SIAM J. MATRIX ANAL. APPL.
Vol. 39, No. 3, pp. 1135–1151

INDEX REDUCTION VIA UNIMODULAR TRANSFORMATIONS*

SATORU IWATA† AND MIZUYO TAKAMATSU‡

Abstract. This paper presents an algorithm for transforming a matrix pencil $A(s)$ into another matrix pencil $U(s)A(s)$ with a unimodular matrix $U(s)$ so that the resulting Kronecker index is at most one. The algorithm is based on the framework of combinatorial relaxation, which combines graph-algorithmic techniques and matrix computation. Our algorithm works for index reduction of linear constant coefficient differential-algebraic equations, including those for which the existing index reduction methods based on Pantelides' algorithm or the signature method are known to fail.

Key words. matrix pencil, index reduction, bipartite matching, combinatorial relaxation, differential-algebraic equations

AMS subject classifications. 15A22, 34A09, 65L80, 05C50

DOI. 10.1137/17M111794X

1. Introduction.

A matrix pencil is a polynomial matrix in which the degree of each entry is at most one. Each matrix pencil $A(s)$ can be brought into its Kronecker canonical form (KCF) by a strict equivalence transformation, i.e., a transformation $PA(s)Q$ with constant nonsingular matrices P and Q. Numerically stable computation of KCF is a challenging problem, which has required enormous efforts [2, 4, 5, 10, 26].

A matrix pencil $A(s)$ is called regular if it is square and $\det A(s)$ is a nonvanishing polynomial. The KCF of a regular matrix pencil is in a block-diagonal form that consists of $N_{\mu_1}, \ldots, N_{\mu_d}$ with $N_\mu = I_\mu + sJ_\mu$ and the residual square block $W_{\mu_0} + sI_{\mu_0}$, where I_μ is the $\mu \times \mu$ identity matrix, J_μ is a $\mu \times \mu$ matrix in which entries of the first superdiagonal are 1 and all the remaining entries are zero, and W_μ is a $\mu \times \mu$ constant matrix. A regular matrix pencil $A(s)$ appears in linear constant coefficient differential-algebraic equations (DAEs), and $\nu(A) := \max_{1 \le i \le d} \mu_i$ is related to numerical difficulty for solving the corresponding DAE. The integer $\nu(A)$ is referred to as the index of nilpotency [7] or the Kronecker index [13]. We use the latter name in this paper.

Let $A(s)$ be an $n \times n$ regular matrix pencil. Previous work given in [8, 9, 17, 23] aims at finding the Kronecker index $\nu(A)$ without obtaining the KCF. They utilize the following characterization:

$$\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1. \tag{1.1}$$

Here, $\delta_k(A)$ denotes the maximum degree of minors of order k in $A(s)$, i.e.,

$$\delta_k(A) = \max\{\deg \det A(s)[I, J] \mid |I| = |J| = k\}, \tag{1.2}$$

where $\deg a(s)$ designates the degree of a polynomial $a(s)$ and $A(s)[I, J]$ denotes the submatrix with row set I and column set J.

*Received by the editors February 22, 2017; accepted for publication (in revised form) by T. Stykel May 3, 2018; published electronically July 3, 2018.
http://www.siam.org/journals/simax/39-3/M111794.html
Funding: This research was supported in part by JST CREST, grant JPMJCR14D2, Japan. The research of the second author was supported in part by JSPS KAKENHI grant 25730009.

†Department of Mathematical Informatics, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan (iwata@mist.i.u-tokyo.ac.jp).
‡Department of Information and System Engineering, Chuo University, Kasuga 1-13-27, Bunkyo-ku, Tokyo 112-8551, Japan (takamatsu@ise.chuo-u.ac.jp).

An easy way to estimate $\delta_k(A)$ is to make use of an upper bound $\hat{\delta}_k(A)$ obtained by solving a matching problem in a bipartite graph. The value $\hat{\delta}_k(A)$ corresponds to the maximum degree of a nonzero term in the determinant expansion and is equal to $\delta_k(A)$ unless there is unlucky numerical cancellation. The previous work [8, 9, 17, 23] presents algorithms to compute $\delta_k(A)$ by exploiting an upper bound $\hat{\delta}_k(A)$.

This paper focuses on the index reduction of a matrix pencil, while the above previous work deals with the index computation. Our aim is to transform $A(s)$ into another matrix pencil with the Kronecker index at most one. More precisely, we present an algorithm for finding a unimodular polynomial matrix $U(s)$ such that $U(s)A(s)$ is a matrix pencil with $\nu(UA) \leq 1$.

Once the KCF of $A(s)$ is obtained together with the transformation matrices, it is straightforward to construct such a unimodular matrix $U(s)$. Since numerical difficulty is inherent in the computation of KCF, we aim at finding $U(s)$ more directly without relying on the KCF. Instead of computing the KCF, our algorithm makes use of (1.1). It is known that the value of $\delta_n(A)$ is invariant under unimodular equivalence transformations, which indicates $\delta_n(UA) = \delta_n(A)$. On the other hand, $\delta_{n-1}(UA) = \delta_{n-1}(A)$ does not hold in general. In order to achieve $\nu(UA) \leq 1$, $U(s)$ needs to satisfy $\delta_{n-1}(UA) \leq \delta_n(UA) = \delta_n(A)$. We find such $U(s)$ efficiently by exploiting $\hat{\delta}_n(A)$ and $\hat{\delta}_{n-1}(A)$, which are upper bounds on $\delta_n(A)$ and $\delta_{n-1}(A)$, respectively.

Our motivation comes from the study of DAEs [1, 3, 7, 12, 22]. Consider a linear constant coefficient DAE

$$F\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} + H\boldsymbol{z}(t) = \boldsymbol{g}(t) \tag{1.3}$$

with an initial condition $\boldsymbol{z}(0) = \boldsymbol{z}_0$, where F and H are constant matrices. By the Laplace transformation, we obtain

$$A(s)\tilde{\boldsymbol{z}}(s) = \tilde{\boldsymbol{g}}(s) + F\boldsymbol{z}_0 \tag{1.4}$$

with the matrix pencil $A(s) = sF + H$.

The numerical difficulty of the DAE (1.3) is measured by the Kronecker index $\nu(A)$. A common approach for solving a high index DAE is to transform it into an equivalent DAE with index at most one, which can be solved easily by numerical methods including the backward differentiation formulas (BDF).

An equivalent DAE can be obtained by differentiating a certain equation and adding it to another equation. Such operations correspond to equivalence row transformations with unimodular polynomial matrix $U(s)$. The Laplace transform of the resulting DAE is in the form of

$$U(s)A(s)\tilde{\boldsymbol{z}}(s) = U(s)(\tilde{\boldsymbol{g}}(s) + F\boldsymbol{z}_0). \tag{1.5}$$

If $U(s)A(s)$ is a matrix pencil with $\nu(UA) \leq 1$, we succeed in transforming (1.4) into (1.5) with index at most one.

Example 1.1. Consider a linear constant coefficient DAE

$$-\dot{z}_1 + \dot{z}_2 + z_3 = g_1(t), \tag{1.6}$$

$$z_1 + \dot{z}_3 = g_2(t), \tag{1.7}$$

$$z_2 + \dot{z}_3 = g_3(t). \tag{1.8}$$

By the Laplace transformation, we obtain (1.4) with

$$A(s) = \begin{pmatrix} -s & s & 1 \\ 1 & 0 & s \\ 0 & 1 & s \end{pmatrix}.$$

Since $\delta_3(A) = \deg\det A(s) = 0$ and $\delta_2(A) = \deg\det \left(\begin{smallmatrix} s & 1 \\ 0 & s \end{smallmatrix}\right) = 2$, we have $\nu(A) = 3$ by (1.1).

With a unimodular polynomial matrix $U(s)$ defined by

$$U(s) = \begin{pmatrix} 1 & s & -s \\ -s & -s^2+1 & s^2 \\ 0 & -1 & 1 \end{pmatrix},$$

the matrix pencil $A(s)$ is transformed into $U(s)A(s) = \left(\begin{smallmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \end{smallmatrix}\right)$ with $\nu(UA) \le 1$. The unimodular polynomial matrix $U(s)$ corresponds to the following transformation for the DAE (1.6)–(1.8):

$$\begin{array}{ll} (1.6) + (1.7)' - (1.8)' & z_3 = g_1(t) + g_2'(t) - g_3'(t), \\ -(1.6)' - (1.7)'' + (1.7) + (1.8)'' & z_1 = -g_1'(t) - g_2''(t) + g_2(t) + g_3''(t), \\ -(1.7) + (1.8) & -z_1 + z_2 = -g_2(t) + g_3(t). \end{array}$$

The coefficient matrix pencil for the resulting DAE coincides with $U(s)A(s)$.

The modeling and simulation software for dynamical systems, such as Dymola, OpenModelica, and MapleSim, is equipped with the index reduction methods based on Pantelides' algorithm [20], the dummy derivative approach [14], or the signature method [21]. These algorithms adopt a structural approach, which extracts a zero/nonzero pattern of coefficients in equations, ignoring the numerical values. Such algorithms are efficient, because they exploit graph-algorithmic techniques. However, the discard of numerical information can cause a failure even for linear constant coefficient DAEs. For nonlinear DAEs, Tan, Nedialkov, and Pryce [24] proposed two symbolic-numeric conversion methods for fixing the signature method without proving its termination. We focus on index reduction of linear constant coefficient DAEs and our algorithm is proved to work for any instances if the numerical computation is carried out exactly.

The algorithms for computing $\delta_k(A)$ given in [8, 9, 17, 23] are based on the framework of "combinatorial relaxation," which combines graph-algorithmic techniques and matrix computation. The combinatorial relaxation approach was invented by Murota [16] for computing the Newton diagram of Puiseux-series solutions to determinantal equations and then applied to the computation of the degree of determinants of polynomial matrices [18]. In combinatorial relaxation algorithms for computing $\delta_k(A)$, we find an upper bound $\hat{\delta}_k(A)$ of $\delta_k(A)$ by solving a matching problem and check if $\hat{\delta}_k(A) = \delta_k(A)$ by constant matrix computation. If $\hat{\delta}_k(A) \neq \delta_k(A)$, then we modify $A(s)$ to improve $\hat{\delta}_k(A)$ without changing $\delta_k(A)$. After a finite number of iterations, the algorithms terminate with $\hat{\delta}_k(A) = \delta_k(A)$. They mainly rely on fast combinatorial algorithms and perform numerical computation only when necessary.

Our index reduction algorithm, which consists of two phases, inherits the idea of combinatorial relaxation. In the first phase, we transform $A(s)$ into another matrix pencil $\tilde{A}(s)$ such that an estimate of $\nu(\tilde{A})$ is at most one. In the second phase, we determine if the estimate is correct. If not, we further transform $\tilde{A}(s)$ into another matrix pencil $\hat{A}(s)$ with $\nu(\hat{A}) \le 1$. In both phases, we exploit a feasible dual solution

of the matching problem, which was also used by Pryce [21] in the interpretation of Pantelides' algorithm [20]. Moreover, we exploit the *tight coefficient matrix* in the second phase. The tight coefficient matrix has been commonly used in combinatorial relaxation algorithms. It is called the *system Jacobian matrix* [21] in the context of DAEs. We remark that the second phase is similar to the linear combination method given by Tan, Nedialkov, and Pryce [24]. By combining the two phases, our algorithm succeeds in obtaining the DAE with index at most one.

The rest of this paper is organized as follows. In section 2, we explain the bipartite matching problems associated with matrix pencils. We present an index reduction algorithm in section 3. Section 4 gives numerical examples, and section 5 concludes this paper.

2. Matrix pencils and matching problems.

2.1. Preliminaries. For a polynomial $a(s)$, we denote the degree of $a(s)$ by $\deg a$, where $\deg 0 = -\infty$ by convention. A polynomial matrix $A(s) = (a_{ij}(s))$ with $\deg a_{ij} \leq 1$ for all (i, j) is called a *matrix pencil*. A matrix pencil $A(s)$ is said to be *regular* if $A(s)$ is square and $\det A(s)$ is a nonvanishing polynomial.

Let us denote by block-diag$(D_1, \ldots, D_b)$ the block-diagonal matrix pencil with diagonal blocks $D_1, \ldots, D_b$. A regular matrix pencil is known to be strictly equivalent to a block-diagonal form, the KCF [6, Chapter XII], in the form of block-diag$(sI_{\mu_0} + W_{\mu_0}, N_{\mu_1}, \ldots, N_{\mu_d})$, where I_μ is the $\mu \times \mu$ identity matrix, W_μ is a $\mu \times \mu$ constant matrix, and N_μ is a $\mu \times \mu$ matrix pencil defined by

$$N_\mu = \begin{pmatrix} 1 & s & 0 & \cdots & 0 \\ 0 & 1 & s & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & s \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

Remember that for a regular matrix pencil $A(s)$, the Kronecker index $\nu(A)$ is defined to be the maximum size of N_μ blocks in the KCF of $A(s)$, i.e., $\max_{1 \leq i \leq d} \mu_i$. It is known [19, Theorem 5.1.8] that $\nu(A)$ is expressed by (1.1).

A polynomial matrix is called *unimodular* if it is square and its determinant is a nonvanishing constant. This implies that a square polynomial matrix is unimodular if and only if its inverse is a polynomial matrix.

2.2. Combinatorial estimate of $\boldsymbol{\delta_n(A)}$. Let $A(s) = (A_{ij}(s))$ be an $n \times n$ regular matrix pencil with row set R and column set C. We construct a bipartite graph $G(A) = (R, C; E(A))$ that has the vertex bipartition (R, C) corresponding to the row set R and the column set C of $A(s)$. The edge set $E(A)$ is defined by $E(A) = \{(i, j) \mid i \in R, j \in C, A_{ij}(s) \neq 0\}$. The weight σ_{ij} of an edge (i, j) is given by $\sigma_{ij} = \deg A_{ij}(s)$. We remark that σ_{ij} is equal to the (i, j) entry of the *signature matrix* in the signature method [21]. Since $A(s)$ is a matrix pencil, σ_{ij} is 0 or 1 for each $(i, j) \in E(A)$. A subset M of $E(A)$ is called a *matching* if every pair of edges in M is disjoint. A matching M is called a *perfect matching* if M covers all the vertices.

Consider the following maximum-weight perfect matching problem P(A):

$$\begin{aligned} &\text{Maximize} && \sum_{(i,j) \in M} \sigma_{ij} \\ &\text{subject to} && M \text{ is a perfect matching.} \end{aligned}$$

Since $A(s)$ is regular, $G(A)$ has a perfect matching. Let $\hat{\delta}_n(A)$ denote the maximum weight of a perfect matching in $G(A)$. Then $\hat{\delta}_n(A)$ is an upper bound on $\delta_n(A)$, i.e.,

$$\delta_n(A) \le \hat{\delta}_n(A). \tag{2.1}$$

From various possible formulations of the dual problem of P(A), we choose the following problem D(A):

$$\text{Minimize} \quad \Delta_n(p,q) := \sum_{i\in R} p_i - \sum_{j\in C} q_j \tag{2.2}$$

$$\text{subject to} \quad p_i - q_j \ge \sigma_{ij} \quad ((i,j)\in E(A)), \tag{2.3}$$

$$p_i \in \mathbb{Z} \quad (i \in R), \tag{2.4}$$

$$q_j \in \mathbb{Z} \quad (j \in C). \tag{2.5}$$

Our index reduction algorithm updates a matrix pencil $A(s)$ and a feasible solution (p,q) of D(A), which is not necessarily optimal. Let (p,q) be a feasible solution of D(A). The weak duality for the maximum-weight perfect matching problem says

$$\hat{\delta}_n(A) = \sum_{(i,j)\in M} \sigma_{ij} \le \sum_{(i,j)\in M} (p_i - q_j) = \sum_{i\in R} p_i - \sum_{j\in C} q_j = \Delta_n(p,q), \tag{2.6}$$

where M denotes a maximum-weight perfect matching. The inequality is due to (2.3). It follows from (2.1) and (2.6) that

$$\delta_n(A) \le \hat{\delta}_n(A) \le \Delta_n(p,q). \tag{2.7}$$

Thus, we can make use of $\Delta_n(p,q)$ as a combinatorial estimate of $\delta_n(A)$.

In order to bound the time complexity of our algorithm, we need an optimal solution of D(A) for the initial matrix pencil $A(s)$ satisfying

$$\min_{i\in R} p_i \ge 0, \quad \min_{j\in C} q_j = 0, \quad \max_{j\in C} q_j \le n. \tag{2.8}$$

We explain a way to obtain such an optimal solution (p,q). It should be noted that construction of (p,q) described below is performed only once at the beginning of the algorithm.

Let M be a maximum-weight perfect matching in $G(A) = (R, C; E(A))$. Consider an auxiliary directed graph $\check{G}_M = (\check{V}, \check{E})$ with $\check{V} = R \cup C \cup \{w\}$ and

$$\check{E} = M \cup \bar{E} \cup \{(w,j) \mid j \in R\},$$

where w is a new vertex and $\bar{E} = \{(i,j) \mid (j,i) \in E(A)\}$. We define the arc length $\gamma : \check{E} \to \mathbb{Z}$ by

$$\gamma(i,j) = \begin{cases} \sigma_{ij} & ((i,j)\in M), \\ -\sigma_{ji} & ((i,j)\in \bar{E}), \\ 0 & (i = w, j \in R). \end{cases}$$

Example 2.1. Consider a 3×3 matrix pencil

$$A(s) = \begin{pmatrix} 1 & s & s \\ 0 & s & s \\ 0 & 0 & 1 \end{pmatrix}$$

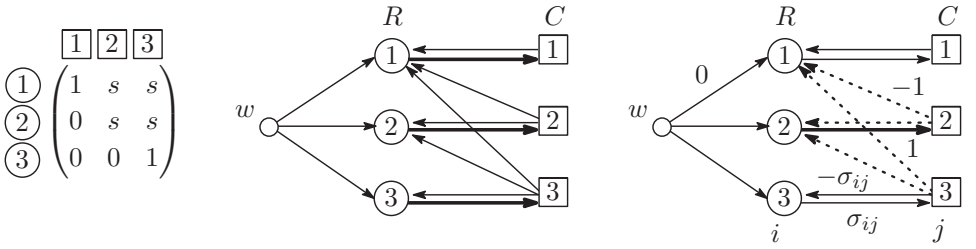

FIG. 1. *A matrix pencil $A(s)$ and two copies of auxiliary directed graphs $\check{G}_M = (\check{V}, \check{E})$. In the middle graph, heavy lines show arcs in M. In the right graph, heavy, solid, and dotted lines represent arcs of weight* 1, 0, −1, *respectively.*

with $R = \{1, 2, 3\}$ and $C = \{1, 2, 3\}$. The bipartite graph $G(A)$ has a maximum-weight perfect matching $M = \{(1, 1), (2, 2), (3, 3)\}$. Figure 1 depicts $A(s)$ and auxiliary directed graphs $\check{G}_M = (\check{V}, \check{E})$.

Let $\rho(i, j)$ be the shortest path distance from $i \in \check{V}$ to $j \in \check{V}$ with respect to the arc length γ in $\check{G}_M$, which might be negative. We denote $\max_{\ell \in C} \rho(w, \ell)$ by $\rho_{\max}$.

LEMMA 2.2. *The pair (p, q) given by*

$$p_i = -\rho(w, i) + \rho_{\max} \quad (i \in R), \tag{2.9}$$

$$q_j = -\rho(w, j) + \rho_{\max} \quad (j \in C) \tag{2.10}$$

is an optimal solution of D(A) *satisfying* (2.8).

Proof. By the definition of (p, q), (2.4) and (2.5) clearly hold. For $(i, j) \in E(A)$, we have $(j, i) \in \bar{E}$ in $\check{G}_M$ and it holds that

$$\rho(w, i) \leq \rho(w, j) + \gamma(j, i) = \rho(w, j) - \sigma_{ij}, \tag{2.11}$$

because the shortest path from w to i is shorter than or equal to a path through j. It follows from (2.9)–(2.11) that

$$p_i - q_j = -\rho(w, i) + \rho(w, j) \geq \sigma_{ij} \quad ((i, j) \in E(A)). \tag{2.12}$$

Thus (2.3) holds. This implies that (p, q) is a feasible solution of D(A).

For $(i, j) \in M$, we have

$$\rho(w, j) \leq \rho(w, i) + \gamma(i, j) = \rho(w, i) + \sigma_{ij},$$

because the shortest path from w to j is shorter than or equal to a path through i. Thus, (2.11) holds with equality, which implies that

$$p_i - q_j = -\rho(w, i) + \rho(w, j) = \sigma_{ij} \quad ((i, j) \in M). \tag{2.13}$$

It follows from (2.9), (2.10), and (2.13) that

$$\sum_{i \in R} p_i - \sum_{j \in C} q_j = -\sum_{i \in R} \rho(w, i) + \sum_{j \in C} \rho(w, j) = \sum_{(i,j) \in M} (-\rho(w, i) + \rho(w, j)) = \sum_{(i,j) \in M} \sigma_{ij},$$

where the second equality holds because M is a perfect matching. Hence (p, q) is optimal to $\mathrm{D}(A)$.

Finally, we show that (p, q) satisfies (2.8). The second condition follows from the definition of q_j. Since $G(A)$ has a perfect matching, each $i \in R$ is adjacent to at least one vertex $j \in C$. Hence we have $p_i \geq q_j + \sigma_{ij} \geq 0$ by (2.12), $q_j \geq 0$, and $\sigma_{ij} \geq 0$. This implies the first condition $\min_{i \in R} p_i \geq 0$.

We now show the last condition. Let P_{ij} denote the shortest path from $i \in \check{V}$ to $j \in \check{V}$. Consider two paths P_{wj} and $P_{w\ell}$. For the last common vertex v in P_{wj} and $P_{w\ell}$, it holds that $\rho(w, \ell) - \rho(w, j) = \rho(v, \ell) - \rho(v, j)$. Note that $\rho(v, \ell)$ is at most the number of arcs in M on $P_{v\ell}$, whereas $-\rho(v, j)$ is at most the number of arcs in $\bar{E}$ on P_{vj}. The sum of these upper bounds is at most n, because P_{vj} and $P_{v\ell}$ have no common vertex except v. Thus we obtain $q_j \leq n$ for every $j \in C$. ∎

2.3. Combinatorial estimate of $\delta_{n-1}(A)$. Let (p, q) be a feasible solution of $\mathrm{D}(A)$. We now introduce a combinatorial estimate of $\delta_{n-1}(A)$ with (p, q). Consider the following matching problem:

$$\begin{aligned} \text{Maximize} \quad & \sum_{(i,j) \in M} \sigma_{ij} \\ \text{subject to} \quad & M \text{ is a matching}, \\ & |M| = n - 1. \end{aligned}$$

The optimal value is denoted by $\hat{\delta}_{n-1}(A)$.

For a submatrix $A(s)[I, J]$ with $|I| = |J| = n - 1$, the defining expansion of the determinant is given by

$$\det A(s)[I, J] = \sum_{\sigma : I \to J} \operatorname{sgn} \sigma \prod_{i \in I} A_{i\sigma(i)}(s),$$

where σ runs over all one-to-one correspondence from I to J and $\operatorname{sgn} \sigma = \pm 1$ is the sign of σ. Then we have

$$\max_{|I| = |J| = n-1} \max_{\sigma : I \to J} \deg \prod_{i \in I} A_{i\sigma(i)}(s) = \hat{\delta}_{n-1}(A),$$

because a nonzero term $\prod_{i \in I} A_{i\sigma(i)}(s)$ corresponds to a matching M with $|M| = n-1$. Combining this with (1.2), we obtain

$$\delta_{n-1}(A) \leq \hat{\delta}_{n-1}(A), \tag{2.14}$$

which says that $\hat{\delta}_{n-1}(A)$ is an upper bound on $\delta_{n-1}(A)$.

For a feasible solution (p, q) of $\mathrm{D}(A)$, we define $\Delta_{n-1}(p, q)$ by

$$\Delta_{n-1}(p, q) := \Delta_n(p, q) - \min_{i \in R} p_i + \max_{j \in C} q_j = \sum_{i \in R} p_i - \min_{i \in R} p_i - \sum_{j \in C} q_j + \max_{j \in C} q_j.$$

LEMMA 2.3. *For a feasible solution (p, q) of $\mathrm{D}(A)$, we have*

$$\hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p, q).$$

Proof. Let M be a maximum-weight matching of size $n-1$ and ∂M denote the set of vertices incident to M. Then we have

$$\hat{\delta}_{n-1}(A) = \sum_{(i,j)\in M} \sigma_{ij} \leq \sum_{(i,j)\in M} (p_i - q_j) = \sum_{i\in R\cap\partial M} p_i - \sum_{j\in C\cap\partial M} q_j$$
$$\leq \sum_{i\in R} p_i - \min_{i\in R} p_i - \sum_{j\in C} q_j + \max_{j\in C} q_j = \Delta_{n-1}(p,q),$$

where the first inequality is due to (2.3) and the second inequality follows from $|R\cap\partial M| = |C\cap\partial M| = n-1$. ∎

It follows from (2.14) and Lemma 2.3 that

$$\delta_{n-1}(A) \leq \hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p,q). \tag{2.15}$$

Thus, we can adopt $\Delta_{n-1}(p,q)$ as a combinatorial upper bound on $\delta_{n-1}(A)$.

Example 2.4. Consider $A(s)$ given in Example 2.1 again. It holds that

$$\delta_{n-1}(A) = \deg\det A(s)[\{2,3\},\{2,3\}] = \deg\det \begin{pmatrix} s & s \\ 0 & 1 \end{pmatrix} = \deg s = 1.$$

Since $G(A)$ has a maximum-weight matching of size 2 in $A(s)[\{1,2\},\{2,3\}] = \left(\begin{smallmatrix} s & s \\ s & s \end{smallmatrix}\right)$, we have $\hat{\delta}_{n-1}(A) = 2$. By computing the shortest path distance from w to each vertex, we obtain $p = (1,1,0)$ and $q = (1,0,0)$. Hence, it holds that

$$\Delta_{n-1}(p,q) = \sum_{i\in R} p_i - \min_{i\in R} p_i - \sum_{j\in C} q_j + \max_{j\in C} q_j = 2 - 0 - 1 + 1 = 2.$$

Thus we obtain $1 = \delta_{n-1}(A) \leq \hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p,q) = 2$.

3. Index reduction algorithm.

3.1. Outline of algorithm. Let $A(s)$ be an $n\times n$ regular matrix pencil, and let (p,q) be a feasible solution of $\mathrm{D}(A)$ satisfying (2.8). By (2.7) and (2.15), we have

$$\delta_n(A) \leq \hat{\delta}_n(A) \leq \Delta_n(p,q), \quad \delta_{n-1}(A) \leq \hat{\delta}_{n-1}(A) \leq \Delta_{n-1}(p,q). \tag{3.1}$$

Our aim is to find a unimodular matrix $U(s)$ such that $\bar{A}(s) = U(s)A(s)$ is a matrix pencil with index $\nu(\bar{A}) \leq 1$. The following algorithm updates a matrix pencil $A(s)$ and a feasible solution (p,q). The upper bounds $\Delta_n(p,q)$ and $\Delta_{n-1}(p,q)$ are non-increasing. The resulting matrix pencil $\bar{A}(s)$ and a feasible solution $(\bar{p},\bar{q})$ of $\mathrm{D}(\bar{A})$ satisfy

$$\delta_n(\bar{A}) = \hat{\delta}_n(\bar{A}) = \Delta_n(\bar{p},\bar{q}), \quad \delta_{n-1}(\bar{A}) = \hat{\delta}_{n-1}(\bar{A}) = \Delta_{n-1}(\bar{p},\bar{q}), \tag{3.2}$$
$$\bar{p}_i \in \{0,1\} \quad (i\in R), \qquad \bar{q}_j = 0 \quad (j\in C). \tag{3.3}$$

We now describe the outline of the index reduction algorithm. The algorithm consists of two phases. We design the first phase with the aim of decreasing $\max_{j\in C} q_j$ until zero. This leads to the condition (3.3), as shown in Lemma 3.3.

We adopt

$$\hat{\nu}(p,q) := \Delta_{n-1}(p,q) - \Delta_n(p,q) + 1$$

as an estimate of $\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1$. Then $\hat{\nu}(p,q) \leq 1$ holds at the end of the first phase, which is shown in Corollary 3.4. It should be remarked that this does not imply $\nu(A) \leq 1$, because $\hat{\nu}(p,q)$ is not an upper bound on $\nu(A)$.

After the first phase, we have

$$\Delta_n(p,q) = \sum_{i\in R} p_i, \quad \Delta_{n-1}(p,q) = \sum_{i\in R} p_i - \min_{i\in R} p_i$$

by the condition (3.3). Hence, it follows from (3.1) that

$$\delta_n(A) \leq \Delta_n(p,q) = \sum_{i\in R} p_i, \quad \delta_{n-1}(A) \leq \Delta_{n-1}(p,q) = \sum_{i\in R} p_i - \min_{i\in R} p_i. \tag{3.4}$$

In the second phase, we decrease $\sum_{i\in R} p_i$ until (3.4) holds with equality. We make use of the *tight coefficient matrix* to check if both $\delta_n(A) = \Delta_n(p,q)$ and $\delta_{n-1}(A) = \Delta_{n-1}(p,q)$ hold without computing $\delta_n(A)$ and $\delta_{n-1}(A)$ directly, as shown in Lemma 3.5. If these equalities hold, we obtain

$$\nu(A) = \delta_{n-1}(A) - \delta_n(A) + 1 = \Delta_{n-1}(p,q) - \Delta_n(p,q) + 1 = \hat{\nu}(p,q) \leq 1.$$

If not, we further update $A(s)$ to another matrix pencil.

A formal description of the entire algorithm is as follows.

Outline of index reduction algorithm.

Step 1: Construct an optimal solution (p,q) of $\mathrm{D}(A)$ satisfying (2.8).

Step 2: If $q_j = 0$ for every $j \in C$, then go to Step 4.

Step 3: Bring $A(s)$ into another matrix pencil $\tilde{A}(s)$ by a unimodular transformation with the aid of the partition of R and C according to (p,q), and construct a feasible solution $(\tilde{p},\tilde{q})$ of $\mathrm{D}(\tilde{A})$ from (p,q). Set $A(s) \leftarrow \tilde{A}(s)$ and $(p,q) \leftarrow (\tilde{p},\tilde{q})$. Go back to Step 2.

Step 4: If both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p,q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p,q)$ hold, then terminate.

Step 5: Bring $A(s)$ into another matrix pencil $\hat{A}(s)$ by a unimodular transformation with the aid of the tight coefficient matrix, and construct a feasible solution $(\hat{p},\hat{q})$ of $\mathrm{D}(\hat{A})$ from (p,q). Set $A(s) \leftarrow \hat{A}(s)$ and $(p,q) \leftarrow (\hat{p},\hat{q})$. Go back to Step 4.

The first phase corresponds to Steps 1–3, while the second phase corresponds to Steps 4–5. In Steps 1–3, we decrease p and q in order to construct a feasible solution (p,q) satisfying (3.3), which implies $\hat{\nu}(p,q) \leq 1$. Then we further decrease p to obtain a feasible solution satisfying (3.2) in Steps 4–5. The details of Steps 3–5 are given in sections 3.2–3.4, respectively.

3.2. Unimodular transformations in Step 3. We describe how to construct $(\tilde{p},\tilde{q})$ from a feasible solution (p,q) of $\mathrm{D}(A)$ satisfying (2.8) in Step 3. For each nonnegative integer h, we define $R_h = \{i \in R \mid p_i = h\}$ and $C_h = \{j \in C \mid q_j = h\}$. Then $A(s)$ is expressed as

$$A(s) = \begin{array}{c} \\ R_\eta \\ R_{\eta-1} \\ \vdots \\ \vdots \\ R_1 \\ R_0 \end{array} \begin{array}{c} \begin{array}{cccccc} C_\eta & C_{\eta-1} & C_{\eta-2} & \cdots & C_1 & C_0 \end{array} \\ \begin{pmatrix} * & ** & ** & \cdots & \cdots & ** \\ O & * & ** & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & ** \\ O & \cdots & \cdots & O & * & ** \\ O & \cdots & \cdots & O & O & * \end{pmatrix} \end{array}$$

for some η, where $*$ and $**$ denote a constant matrix and a matrix pencil, respectively. Since $A(s)$ is regular, the submatrix $A(s)[R_0, C_0]$ is of full row-rank, and hence we can express it as $\begin{pmatrix} * & H_0 \end{pmatrix}$ with a nonsingular constant matrix H_0.

Express the submatrix $A(s)[R_1 \cup R_0, C_0]$ as

$$\begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} \begin{array}{c} C_0 \end{array} \\ \begin{pmatrix} ** & sF_1 + H_1 \\ * & H_0 \end{pmatrix} \end{array}$$

with constant matrices F_1 and H_1. By multiplying a unimodular matrix $\left(\begin{smallmatrix} I & -sF_1H_0^{-1} \\ O & I \end{smallmatrix}\right)$ from the left, we obtain

$$\begin{array}{c} \\ R_1 \\ R_0 \end{array} \begin{array}{c} \begin{array}{c} C_0 \end{array} \\ \begin{pmatrix} sF_2 + H_2 & H_1 \\ * & H_0 \end{pmatrix} \end{array}$$

with constant matrices F_2 and H_2. Since $A(s)[R_0, C_1] = O$, this transformation does not change $A(s)[R_1, C_1]$.

Then consider the submatrix $(sF_2 + H_2 \mid H_1)$, which can be transformed into

$$\left(\begin{array}{cc|c} sF_3 + H_3 & ** & * \\ * & * & * \end{array}\right)$$

by row transformations, where F_3 and H_3 are constant matrices with F_3 being nonsingular, so that the lower part does not contain s.

As a result, we obtain another matrix pencil $\tilde{A}(s)$ satisfying the following conditions:

- It holds that

$$\tilde{A}(s)[R_1 \cup R_0, C_1 \cup C_0] = \left(\begin{array}{c|cc|c} * & sF_3 + H_3 & ** & * \\ * & * & * & * \\ \hline O & * & * & H_0 \end{array}\right), \tag{3.5}$$

  where the first two row sets correspond to R_1, the last row set corresponds to R_0, the first column set corresponds to C_1, and the last three column sets correspond to C_0.
- The other entries coincide with the corresponding entries of $A(s)$.

Let us denote the first row set of (3.5) by S. We construct $(\tilde{p}, \tilde{q})$ from (p, q) by

$$\begin{aligned} &\tilde{p}_i = p_i - 1 \quad (i \in R \setminus (R_0 \cup S)), && \tilde{p}_i = p_i \quad (i \in R_0 \cup S), \\ &\tilde{q}_j = q_j - 1 \quad (j \in C \setminus C_0), && \tilde{q}_j = q_j = 0 \quad (j \in C_0). \end{aligned}$$

The following lemma ensures that $(\tilde{p}, \tilde{q})$ is a feasible solution of $\mathrm{D}(\tilde{A})$.

LEMMA 3.1. *Let (p, q) be a feasible solution of* $\mathrm{D}(A)$ *satisfying* (2.8). *Then $(\tilde{p}, \tilde{q})$ is a feasible solution of* $\mathrm{D}(\tilde{A})$ *satisfying* (2.8).

Proof. By the construction rule of $\tilde{A}(s)$, we have $p_i - q_j \geq \tilde{\sigma}_{ij}$. If $\tilde{p}_i - \tilde{q}_j \geq p_i - q_j$ holds, then $\tilde{p}_i - \tilde{q}_j \geq p_i - q_j \geq \tilde{\sigma}_{ij}$ also holds.

Consider the case with $\tilde{p}_i - \tilde{q}_j < p_i - q_j$. This implies that $i \in R \setminus (R_0 \cup S)$ and $j \in C_0$. Then we have $\tilde{p}_i - \tilde{q}_j = p_i - 1$. If $i \notin R_1$ holds, it follows from $p_i \geq 2$ that $\tilde{p}_i - \tilde{q}_j = p_i - 1 \geq 1 \geq \tilde{\sigma}_{ij}$. Next, suppose $i \in R_1 \setminus S$. Then we have $p_i = 1$ and $\tilde{\sigma}_{ij} = 0$ for $(i, j) \in E(\tilde{A})$ by (3.5). Hence $\tilde{p}_i - \tilde{q}_j = p_i - 1 = 0 \geq \tilde{\sigma}_{ij}$ holds. Moreover, $(\tilde{p}, \tilde{q})$ satisfies (2.8) by the construction rule. ∎

The following lemma shows that the values of the right-hand sides in (3.1) decrease or remain the same when we update (p, q) to $(\tilde{p}, \tilde{q})$.

LEMMA 3.2. *Let (p, q) be a feasible solution of* $\mathrm{D}(A)$ *satisfying* (2.8). *The dual solution $(\tilde{p}, \tilde{q})$ obtained by the above procedure satisfies*

$$\Delta_n(p, q) \geq \Delta_n(\tilde{p}, \tilde{q}), \quad \Delta_{n-1}(p, q) \geq \Delta_{n-1}(\tilde{p}, \tilde{q}).$$

Proof. By the definition of $\tilde{p}_i$ and $\tilde{q}_j$, we have

$$\sum_{i \in R} \tilde{p}_i - \sum_{j \in C} \tilde{q}_j = \sum_{i \in R} p_i - |R \setminus (R_0 \cup S)| - \sum_{j \in C} q_j + |C \setminus C_0|.$$

Since F_3 and H_0 in (3.5) are nonsingular, $\tilde{A}(s)[R_0 \cup S, C_0]$ is of full row-rank. Hence we have $|R_0 \cup S| \leq |C_0|$, which implies that

$$|R \setminus (R_0 \cup S)| \geq |C \setminus C_0|. \tag{3.6}$$

Thus the first inequality holds.

By the definition of $\tilde{p}$, the value of $\min_{i \in R} \tilde{p}_i$ is equal to $\min_{i \in R} p_i$ or $\min_{i \in R} p_i - 1$. Since $\sum_{i \in R} \tilde{p}_i = \sum_{i \in R} p_i - |R \setminus (R_0 \cup S)|$ holds, we have

$$\sum_{i \in R} \tilde{p}_i - \min_{i \in R} \tilde{p}_i \leq \sum_{i \in R} \tilde{p}_i - \min_{i \in R} p_i + 1 = \sum_{i \in R} p_i - \min_{i \in R} p_i - |R \setminus (R_0 \cup S)| + 1.$$

Now $C \neq C_0$ holds, because the condition in Step 2 is not fulfilled. Hence

$$\sum_{j \in C} \tilde{q}_j - \max_{j \in C} \tilde{q}_j = \sum_{j \in C} q_j - \max_{j \in C} q_j - (|C \setminus C_0| - 1)$$

follows. Thus we obtain

$$\begin{aligned} \Delta_{n-1}(\tilde{p}, \tilde{q}) &\leq \sum_{i \in R} p_i - \min_{i \in R} p_i - |R \setminus (R_0 \cup S)| - \sum_{j \in C} q_j + \max_{j \in C} q_j + |C \setminus C_0| \\ &\leq \sum_{i \in R} p_i - \min_{i \in R} p_i - \sum_{j \in C} q_j + \max_{j \in C} q_j \\ &= \Delta_{n-1}(p, q), \end{aligned}$$

where the second inequality is due to (3.6). ∎

By executing Steps 1–3, we obtain a matrix pencil $A(s)$ and a feasible solution (p, q) of $\mathrm{D}(A)$ satisfying (3.3) as follows.

LEMMA 3.3. *At the end of Phase* 1, *we obtain (p, q) such that $p_i \in \{0, 1\}$ for every $i \in R$ and $q_j = 0$ for every $j \in C$. Moreover, the number of iterations in Phase* 1 *is at most n.*

Proof. Step 2 ensures that $q_j = 0$ for every $j \in C$. Since $\sigma_{ij} = 0$ or 1, this implies $p_i \in \{0, 1\}$ for each $i \in R$. At each iteration, $\max_{j \in C} q_j$ decreases by one. Lemma 2.2 ensures that $\max_{j \in C} q_j \leq n$ holds for an initial solution (p, q), which indicates that the number of iterations is at most n. ∎

Lemma 3.3 leads to the following corollary.

COROLLARY 3.4. *At the end of Phase* 1*, we have* $\hat{\nu}(p, q) \leq 1$.

Proof. By Lemma 3.3, $p_i \in \{0, 1\}$ holds for every $i \in R$ and $q_j = 0$ holds for every $j \in C$. Let m denote the number of rows with $p_i = 1$. Then we have

$$\Delta_n(p, q) = m, \quad \Delta_{n-1}(p, q) = \begin{cases} m & (m < n), \\ m - 1 & (m = n). \end{cases} \tag{3.7}$$

Hence it holds that

$$\hat{\nu}(p, q) = \Delta_{n-1}(p, q) - \Delta_n(p, q) + 1 = \begin{cases} 1 & (m < n), \\ 0 & (m = n). \end{cases}$$ ∎

3.3. Test for tightness in Step 4. In this section, we present how to check if both $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ hold in Step 4.

Suppose that we have a feasible solution (p, q) of $\mathrm{D}(A)$ such that $p_i \in \{0, 1\}$ for every $i \in R$ and $q_j = 0$ for every $j \in C$. The *tight coefficient matrix* of $A(s)$ is defined to be the constant matrix $A^{\#} = (A^{\#}_{ij})$ with $A^{\#}_{ij}$ being the coefficient of $s^{p_i - q_j}$ in $A_{ij}(s)$. The following lemma enables us to check $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ and $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ efficiently.

LEMMA 3.5. *The tight coefficient matrix* $A^{\#}$ *is nonsingular if and only if both* $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$ *and* $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$ *hold.*

Proof. Note that $\det A(s) = s^{\Delta_n(p,q)}\{\det A^{\#} + \mathrm{o}(1)\}$ holds, where $\mathrm{o}(1)$ denotes an expression consisting of negative powers of s. Therefore, if $\delta_n(A) = \Delta_n(p, q)$, then $A^{\#}$ must be nonsingular. Conversely, if $A^{\#}$ is nonsingular, then $\delta_n(A) = \Delta_n(p, q)$, which together with (3.1) implies $\delta_n(A) = \hat{\delta}_n(A) = \Delta_n(p, q)$. The nonsingularity of $A^{\#}$ further implies that there exists a nonsingular submatrix $A^{\#}[I, J]$ such that $|I| = |J| = n-1$ and $I \supseteq R^*$, where $R^* = \{i \in R \mid p_i > \min_{\ell \in R} p_\ell\}$. Since $\det A(s)[I, J] = s^{\Delta_{n-1}(p,q)}\{\det A^{\#}[I, J] + \mathrm{o}(1)\}$, we have $\delta_{n-1}(A) \geq \Delta_{n-1}(p, q)$, which together with (3.1) implies $\delta_{n-1}(A) = \hat{\delta}_{n-1}(A) = \Delta_{n-1}(p, q)$. ∎

By Lemma 3.5, we can perform Step 4 by checking the nonsingularity of $A^{\#}$. In numerical computation, floating-point operations induce round-off errors, which make it nontrivial to check the nonsingularity. A common approach to obviate this difficulty is to use the singular value decomposition [27]. In fact, the MATLAB function called `rank` returns the number of singular values that are larger than a tolerance.

3.4. Unimodular transformations in Step 5. Let $A(s)$ be a matrix pencil in Step 5. The algorithm has detected that the condition in Step 4 is not fulfilled, i.e.,

the tight coefficient matrix $A^{\#}$ is singular. Hence there exists a nonzero row vector $\boldsymbol{u} = (u_i \mid i \in R)$ such that

$$\boldsymbol{u}A^{\#} = \mathbf{0}.$$

By executing the Gaussian elimination on $A^{\#}$ with column transformations, we can find $\boldsymbol{u}$ such that $\operatorname{supp} \boldsymbol{u} := \{i \in R \mid u_i \neq 0\}$ is minimal with respect to set inclusion.

By the definition of $A^{\#}$, we have $A^{\#}[R_0, C] = A(s)[R_0, C]$. Since $A(s)$ is regular, $A^{\#}[R_0, C]$ is of full row-rank. This implies that there exists $\ell \in \operatorname{supp} \boldsymbol{u}$ with $p_\ell = 1$.

We now define U by

$$U_{ik} = \begin{cases} u_k/u_\ell & (i = \ell), \\ 1 & (k = i \neq \ell), \\ 0 & (k \neq i \neq \ell). \end{cases}$$

We remark that the row set and the column set of U correspond to $R_1 \cup R_0$ and $U[R_0, R_1] = O$. We denote by $\operatorname{diag}(s; p)$ the square diagonal matrix with each (i, i) entry being s^{p_i}. Then the polynomial matrix $U(s) = \operatorname{diag}(s; p) \cdot U \cdot \operatorname{diag}(s; -p)$ is unimodular.

Since $A(s)$ can be expressed as

$$A(s) = \operatorname{diag}(s; p) \cdot \left(A^{\#} + \frac{1}{s} \begin{pmatrix} A(0)[R_1, C] \\ O \end{pmatrix} \right),$$

it holds that

$$\begin{aligned} U(s)A(s) &= \operatorname{diag}(s; p) \cdot U \cdot \left(A^{\#} + \frac{1}{s} \begin{pmatrix} * \\ O \end{pmatrix} \right) = \operatorname{diag}(s; p) \cdot \left(UA^{\#} + \frac{1}{s} \begin{pmatrix} * & * \\ O & * \end{pmatrix} \begin{pmatrix} * \\ O \end{pmatrix} \right) \\ &= \operatorname{diag}(s; p) \cdot \left(UA^{\#} + \frac{1}{s} \begin{pmatrix} * \\ O \end{pmatrix} \right) = \operatorname{diag}(s; p) \cdot UA^{\#} + \begin{pmatrix} * \\ O \end{pmatrix}, \end{aligned}$$

where $*$ denotes a constant matrix. Hence $U(s)A(s)$ remains to be a matrix pencil. Since the ℓth row vector of $UA^{\#}$ is zero, $U(s)A(s)$ does not contain s in the ℓth row. Hence we can decrease $p_\ell = 1$ by one. By setting

$$\hat{A}(s) := U(s)A(s), \quad \hat{p}_i := \begin{cases} 0 & (i = \ell), \\ p_i & (i \neq \ell), \end{cases} \quad \hat{q} := q,$$

we obtain another matrix pencil $\hat{A}(s)$ and feasible solution $(\hat{p}, \hat{q})$.

LEMMA 3.6. *The number of iterations in Phase* 2 *is at most* n.

Proof. At each iteration, the number of rows with $p_i = 0$ increases by one. ∎

At the end of the index reduction algorithm, we obtain a matrix pencil with index at most one.

THEOREM 3.7. *The algorithm finds in* $\mathrm{O}(n^4)$ *time a unimodular matrix* $U(s)$ *such that the Kronecker index of* $\bar{A}(s) = U(s)A(s)$ *is at most one.*

Proof. When the algorithm terminates, we obtain $\bar{A}(s)$ and an optimal solution $(\bar{p}, \bar{q})$ of $\mathrm{D}(\bar{A})$ satisfying (3.2) and (3.3) by Lemmas 3.3 and 3.5. Let m denote the

number of rows with $p_i = 1$. Then we have (3.7) for $(\bar{p}, \bar{q})$. Hence the Kronecker index $\nu(\bar{A})$ is given by

$$\nu(\bar{A}) = \delta_{n-1}(\bar{A}) - \delta_n(\bar{A}) + 1 = \Delta_{n-1}(\bar{p}, \bar{q}) - \Delta_n(\bar{p}, \bar{q}) + 1 = \begin{cases} 1 & (m < n), \\ 0 & (m = n). \end{cases}$$

Thus the index of $\bar{A}(s)$ is at most one.

In Step 1, we solve a maximum-weighted perfect matching problem. This can be performed in $O(n^3)$ time by the Hungarian method [11, 15, 25]. Steps 3 and 5 require the Gaussian elimination, which costs $O(n^3)$ time at each iteration. Since the number of iterations of Steps 3 and 5 is $O(n)$ by Lemmas 3.3 and 3.6, the total time complexity is $O(n^4)$. ∎

4. Examples. We give three examples below.

Example 4.1. The following is a famous example for which Pantelides' algorithm does not work:

$$\begin{aligned} z_1 - \dot{z}_1 + 2z_2 + 3z_3 &= 0, \\ z_1 + z_2 + z_3 + 1 &= 0, \\ 2z_1 + z_2 + z_3 &= 0. \end{aligned}$$

The corresponding matrix pencil $A(s)$ is expressed as

$$A(s) = \begin{pmatrix} -s+1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}.$$

By $\delta_2(A) = 1$ and $\delta_3(A) = 0$, we have $\nu(A) = 2$. However, when we apply Pantelides' algorithm [20] to $A(s)$, the algorithm terminates without detecting equations to be differentiated. Pantelides' algorithm is adopted in the MATLAB function called `reduceDAEIndex`. In fact, this function does not work for the above DAE.

Let us apply our algorithm to $A(s)$. In Step 1, we find an optimal solution $p = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$ of $D(A)$. In Step 3, we obtain another solution $p = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ without changing $A(s)$. Then we go to Step 4 by $q = \mathbf{0}$. The tight coefficient matrix $A^{\#} = \left(\begin{smallmatrix} -1 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{smallmatrix}\right)$ is singular. In Step 5, we have $\boldsymbol{u} = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix}$ and $U(s) = \left(\begin{smallmatrix} 1 & -s & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$. The matrix pencil $A(s)$ is transformed into $\bar{A}(s) = U(s)A(s) = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{smallmatrix}\right)$ with $p = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$. Then we obtain $\nu(\bar{A}) = 1$.

Example 4.2. Next, consider another matrix pencil

$$A(s) = \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & s \end{pmatrix}.$$

It follows from $\delta_3(A) = 2$ and $\delta_4(A) = 0$ that $\nu(A) = 3$. We apply the algorithm described in section 3 to $A(s)$.

In Step 1, we find an optimal solution $p = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$ and $q = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$ of $D(A)$. Then we go to Step 3 by $q \neq \mathbf{0}$. In Step 3, we delete s in the last row

by row transformations and obtain a feasible dual solution $p' = \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix}$ and $q' = \begin{pmatrix}0 & 0 & 0 & 0\end{pmatrix}$ as follows:

$$A(s) = \begin{matrix} \\ R_1 \\ \\ \\ \end{matrix}\!\!\begin{matrix} \begin{matrix} C_1 & C_0 \end{matrix} \\ \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & s \end{pmatrix} \end{matrix} \quad \longrightarrow \quad A'(s) = U^{\circ}(s)A(s) = \begin{matrix} \\ R_1 \\ \\ R_0 \end{matrix}\!\!\begin{matrix} \begin{matrix} C_0 \end{matrix} \\ \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & s \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix},$$

where $U^{\circ}(s) = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{smallmatrix}\right)$. We return to Step 2 and then go to Step 4 by $q' = \mathbf{0}$. The tight coefficient matrix $A^{\#} = \left(\begin{smallmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{smallmatrix}\right)$ is singular in Step 4, and we have $\boldsymbol{u}' = \begin{pmatrix}0 & 1 & -1 & 1\end{pmatrix}$ and $U'(s) = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -s & s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right)$ in Step 5. The matrix pencil $A'(s)$ is transformed into

$$A''(s) = U'(s)A'(s) = \begin{matrix} \\ R_1 \\ \\ R_0 \end{matrix}\!\!\begin{matrix} \begin{matrix} C_0 \end{matrix} \\ \begin{pmatrix} 0 & 1 & s & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

with $p'' = \begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}$ and $q'' = \begin{pmatrix}0 & 0 & 0 & 0\end{pmatrix}$.

Returning to Step 4, the tight coefficient matrix $A^{\#} = \left(\begin{smallmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{smallmatrix}\right)$ is also singular. In Step 5, we have $\boldsymbol{u}'' = \begin{pmatrix}1 & -1 & 0 & 0\end{pmatrix}$ and $U''(s) = \left(\begin{smallmatrix} 1 & -s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right)$. The matrix pencil $A''(s)$ is transformed into

$$\bar{A}(s) = U''(s)A''(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

with $\bar{p} = \begin{pmatrix}0 & 0 & 0 & 0\end{pmatrix}$ and $\bar{q} = \begin{pmatrix}0 & 0 & 0 & 0\end{pmatrix}$. Returning to Step 4, the tight coefficient matrix $A^{\#} = \bar{A}(s)$ is nonsingular and hence we terminate the algorithm.

As a result, we obtain a unimodular matrix $U(s)$ and a matrix pencil $\bar{A}(s)$ with $\nu(\bar{A}) = 1$ expressed as

$$U(s) = U''(s)U'(s)U^{\circ}(s) = \begin{pmatrix} 1 & s^2 - s & s^2 & -s^2 \\ 0 & -s + 1 & -s & s \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}, \quad \bar{A}(s) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

In Examples 4.1 and 4.2, we have obtained a constant matrix $\bar{A}(s)$, which means that the corresponding DAEs are systems of algebraic equations. This is not always the case. We show a simple example which leads to $\bar{A}(s)$ containing s.

Example 4.3. Consider a matrix pencil

$$A(s) = \begin{pmatrix} 1 & 0 & 0 \\ s & 0 & 1 \\ 0 & s & s \end{pmatrix}.$$

By $\delta_2(A) = 2$ and $\delta_3(A) = 1$, we have $\nu(A) = 2$.

In Step 1, we find an optimal solution $p = \begin{pmatrix} 0 & 1 & 2 \end{pmatrix}$ and $q = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$ of $\mathrm{D}(A)$. Then we go to Step 3 by $q \neq \mathbf{0}$. In Step 3, we delete s in the second row by row transformations and obtain a feasible dual solution $\bar{p} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$ and $\bar{q} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ as follows:

$$A(s) = \begin{pmatrix} 1 & 0 & 0 \\ s & 0 & 1 \\ 0 & s & s \end{pmatrix} \quad \longrightarrow \quad \bar{A}(s) = U(s)A(s) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & s & s \end{pmatrix},$$

where $U(s) = \left(\begin{smallmatrix} 1 & 0 & 0 \\ -s & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$. We return to Step 2 and then go to Step 4 by $\bar{q} = \mathbf{0}$. Since the tight coefficient matrix $A^{\#} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right)$ is nonsingular in Step 4, we terminate the algorithm.

5. Conclusion. We have presented a new index reduction algorithm of matrix pencils which makes use of unimodular transformations. The algorithm is based on the framework of combinatorial relaxation, which combines graph-algorithmic techniques and matrix computation. Our algorithm can be used as an index reduction method for linear constant coefficient DAEs. It works correctly for any such DAEs including those for which Pantelides' algorithm or the signature method are known to fail. An extension of our algorithm to index reduction of nonlinear DAEs is left for future investigation.

REFERENCES

[1] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, 1998.

[2] T. Beelen and P. Van Dooren, *An improved algorithm for the computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 105 (1988), pp. 9–65.

[3] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, 2nd ed., Classics in Appl. Math., SIAM, Philadelphia, 1996.

[4] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A-\lambda B$: Robust software with error bounds and applications. Part* I*: Theory and algorithms*, ACM Trans. Math. Software, 19 (1993), pp. 160–174.

[5] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A-\lambda B$: Robust software with error bounds and applications. Part* II*: Software and applications*, ACM Trans. Math. Software, 19 (1993), pp. 175–201.

[6] F. R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 1959.

[7] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II, 2nd ed., Springer-Verlag, Berlin, 1996.

[8] S. Iwata, *Computing the maximum degree of minors in matrix pencils via combinatorial relaxation*, Algorithmica, 36 (2003), pp. 331–341.

[9] S. Iwata, K. Murota, and I. Sakuta, *Primal-dual combinatorial relaxation algorithms for the maximum degree of subdeterminants*, SIAM J. Sci. Comput., 17 (1996), pp. 993–1012.

[10] B. Kågström, *RGSVD—an algorithm for computing the Kronecker structure and reducing subspaces of singular $A - \lambda B$ pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.

[11] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart., 2 (1955), pp. 83–97.

[12] P. Kunkel and V. Mehrmann, *Differential-Algebraic Equations: Analysis and Numerical Solutions*, European Mathematical Society, Zürich, 2006.
[13] R. Lamour, R. März, and C. Tischendorf, *Differential-Algebraic Equations: A Projector Based Analysis*, Springer-Verlag, Berlin, 2013.
[14] S. E. Mattsson and G. Söderlind, *Index reduction in differential-algebraic equations using dummy derivatives*, SIAM J. Sci. Comput., 14 (1993), pp. 677–692.
[15] J. Munkres *Algorithms for the assignment and transportation problems*, J. SIAM, 5 (1957), pp. 32–38.
[16] K. Murota, *Computing Puiseux-series solutions to determinantal equations via combinatorial relaxation*, SIAM J. Comput., 19 (1990), pp. 1132–1161.
[17] K. Murota, *Combinatorial relaxation algorithm for the maximum degree of subdeterminants: Computing Smith-McMillan form at infinity and structural indices in Kronecker form*, Appl. Algebra Engrg. Comm. Comput., 6 (1995), pp. 251–273.
[18] K. Murota, *Computing the degree of determinants via combinatorial relaxation*, SIAM J. Comput., 24 (1995), pp. 765–796.
[19] K. Murota, *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin, 2000.
[20] C. C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 213–231.
[21] J. D. Pryce, *A simple structural analysis method for DAEs*, BIT, 41 (2001), pp. 364–394.
[22] R. Riaza, *Differential-Algebraic Systems: Analytical Aspects and Circuit Applications*, World Scientific, Singapore, 2008.
[23] S. Sato, *Combinatorial relaxation algorithm for the entire sequence of the maximum degree of minors*, Algorithmica, 77 (2017), pp. 815–835.
[24] G. Tan, N. S. Nedialkov, and J. D. Pryce, *Symbolic-numeric methods for improving structural analysis of differential-algebraic equation systems*, in Mathematical and Computational Approaches in Advancing Modern Science and Engineering, J. Bélair et al., eds., Springer, New York, 2016, pp. 763–773.
[25] N. Tomizawa, *On some techniques useful for solution of transportation network problems*, Networks, 1 (1971), pp. 173–194.
[26] P. Van Dooren, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–140.
[27] D. S. Watkins, *Fundamentals of Matrix Computations*, 2nd ed., John Wiley & Sons, New York, 2002.