least, then there exists a unique smallest dual-optimal pair $c^*$ and $d^*$.

Proof. The $\Sigma$ matrix contains a transversal $T$ at least, then there must exist a HVT $T$ from the finiteness of transversal. From the Lemma 2.1 assume $T$ is any HVT. According to the primal-dual principle, dual-optimal pair $c$ and $d$ must exist, that is, $VC = VC(T)$ is a non-empty set by Lemma 2.2. Moreover, It is easy to know that for any non-negative vector $\Theta = (\theta, \theta, ..., \theta)$, $c + \Theta$ and $d + \Theta$ is also dual-optimal. Then $VC$ is a infinite set. Define

$$VC_1 = \{||c||_1 : c \in VC\} \text{ and } \alpha = \inf\{VC_1\}(\geq 0).$$ \hfill (2.10)

(The existence of smallest dual-optimal) In fact, the coefficients of all the constraint equations in Problem 2.4 are 1 or $-1$, and each $\sigma_{ij} \in \Sigma$ is integer. Thus, all the vertices of the feasible region in
Linear programming are integer. Then the smallest dual-optimal also are integer. Therefore, there exists \( c^* \in VC \) such that \( \alpha = ||c^*||_1 \), that is, \( c^* \) and \( d^* = D(c^*) \) is the smallest dual-optimal.

(The uniqueness of smallest dual-optimal) Assume that there are two different smallest dual-optimal pair \( c^*, d^* \) and \( c^0, d^0 \) such that
\[
\alpha = ||c^*||_1 = ||c^0||_1. \tag{2.11}
\]
There must exist \( i_0 \in \{1, 2, ..., n\} \) such that \( c^*_{i_0} \neq c^0_{i_0} \). According to the following rules, construct vector pair \( c^\alpha \) and \( d^\alpha \). For given HVT \( T \) and each \( i \), if \( c^*_i \geq c^o_i \), define \( c^\alpha_i = c^*_i \) and \( d^\alpha_j = d^o_j \) such that \( (i, j) \in T \); otherwise, define \( c^\alpha_i = c^*_i \) and \( d^\alpha_j = d^* \).

Firstly, it is verified that \( c^\alpha \) and \( d^\alpha \) are the Dual feasible solution. By the definition of \( c^\alpha \) and \( d^\alpha \), for each \( j \in \{1, 2, ..., n\} \), obtain
\[
d^\alpha_j = d^o_j = c^*_i + \sigma_{ij} - c^\alpha_j \geq c^\alpha_i + \sigma_{ij}, i = 1, 2, ..., n; \tag{2.12}
\]
and
\[
d^\alpha_j = d^o_j = c^*_i + \sigma_{ij} - c^\alpha_j \geq c^\alpha_i + \sigma_{ij}, i = 1, 2, ..., n. \tag{2.13}
\]
Then together with (2.12) and (2.13), for each \( j \in \{1, 2, ..., n\} \), get
\[
d^\alpha_j \geq c^\alpha_i + \sigma_{ij}, i = 1, 2, ..., n. \tag{2.14}
\]

That is \( c^\alpha \) and \( d^\alpha \) are the Dual feasible solution.

Furthermore, note that \( T \) is HVT, \( c^*, d^* \) and \( c^0, d^0 \) are dual-optimal pair. By Lemma 2.4(iii), obtain
\[
d^\alpha_j - c^\alpha_j = \sigma_{ij}, \quad d^\alpha_j - c^\alpha_j = \sigma_{ij}, \tag{2.15}
\]
for each \( (i, j) \in T \). By (2.15), we have
\[
d^\alpha_j - c^\alpha_j = \sigma_{ij}, \tag{2.16}
\]
for each \( (i, j) \in T \). Combining (2.14) and (2.16), it is indicated that \( c^\alpha \) and \( d^\alpha \) are also the dual-optimal by Lemma 2.4(iii), that is, \( c^\alpha \in VC \). But \( ||c^\alpha||_1 < ||c^\alpha||_1 = \alpha \) or \( ||c^\alpha||_1 < ||c^\alpha||_1 = \alpha \), which is in conflict with (2.10). Therefore, the smallest dual-optimal is unique.

('Smallest' in '≺' sense) Set \( c^* \) and \( d^* \) are the smallest dual-optimal, \( c^0 \) and \( d^0 \) are any dual-optimal, then obtain \( \alpha = ||c^*||_1 \leq ||c^0||_1 \). Assume that there exists \( i_0 \in \{1, 2, ..., n\} \) such that \( c^*_{i_0} > c^0_{i_0} \). We can construct the new dual-optimal \( c^{\alpha}(\in VC) \) and \( d^{\alpha} \) such that \( ||c^{\alpha}||_1 < ||c^*||_1 = \alpha \) by the method described above, which is also in conflict with (2.10). So obtain \( c^* < c^0 \), and \( d^* = D(c^*) ≺ D(c) = d^0 \) \( \square \).

According to the above lemmas, we can prove the termination of fixed-point iteration algorithm and analyze its complexity.

**Lemma 2.6.** The fixed-point iteration algorithm can find the unique smallest dual-optimal pair \( c^* \) and \( d^* \) of Problem 2.4 by at most \( ||c^*||_1 + 1 \) iterations.

**Proof.** Set \( c^{(1)} = \phi(0)(\succ 0) \), \( c^{(k)} = \phi(c^{(k-1)}) = \phi^k(0) \) for \( k \in \mathbb{N}^+ \). It is verified that \( \phi(- \phi_T) \) is monotone operator from \( \mathbb{R}^n \) to \( \mathbb{R}^n \), that is, if \( c \prec c^* \), \( \phi(c) \prec \phi(c^*) \). So \( \{c^{(k)}\} \) is an increasing sequence in “≺” sense, and \( ||c^{(k)}||_1 \) is also an increasing sequence. Note that \( T \) is HVT, then exist the unique smallest dual-optimal pair \( c^*(\succ 0) \) and \( d^* \) by Lemma 2.5. According to the monotonicity of \( \phi \), obtain
\[
c^{(k)} \prec c^*, \text{ for } k \in \mathbb{N}^+, \tag{2.17}
\]
and then \( ||c^{(k)}||_1 \leq ||c^*||_1 \). It is indicated that \( \{||c^{(k)}||_1\} \) is bounded. Based on bounded monotonic principle, exists \( \beta \) such that
\[
||c^{(k)}||_1 \rightarrow \beta(\leq ||c^*||_1), k \rightarrow \infty. \tag{2.18}
\]
Assume \( \{||c^{(k)}||_1\} \) is strictly increasing sequence. For all \( c^{(k)} \) are integer, we have \( ||c^{(k)}||_1 \geq k \). So \( \beta = +\infty \), which is in conflict with (2.18). It is shown that exists \( m \in \mathbb{N}^+ \) such that \( ||c^{(m)}||_1 = ||c^{(m+1)}||_1 \). Moreover, obtain \( c^{(m)} = c^{(m+1)} \) and \( c^{(l)} = c^{(m)} \) for \( \forall l > m \). Set \( c^0 = c^{(m)} \), by (2.17), we have \( c^0 \prec c^* \).
In addition, $c^*$ is the smallest dual-optimal and $c^o$ is the non-negative fix-point, then we get $c^* \prec c^o$ by Lemma 2.2 and 2.3. Therefore, $c^* = c^o$ and $d^* = D(c^*)$. Note that the sequence $\|c^{(k)}\|_1$ increases by 1 on each iteration at most. So the Algorithm 1 finds the smallest dual-optimal pair $c^*$ and $d^*$ by at most $\|c^*\|_1 + 1$ iterations.

Now, we are able to give the main theorem below.

**Theorem 2.7.** If the $\Sigma$ matrix of given DAE systems in Problem 2.4 contains some transversal $T$, then the unique smallest dual-optimal pair $c^*$ and $d^*$ can be found in time $O(n^3 + \|c^*\|_1 \cdot n^2)$ via fixed-point iteration algorithm.

**Proof.** It can be easily proved by Lemma 2.5 and 2.6.

**Example 2.8.** Consider the application of the Algorithm 1 to nonlinear DAE system $f = (f_1, f_2, f_3) = 0$ in three dependent variables $x_1(t), x_2(t), x_3(t)$ with known forcing functions $u_i(t)(i = 1, 2, 3)$:

\[
\begin{align*}
f_1 &= \ddot{x}_1 + x_1 + u_1(t) \\
f_2 &= \ddot{x}_2 + x_2 + u_2(t) \\
f_3 &= x_1^2 + x_2^2 + u_3(t)
\end{align*}
\]

The corresponding signature matrix is

\[
\Sigma = \begin{pmatrix} 2^* & 0 \\ 1 & 0^* \\ 0 & 0^* \end{pmatrix},
\]

where we have already marked the HVT with asterisks, and the elements in the blanks of $\Sigma$ are $-\infty$. We give the main process of Algorithm 1 below,

\[
\begin{align*}
\Sigma & \Rightarrow \begin{bmatrix} x_1 & x_2 & x_3 \\ f_1 & 2^* & 0 \\ f_2 & 1 & 0^* \\ f_3 & 0 & 0^* \end{bmatrix}, \\
& \Rightarrow \begin{bmatrix} x_1 & x_2 & x_3 \\ f_1 & 2^* & 0 \\ f_2 & 1 & 0^* \\ f_3 & 0 & 0^* \end{bmatrix} \\
& \Rightarrow \begin{bmatrix} x_1 & x_2 & x_3 \\ f_1 & 2^* & 0 \\ f_2 & 1 & 0^* \\ f_3 & 0 & 0^* \end{bmatrix} ^{(i)} \\
& \Rightarrow \begin{bmatrix} x_1 & x_2 & x_3 \\ f_1 & 2^* & 0 \\ f_2 & 1 & 0^* \\ f_3 & 0 & 0^* \end{bmatrix} ^{(i)} \\
& \Rightarrow \begin{bmatrix} x_1 & x_2 & x_3 \\ f_1 & 2^* & 0 \\ f_2 & 1 & 0^* \\ f_3 & 0 & 0^* \end{bmatrix} ^{(i)} ,
\end{align*}
\]

where $c^{(i)}$, $c^{(i)}$, and $d^{(i)}$ mean the $i$th iteration for $c'$, $c$ and $d$, respectively. Therefore, obtain the smallest offsets $c = (0, 0, 1)$ and $d = (2, 1, 0)$ of the DAE systems.

**Remark 2.9.** We can easily know that if $(c, d)$ is a solution of Problem 2.4 then $(c + \Theta, d + \Theta)$ is also a solution for any non-negative constant vector $\Theta = (\theta, \ldots, \theta)$. Conversely, this is not correct. $c = (0, 0, 1), d = (2, 1, 0)$ and $c' = (0, 1, 2), d' = (2, 2, 1)$ are solutions of the above DAE, but there does not exists constant between $c$ and $c'$ or $d$ and $d'$. 
3 Block fixed-point iteration method

When dealing with DAE systems of large dimensions, an important manipulation is the block triangularization of the system \[ \Sigma \] matrix, which allows to decompose the overall system into subsystems which can be solved in sequence. Similarly, considering the index reduction for large-scale systems, it is necessary to linearization of the system \[ \Sigma \], which allows to decompose the overall system into subsystems which can be dealt with.

When dealing with DAE systems of large dimensions, an important manipulation is the block triangularization of the system \[ \Sigma \] matrix, which allows to decompose the overall system into subsystems which can be solved in sequence. Similarly, considering the index reduction for large-scale systems, it is necessary to linearization of the system \[ \Sigma \], which allows to decompose the overall system into subsystems which can be handled.

When dealing with DAE systems of large dimensions, an important manipulation is the block triangularization of the system \[ \Sigma \] matrix, which allows to decompose the overall system into subsystems which can be solved in sequence. Similarly, considering the index reduction for large-scale systems, it is necessary to linearization of the system \[ \Sigma \], which allows to decompose the overall system into subsystems which can be dealt with.

When dealing with DAE systems of large dimensions, an important manipulation is the block triangularization of the system \[ \Sigma \] matrix, which allows to decompose the overall system into subsystems which can be solved in sequence. Similarly, considering the index reduction for large-scale systems, it is necessary to linearization of the system \[ \Sigma \], which allows to decompose the overall system into subsystems which can be handled.

When dealing with DAE systems of large dimensions, an important manipulation is the block triangularization of the system \[ \Sigma \] matrix, which allows to decompose the overall system into subsystems which can be solved in sequence. Similarly, considering the index reduction for large-scale systems, it is necessary to linearization of the system \[ \Sigma \], which allows to decompose the overall system into subsystems which can be dealt with.

3.1 Fixed-point iteration method with parameter

The Dual Problem with parameter \( \Sigma \) matrix with BTF below

\[
M = \begin{pmatrix}
M_{11} & M_{1,2} & \cdots & M_{1,\ell} \\
M_{2,1} & M_{2,2} & \cdots & M_{2,\ell} \\
\vdots & \vdots & \ddots & \vdots \\
M_{\ell,1} & M_{\ell,2} & \cdots & M_{\ell,\ell}
\end{pmatrix},
\]

where the elements in the blanks of \( M \) are \( -\infty \), and diagonal matrix \( M_{i,i} \) is square and irreducible \[15\], for \( i = 1, 2, \ldots, \ell \).

In the following section, assume the given DAE systems are structurally nonsingular, meaning that the \( \Sigma \) matrix of the systems exists a transversal, then obtain the BTF \[3.1\] of the \( \Sigma \) matrix. The main idea of block fixed-point iteration method for \( \Sigma \) matrix with BTF is to use the fixed-point iteration method with parameter mentioned below to process each diagonal matrix in block upper triangulated signature matrix from top to bottom in sequence. We give the fixed-point iteration method with parameter, firstly.

3.1 Fixed-point iteration method with parameter

The Dual Problem with parameter \( p \) is defined as follows:

**Definition 3.1.** The Dual Problem with parameter \( p \) is defined via:

\[
\begin{align*}
\min_{c,d} & \quad z = \sum_j d_j - \sum_i c_i, \\
\text{s.t.} & \quad d_j - c_i \geq \sigma_{ij} \text{ for all } (i, j), \\
& \quad d_j \geq p_j, \\
& \quad c_i \geq 0 \text{ for all } i.
\end{align*}
\]

(3.2)

For any given parameter \( p \), obtain the fixed-point iteration algorithm with parameter (PFPIA) below just by modified the fixed-point iteration algorithm.

**Lemma 3.2.** Let \( p \) is any given parameter. Assume that the \( \Sigma \) matrix in the Problem contains a transversal \( T \) at least, then there exists a unique smallest dual-optimal pair \( c^* \) and \( d^* \) such that

\[
c^* = \min \{ c | D(c) \succ p, \ c \in VC \}.
\]

Moreover, if the used transversal \( T \) is HVT, the Algorithm finds the unique smallest dual-optimal pair \( c^* \) and \( d^* \) by at most \( ||c^*||_1 - ||\max(C_T(p), 0)|| \) iterations.

**Proof.** Just modify the proof of Lemma properly.

**Remark 3.3.** If \( p_j \leq \max_i \sigma_{ij} \), for \( j = 1, 2, \ldots, n \), and \( d_j \geq \max_i \sigma_{ij} \) derived from Problem (3.2), for \( j = 1, 2, \ldots, n \). So obtain \( d_j \geq p_j \), for \( j = 1, 2, \ldots, n \), that is, the constraint condition \( d_j \geq p_j \) in problem (3.2) can be deleted. Therefore, the Problem turns into Problem (2.4).

**Example 3.4.** Consider the application of the Algorithm to nonlinear DAE systems \( f = (f_4, f_5, f_6) = 0 \) in three dependent variables \( x_4(t), x_5(t), x_6(t) \) with known forcing functions \( u_i(t)(i = 4, 5, 6) \):

\[
\begin{align*}
f_4 &= \dot{x}_4 + x_6 + u_4(t) \\
f_5 &= \dot{x}_5 + x_6 + u_5(t) \\
f_6 &= x_4^2 + x_5^2
\end{align*}
\]

and the given parameter is \( p = (0, 0, 2) \).
\textbf{Algorithm 2} Fixed-point iteration algorithm with parameter (PFPIA)

\textbf{Input:}
\begin{itemize}
  \item $\Sigma$ is signature matrix for given DAE systems
  \item $T$ is HVT of $\Sigma$ by Kuhn-Munkres algorithm
  \item $p$ is given parameter vector
\end{itemize}

\textbf{Output:}
\begin{itemize}
  \item $c$ and $d$
\end{itemize}

1: Set $c' \leftarrow 0$
2: Set $c' \leftarrow C_T(p)$
3: Set $c' \leftarrow \max\{c', 0\}$
4: Set $d \leftarrow D(c')$
5: Set $c \leftarrow C_T(d)$
6: while $c \neq c'$ do
7: Set $c' \leftarrow c$
8: Set $d \leftarrow D(c)$
9: Set $c \leftarrow C_T(d)$
10: end while
11: return $c, d$

We give below the main process of Algorithm 2.

\[
\begin{align*}
\Sigma \Rightarrow & \begin{bmatrix} f_4 & 2^* & 0 & 0 \\ f_5 & 1 & 0^* & 2 \\ f_6 & 0 & 0^* & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} f_4 & 2^* & 0 & 0 \\ f_5 & 1 & 0^* & 2 \\ f_6 & 0 & 0^* & 0 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\Sigma \Rightarrow & \begin{bmatrix} f_4 & 2^* & 0 & 0 & 0 \\ f_5 & 1 & 0^* & 2 & 2 \\ f_6 & 0 & 0^* & 0 & 3 \\ \end{bmatrix} \Rightarrow \begin{bmatrix} f_4 & 2^* & 0 & 0 & 0 \\ f_5 & 1 & 0^* & 2 & 2 \\ f_6 & 0 & 0^* & 0 & 3 \\ \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\Sigma \Rightarrow & \begin{bmatrix} f_4 & 2^* & 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} f_4 & 2^* & 0 & 0 & 0 \\ f_5 & 1 & 0^* & 2 & 2 \\ f_6 & 0 & 0^* & 0 & 3 \\ \end{bmatrix}
\end{align*}
\]

where $c'^{(i)}$, $c^{(i)}$ and $d^{(i)}$ mean the $i$th iteration for $c'$, $c$ and $d$, respectively. Then obtain the smallest offsets $c = (1, 2, 3)$ and $d = (3, 3, 2)$ for the DAEs.

\section{3.2 Block fixed-point iteration method}

The given DAE systems are structurally nonsingular, obtain the $\Sigma$ matrix $M$ of Problem 2.4 with block triangular form $(3.1)$, and \( \sum n_i = n \), where $n_i$ is the order of $M_{ii}$, $i = 1, 2, \ldots, \ell$. Without loss of generality, assume the associated graph of $M$ is connected here. In order to find the unique smallest dual-optimal, we give some necessary symbols as follows.
Let the parameter vector is \( \mathbf{p} = (p_1, p_2, ..., p_\ell) \) with \( \ell \) sections, the dual-optimal are \( \mathbf{c} = (c_1, c_2, ..., c_\ell) \) and \( \mathbf{d} = (d_1, d_2, ..., d_\ell) \), where the dimension of \( p_i, c_i \) and \( d_i \) are \( n_i \) for \( i = 1, 2, ..., \ell \). For given \( n \times r \) order matrix \( B \) and \( B' \leq n \) order vector \( \mathbf{q} \), \( r \) order vector \( \mathbf{w} \), the mapping \( B' = \text{RowAdd}(B, \mathbf{q}) \) is defined as \( B'_{i,j} = B_{i,j} + q_i \), for \( i = 1, 2, ..., n, j = 1, 2, ..., r \); the mapping \( \mathbf{w} = \text{ColumnMax}(B) \) is defined via \( w_j = \max_{i \in \{1, 2, ..., n\}} B_{i,j} \), for \( j = 1, 2, ..., r \). Then we give block fixed-point iteration algorithm in the following.

### Algorithm 3 Block fixed-point iteration algorithm

**Input:**
- \( M \) is \( \Sigma \) matrix of given DAE systems with BTF

**Output:**
- \( c = (c_1, c_2, ..., c_\ell) \) and \( d = (d_1, d_2, ..., d_\ell) \)
1. Set \( \mathbf{p} = (p_1, p_2, ..., p_\ell) \), \( p_j = 0, j = 1, 2, ..., \ell \).
2. Set \( c = (c_1, c_2, ..., c_\ell) \), \( c_j = 0, j = 1, 2, ..., \ell \)
3. Set \( d = (d_1, d_2, ..., d_\ell) \), \( d_j = 0, j = 1, 2, ..., \ell \).
4. Get \( (c_1, d_1) = \text{PFPIA}(M_{11}, \mathbf{p}_1) \).
5. for \( i \) from 2 to \( \ell \) do
6. Update: \([M_{i-1,1}, ..., M_{i-1,\ell}] ← \text{RowAdd}([M_{i-1,1}, ..., M_{i-1,\ell}], c_{i-1}) \).
7. Update: \((p_i, ..., p_\ell) ← \text{ColumnMax}((M_{1,i}, M_{1,\ell}), ..., (M_{i-1,i}, M_{i-1,\ell})) \).
8. Get \( (c_i, d_i) = \text{PFPIA}(M_{ii}, \mathbf{p}_i) \).
9. end for
10. return \( c, d \)

In order to obtain a complete theoretical analysis of block fixed-point iteration method, we give some necessary lemmas, firstly.

**Lemma 3.5.** (See [15]). For given \( \Sigma \) matrix \( M \) with BTF, if \( T_i \) is HVT of \( M_{ii} \), then \( T = \bigcup_{i=1}^{\ell} T_i \) is HVT of \( M \).

**Lemma 3.6.** Assume that the Problem 2.4 of DAE systems is structurally nonsingular, then fixed-point iteration algorithm gives the same smallest dual-optimal pair with block fixed-point iteration algorithm.

**Proof.** Without loss of generality, the \( \Sigma \) matrix \( M \) of Problem 2.4 with block triangular forms, and \( \sum_{i=1}^{\ell} n_i = n \). Set \( \mathbf{p} = (p_1, p_2, ..., p_\ell) \) with \( \ell \) sections is the parameter vector; \( \mathbf{c}^o = (c_1^o, c_2^o, ..., c_\ell^o) \) and \( \mathbf{d}^o = (d_1^o, d_2^o, ..., d_\ell^o) \) are the smallest dual-optimal found by block fixed-point iteration algorithm; \( \mathbf{c}^* = (c_1^*, c_2^*, ..., c_\ell^*) \) and \( \mathbf{d}^* = (d_1^*, d_2^*, ..., d_\ell^*) \) are the smallest dual-optimal found by fixed-point iteration algorithm.

For \( \ell \) is integer, we prove the lemma by mathematical induction. Considering about \( \ell = 1 \), it is easy to know \( p_1 = 0 \), so then \( c_1^o = c_1^* \) and \( d_1^o = d_1^* \), that is, the lemma is true. Assume the lemma is true when \( \ell = N - 1 \), that is,

\[
c_k^o = c_k^*, \quad d_k^o = d_k^*, \quad (3.3)
\]

for \( k = 1, 2, ..., N - 1 \). We now consider about \( \ell = N \). By (3.3), obtain

\[
\mathbf{p}_N = \text{ColumnMax}(\text{RowAdd}(\begin{pmatrix} M_{1,N} \\ \vdots \\ M_{N-1,N} \end{pmatrix}, \begin{pmatrix} c_1^o \\ \vdots \\ c_{N-1}^o \end{pmatrix}))
\]
Proof. We can easily prove the theorem by Theorem 2.7 and Lemma 3.2 and 3.6. 

\[ \text{ColumnMax}(\text{RowAdd}\left( \begin{pmatrix} M_{1,N} & \vdots & c_1^* \\ \vdots & \ddots & \vdots \\ M_{N-1,N} & \vdots & c_{N-1}^* \end{pmatrix} \right)) \]

it is easily verified that \( c_N^* \) and \( d_N^* \) are the dual-optimal of Problem 3.2 with parameter \( p_N \). By Lemma 3.2 we obtain

\[ c_N^* < c_N^*, \quad d_N^* < d_N^*. \quad (3.4) \]

On the other hand, construct \( c^{**} = (c_1^*, \ldots, c_{N-1}^*, c_N^*) \) and \( d^{**} = (d_1^*, \ldots, d_{N-1}^*, d_N^*) \). From 3.3 and 3.4, \( c^{**} \) and \( d^{**} \) are the dual feasible solution of Problem 2.4. By Lemma 2.1(iii), obtain

\[ d_{k,\sigma}^* - c_{k,\sigma}^* = \sigma_{k,\sigma}, \quad (3.5) \]

for each \((i_k, j_k) \in T_k, k = 1, 2, \ldots, N-1, \) and

\[ d_{N,N}^* - c_{N,N}^* = \sigma_{i_N,j_N} \quad (3.6) \]

for each \((i_N, j_N) \in T_N. \) From 3.3 and 3.4, Lemma 3.6 we have

\[ d_j^* - c_i^* = \sigma_{i,j} \quad (3.7) \]

for each \((i, j) \in T = \bigcup_{i=1}^N T_i. \) That is, \( c^{**} \) and \( d^{**} \) are dual-optimal of Problem 2.4 with \( \Sigma = M \) by Lemma 2.1(iii). Note that \( c^* \) and \( d^* \) are the smallest dual-optimal of Problem 2.4. By Lemma 2.6 obtain

\[ c^* < c^{**}, d^* < d^{**}. \quad (3.8) \]

Moreover, by 3.8, we get

\[ c_N^* < c_N^*, d_N^* < d_N^*. \quad (3.9) \]

Combining 3.4 with 3.9, obtain

\[ c_N^* = c_N^*, d_N^* = d_N^*. \quad (3.10) \]

So we have \( c^* = c^o \) and \( d^* = d^o. \) It is shown that the lemma is true. \( \square \)

Now, we are able to obtain the main theorem as follows.

**Theorem 3.7.** If the \( \Sigma \) matrix for given DAE systems with BTF \( \Sigma \) is structurally nonsingular, then the unique smallest dual-optimal pair \( c^* = (c_1^*, c_2^*, \ldots, c_{\ell}^*) \) and \( d^* \) of the DAE can be found in time \( O(\sum_{i=1}^\ell n_i^3 + ||c_i||_1 \cdot n_i^2) \) by block fast index reduction algorithm. Furthermore, if \( n_i = r \) for each \( i \), i.e., \( n = \ell \cdot r \), then the time is \( O(\ell \cdot r^3 + ||c^*||_1 \cdot r^2) \).

**Proof.** We can easily prove the theorem by Theorem 2.7 and Lemma 3.2 and 3.6 \( \square \)

**Example 3.8.** Consider the application of the Algorithm 3 to nonlinear DAE system \( f = (f_1, f_2, \ldots, f_6) = 0 \) in six dependent variables \( x_1(t), x_2(t), \ldots, x_6(t) \) with known forcing functions \( u_i(t)(i = 1, 2, \ldots, 6): \)

\[
\begin{align*}
    f_1 &= \dot{x}_1 + x_3 + u_1(t) \\
    f_2 &= \dot{x}_2 + x_3 + u_2(t) \\
    f_3 &= \dot{x}_3 + x_2^2 + \dot{x}_6 + u_3(t) \\
    f_4 &= \dot{x}_4 + x_6 + u_4(t) \\
    f_5 &= \dot{x}_5 + x_6 + u_5(t) \\
    f_6 &= x_4^2 + x_5^2 + u_6(t)
\end{align*}
\]
The corresponding signature matrix is

\[
\Sigma = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 2 \\
1 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

where we have already marked the \( HVT \) with asterisks; \( p = (0, 0, 0, 0, 0, 2) \) is the corresponding parameter vector; \( \tilde{c}^*_i = (0, 0, 1) \) and \( \tilde{d}^*_i = (2, 1, 0) \) are the local smallest offsets for each diagonal signature matrix \( \Sigma_{ii} \), \( i = 1, 2 \) using fixed-point iteration algorithm; \( c^* = (0, 0, 1, 2, 3) \) and \( d^* = (2, 1, 0, 3, 3, 2) \) are the global smallest offsets for \( \Sigma \) matrix directly using fixed-point iteration algorithm.

In the following, the main process of block fixed-point iteration algorithm is shown. The signature matrix \( \Sigma \) above contains two blocks. For \( i = 1 \), \( p_1 = (0, 0, 0) \) is the parameter for the first diagonal block, get \( c^*_1 = \tilde{c}^*_1 = (0, 0, 1) \) and \( d^*_1 = \tilde{d}^*_1 = (2, 1, 0) \) from Example 3.8 and remark 3.8. For \( i = 2 \), get \( p_2 = (0, 0, 2) \) which is the parameter for the second diagonal block, and obtain \( c^*_2 = (1, 2, 3) \), \( d^*_2 = (3, 3, 2) \) from Example 3.8. So \( c^* = (c^*_1, c^*_2) = (0, 0, 1, 2, 3) \) and \( d^* = (d^*_1, d^*_2) = (2, 1, 0, 3, 3, 2) \) are the smallest offsets for the DAE system.

**Remark 3.9.** From the Example 3.8, the local smallest offset for the second diagonal signature matrix \( \Sigma_{22} \) are \( \tilde{c}^*_2 = (0, 0, 1) \) and \( \tilde{d}^*_2 = (2, 1, 0) \); while the corresponding global smallest offsets are \( c^*_2 = (1, 2, 3) \) and \( d^*_2 = (3, 3, 2) \). So there does not exist some constant between \( \tilde{c}^*_i \) and \( c^*_i \) or \( \tilde{d}^*_i \) and \( d^*_i \), which is in conflict with Theorem 4.4 in [13]. It is indicated that our block fixed-point iteration method can find the ‘smallest’ optimal offsets for DAE with BTF.

**4 Conclusions**

In this paper, we reinforce the theoretical foundation for Pryce’s structural index reduction method of DAE systems, prove the existence and uniqueness of the smallest offsets, and then show the polynomial complexity for finding optimal index reduction of the \( \Sigma \) matrix for given DAEs.

To solve large scale DAE systems with block structure, we describe a block fixed-point iteration method which can be applied to a sequence of sub-systems rather than the whole system. Accordingly, the time complexity of our method decreases proportionally with the number of the diagonal blocks in the signature matrix. Especially, as explained in Example 3.8 Pryce’s block index reduction method in [13] may fail to find the smallest offsets which can be obtained by our block fixed-point iteration method.

As pointed in the Campbell-Griepentrog Robot Arm [3] and the special DAE with parameter [6], Pryce’s structural analysis method fails to find a DAE’s true structure because of producing an identically singular Jacobian. We believe these situations have appeared rarely in the practical applications. Compared with other structural index reduction methods, our method can address a fairly wide class of large-scale DAE systems precisely and efficiently. And the actual performance of block fixed-point iteration algorithm will discuss in future work.

**Acknowledgements** This work was partially supported by China 973 Project (Grant No. NKBRPC-2011CB302402), National Natural Science Foundation of China (Grant Nos. 11001040, 91118001, 11171053), the West Light
References

1. Brenan K E, Campbell S L, Petzold L R. Numerical Solution of Initial-Value Problems in Differential Algebraic Equations. SIAM Publications, Philadelphia, PA, 2nd edition, 1996.
2. Campbell S L, Geier C W. The index of general nonlinear DAEs. Numerische Mathematik, 1995, 72: 173-196.
3. Campell S L, Griepentrog E. Solvability of general differential algebraic equations. SIAM J Sci Comput, 1995, 16(2): 257–270.
4. Duff I S, Uçar B. On the block triangular form of symmetric matrices. SIAM Review, 2010, 53(3): 455-470.
5. Gear C W, Petzold L R. ODE methods for the solution of differential/algebraic systems. SIAM J Numer Anal, 1983, 21(4): 716–728.
6. Lamour R, März R. Detecting structures in differential algebraic equations: Computational aspects. Journal of Computational and Applied Mathematics, 2012, 236: 4055-4066.
7. Lamour R, Tischendorf C, März R. Differential-Algebraic Equations: A Projector Based Analysis. Springer-Verlag Berlin Heidelberg, 2013.
8. Maffezioli C, Girelli R, Lluka P. Generating efficient computational procedures from declarative models. Simulation Practice and Theory, 1996, 4:303–317.
9. Modelica Association. Modelica-A unified object-oriented language for systems modeling: language specification. https://www.modelica.org/documents/ModelicaSpec33.pdf, 2012.
10. Nedialkov N S, Pryce J D, Tan G N. DAESA-A matlab tool for structural analysis of DAEs: Software. http://www.cas.mcmaster.ca/~nedialk/PAPERS/DAEs/daesa_software/daesaSoftware.pdf, 2012.
11. Pantelides C C. The consistent initialization of differential-algebraic systems. SIAM J Sci Stat Comput, 1988, 9(2): 213–231.
12. Petzold L R. Differential/algebraic equations are not ODEs. SIAM J Sci Stat Comp, 1982, 3: 367–384.
13. Pothen A, Fan C. Computing the block triangular form of a sparse matrix. ACM Transactions on Math Softw, 1990, 16(4): 303–324.
14. Pryce J D. A simple structural analysis method for DAEs. BIT, 2001, 41(2): 364–394.
15. Pryce J D, Nedialkov N S, Tan G N. DAESA-A matlab tool for structural analysis of DAEs: Theory. http://www.cas.mcmaster.ca/cas/0reports/CAS-12-01-NN.pdf, 2012.
16. Qin X L, Wu W Y, Feng Y, Reid G. Structural analysis of high index DAE for process simulation. Int J Model Simul Sci Comput, 2013, 4(4): 1–16. DOI:10.1142/S1793962313420087.
17. Rheinboldt W C. Differential-algebraic systems as differential equations on manifolds. Math Comp, 1984, 43:473–482.
18. Schrijver A. Combinatorial Optimization: Polyhedra and Efficiency. Springer-Verlag Berlin Heidelberg, 2004.
19. Wu W Y, Reid G, Ilie S. Implicit Riquier Bases for PDAE and their Semi-Discretizations. J Symb Comput 01/2009; 44:923-941. DOI:10.1016/j.jsc.2008.04.020.