11.

Primal-Dual Combinatorial Relaxation Algorithms for the Maximum Degree of Subdeterminants

Satoru IWATA 岩田 覚
Kazuo MUROTA 室田 一雄
Izumi SAKUTA 作田 泉

Abstract. A primal-dual framework of combinatorial relaxation algorithms is proposed for computing the highest degree of a minor of order $k$ of a rational function matrix. The algorithm can be used for computing the index of nilpotency of a matrix pencil (or the index of the associated DAE). It is a linear algebraic version of the Hungarian method for the assignment problem. The proposed framework stands in contrast with the previous combinatorial relaxation algorithm based on weighted matchings, and may also be regarded as an extension of the Wolovich algorithm for row/column properness. Several algorithms are evaluated through computer experiments.

Key words. combinatorial relaxation, degree of subdeterminant, index of DAE, Kronecker form, Smith-McMillan form at infinity

11.1 Introduction

Let $A(x) = (A_{ij}(x))$ be an $m \times n$ rational function matrix with $A_{ij}(x)$ being a rational function in $x$ with coefficients from the real number field $\mathbf{R}$. This paper deals with algorithms for computing the highest degree of a minor of order $k$ ($\leq \min(m,n)$) of $A(x)$:

$$\delta_k(A) = \max_{I,J}\{\deg\det A[I,J] \mid |I| = |J| = k\}, \tag{1}$$

where $A[I,J]$ denotes the submatrix of A with row-set I and column-set J and the degree of a rational function $f(x) = p(x)/q(x)$ is defined by $\deg f(x) = \deg p(x) - \deg q(x)$. By convention we put $\deg f(x) = -\infty$ if $f(x) = 0$. This is one of the fundamental problems in combinatorial matrix theory [3] and has various applications in practice.

By computing $\delta_k(A)$ $(k = 1,2,\cdots)$ we can obtain the Smith-McMillan form at infinity that is known also as the structure at infinity in the literature of control theory (Commault–Dion[5], Hautus[12], Verghese–Kailath[22]). When $A(x)$ is a regular pencil of order n, on the other hand, $\delta_k(A)$ $(k = 1,2,\cdots)$ determines the structural indices of its Kronecker form (see Murota[15,Section 2.2]), and hence the index ν as well as the dynamical degree of freedom of the associated differential-algebraic equations (DAEs) (Brenan–Campbell–Petzold[2], Gear[9,10], Hairer–Wanner[11]). To be specific, the index is given by $\nu = \delta_{n-1}(A) - \delta_n(A) + 1$, whereas the dynamical degree of freedom is equal to $\delta_n(A)$. Thus the problem of computing $\delta_k(A)$ $(k = 1,2,\cdots)$ has a fundamental engineering significance.

The present paper aims to propose a primal-dual framework of combinatorial relaxation algorithms for computing the maximum degree of subdeterminants. Like the Hungarian method for the assignment problem [1,20], the algorithms keep a set of admissible potentials (feasible dual variables), which determines the estimate (dual objective value), and check its tightness (dual optimality) by solving a simpler (restricted primal) problem. If it turns out to be nontight, we obtain information useful for modifying the potentials as well as the matrix. Thus the proposed framework can be interpreted as a linear algebraic version of the Hungarian method. It stands in contrast with the previous combinatorial relaxation algorithm based on weighted-matchings, and may also be regarded as an extension of the Wolovich algorithm for row/column properness. Several algorithms are evaluated through computer experiments.

11.2 Combinatorial relaxation algorithms

In this section, we present a new framework of combinatorial relaxation type algorithms for computing δ_k. This shares the approach of "combinatorial relaxation" with the previous algorithm of Murota[15], while featuring potentials (dual variables) rather than matchings in constructing the combinatorial estimate. The proposed framework may be thought of as a linear algebraic version of the primal-dual approach to the assignment problem that is well known in the context of combinatorial optimization[1,20].

11.2.1 Outline of the proposed framework

For a rational function matrix $A(x)$, let R and C denote the row-set and the column-set, respectively. By potentials we mean integer numbers associated with rows or columns. With appropriately chosen potentials, say, $\rho = (\rho_i \mid i \in R)$ and $\gamma = (\gamma_j \mid j \in C)$, we compute an estimate $\widehat{\delta}_k(A;\rho,\gamma)$ of $\delta_k(A)$ such that $\delta_k(A) \leq \widehat{\delta}_k(A;\rho,\gamma)$. We then test for tightness (equality) in this inequality without computing $\delta_k(A)$. If the estimate turns out to be nontight, we modify the matrix $A(x)$ as well as the potentials.

The outline of the combinatorial relaxation algorithm to compute δ_k for a fixed k is summarized as follows:

Phase 0: Choose appropriate (ρ,γ).

Phase 1: Compute the combinatorial upper estimate $\widehat{\delta}_k(A;\rho,\gamma)$.

Phase 2: Test whether $\widehat{\delta}_k(A;\rho,\gamma) = \delta_k(A)$ or not. If the equality holds, output $\widehat{\delta}_k(A;\rho,\gamma)$ and stop.

Phase 3: Modify A to another matrix A' and (ρ,γ) to (ρ',γ') such that $\delta_k(A') = \delta_k(A)$ and $\widehat{\delta}_k(A';\rho',\gamma') < \widehat{\delta}_k(A;\rho,\gamma)$, namely A' has the same true value as A and an improved estimate. Put $A := A'$, $\rho := \rho'$, $\gamma := \gamma'$ and go to Phase 1.

11.2.2 Phase 1 — Combinatorial estimation

The combinatorial estimate, denoted as $\widehat{\delta}_k(A;\rho,\gamma)$ in Section 11.2.1, is specified in this section. Define c_{ij} by

$$c_{ij} = \deg A_{ij}(x), \tag{1}$$

where by convention we put $c_{ij} = -\infty$ if $A_{ij}(x) = 0$. We choose a set of potentials $\rho = (\rho_i \mid i \in R)$ and $\gamma = (\gamma_j \mid j \in C)$ which satisfies the condition

$$\rho_i + \gamma_j \geq c_{ij} \quad \text{for } i \in R \text{ and } j \in C. \tag{2}$$

Such potentials are said to be admissible. Then for $I \subseteq R$ and $J \subseteq C$ with $|I| = |J|$ it holds that

$$\deg\det A[I,J] \leq \rho(I) + \gamma(J), \tag{3}$$

where $\rho(I) = \sum_{i\in I} \rho_i$ and $\gamma(J) = \sum_{j\in J} \gamma_j$.

By considering the maximum possible value of the right-hand side, i.e., by introducing

$$\widehat{\delta}_k(A;\rho,\gamma) = \max_{|I|=k} \rho(I) + \max_{|J|=k} \gamma(J),$$

we obviously have

$$\rho(I) + \gamma(J) \le \widehat{\delta}_k(A;\rho,\gamma) \tag{4}$$

for $I \subseteq R$ and $J \subseteq C$ with $|I| = |J| = k$. The inequalities (3) and (4) imply

$$\delta_k(A) \le \widehat{\delta}_k(A;\rho,\gamma), \tag{5}$$

which shows that $\widehat{\delta}_k(A;\rho,\gamma)$ serves as an upper bound on the true value. Remember that (ρ,γ) must be admissible in the sense of (2) before (5) can be true.

The estimate $\widehat{\delta}_k(A;\rho,\gamma)$ admits an alternative expression suitable for its efficient computation. Consider the following permutations for the row-set R and the column-set C of A:

$$\tau : \{1,\ldots,m\} \to R \qquad \text{such that } \rho_{\tau(1)} \ge \rho_{\tau(2)} \ge \cdots \ge \rho_{\tau(m)}; \tag{6}$$

$$\sigma : \{1,\ldots,n\} \to C \qquad \text{such that } \gamma_{\sigma(1)} \ge \gamma_{\sigma(2)} \ge \cdots \ge \gamma_{\sigma(n)}. \tag{7}$$

Then it is easy to see that

$$\widehat{\delta}_k(A;\rho,\gamma) = \sum_{i=1}^{k} \rho_{\tau(i)} + \sum_{j=1}^{k} \gamma_{\sigma(j)}.$$

11.2.3 Phase 2 — Test for tightness

This section provides a way to test whether $\delta_k(A) = \widehat{\delta}_k(A;\rho,\gamma)$ or not without computing $\delta_k(A)$ directly.

Define $I^\sharp, J^\sharp, \tilde{I}, \tilde{J}$ as follows:

$$I^\sharp = \{i \in R \mid \rho_i > \rho_{\tau(k+1)}\}, \qquad \tilde{I} = \{i \in R \mid \rho_i \ge \rho_{\tau(k)}\}; \tag{8}$$

$$J^\sharp = \{j \in C \mid \gamma_j > \gamma_{\sigma(k+1)}\}, \qquad \tilde{J} = \{j \in C \mid \gamma_j \ge \gamma_{\sigma(k)}\}. \tag{9}$$

Exceptionally, in case $k = m$, we put $I^\sharp = \tilde{I} = R$ and in case $k = n$, we put $J^\sharp = \tilde{J} = C$. Then $\rho(I)$ attains the maximum subject to $|I| = k$ if and only if $I^\sharp \subseteq I \subseteq \tilde{I}$. Similarly $\gamma(J)$ is maximized subject to $|J| = k$ if and only if $J^\sharp \subseteq J \subseteq \tilde{J}$. Therefore (4) holds with equality if and only if $I^\sharp \subseteq I \subseteq \tilde{I}$ and $J^\sharp \subseteq J \subseteq \tilde{J}$. Define a constant matrix $A^\sharp$ by

$$A^\sharp = (A^\sharp_{ij}), \quad A^\sharp_{ij} = \begin{cases} \lim_{x\to\infty} x^{-\rho_i-\gamma_j} A_{ij}(x) & \text{if } i \in \tilde{I} \text{ and } j \in \tilde{J}; \\ 0 & \text{otherwise.} \end{cases}$$

命題 11.2.1. (tightness) The following three conditions (a)–(c) are equivalent.

(a) $\delta_k(A) = \widehat{\delta}_k(A;\rho,\gamma)$.

(b) There exist $I \supseteq I^\sharp$ and $J \supseteq J^\sharp$ such that $\operatorname{rank} A^\sharp[I,J] = |I| = |J| = k$.

(c) The following four conditions (r1)–(r4) are satisfied:

(r1) $\operatorname{rank} A^\sharp[R,C] \geq k$;

(r2) $\operatorname{rank} A^\sharp[I^\sharp,C] = |I^\sharp|$;

(r3) $\operatorname{rank} A^\sharp[R,J^\sharp] = |J^\sharp|$;

(r4) $\operatorname{rank} A^\sharp[I^\sharp,J^\sharp] \geq |I^\sharp| + |J^\sharp| - k$.

According to Proposition 11.2.1. above, the test for $\widehat{\delta}_k = \delta_k$ in Phase 2 can be reduced to computing the ranks of the four constant matrices. Actually, for (r1), (r2) and (r4), we use the Gaussian elimination by elementary row transformations, and for (r3), we use the Gaussian elimination by elementary column transformations.

11.2.4 Phase 3 — Modification

When the estimate $\widehat{\delta}_k(A;\rho,\gamma)$ has turned out to be nontight, we modify the matrix as well as the potentials. Recall that the maximum degree of subdeterminants of a fixed order is invariant under biproper equivalence transformations. That is,

$$A'(x) = U(x)A(x)V(x) \tag{10}$$

satisfies $\delta_k(A') = \delta_k(A)$ if both $U(x)$ and $V(x)$ are biproper. We use this type of transformation in the modification of the matrix $A(x)$.

Let U be a nonsingular constant matrix of order m such that

$$U_{hi} = 0 \quad \text{if } \rho_h > \rho_i,$$

and define a transformation matrix $U(x)$ by

$$U(x) = \operatorname{diag}(x;\rho) \cdot U \cdot \operatorname{diag}(x;-\rho), \tag{11}$$

where $\operatorname{diag}(x;\rho)$ designates the diagonal matrix $\operatorname{diag}(x^{\rho_1}, x^{\rho_2}, \cdots, x^{\rho_m})$. Then $U(x)$ is a biproper matrix. Similarly, using a nonsingular constant matrix V that satisfies the condition:

$$V_{jh} = 0 \quad \text{if } \gamma_h > \gamma_j,$$

we define a transformation matrix $V(x)$ by

$$V(x) = \operatorname{diag}(x;-\gamma) \cdot V \cdot \operatorname{diag}(x;\gamma), \tag{12}$$

which is also a biproper matrix.

We now explain how to construct such a pair of constant matrices U and V efficiently according to which of the four conditions (r1)–(r4) is violated by the constant matrix $A^\sharp$ (see Proposition 11.2.1. (c)).

When (r1) is violated: In the course of checking the condition (r1) by means of the Gaussian row elimination with column pivoting, applied to the matrix $A^\sharp$ with rows rearranged by τ according to the magnitude of ρ_i, we obtain a nonsingular matrix S of order m such that $S_{\tau(l)\tau(i)} = 0$ if $l < i$ and that $SA^\sharp$ has $(m - \operatorname{rank} A^\sharp)$ zero vectors among its row vectors. Let H be a subset of $\tilde{I}$ such that $|H| = |\tilde{I}| - k + 1$ and $(SA^\sharp)[H, C] = O$. Construct the constant matrix U as follows:

$$U[H, R] = S[H, R],\ U[R - H, H] = O,\ U[R - H, R - H] = I_{|R-H|}, \tag{13}$$

where $I_{|R-H|}$ denotes the unit matrix. Let V be the unit matrix of order n.

When (r2) is violated: In checking the condition (r2) by the Gaussian row elimination with column pivoting, applied to the matrix $A^\sharp[I^\sharp, C]$ with rows rearranged by τ according to the magnitude of ρ_i, we obtain a nonsingular matrix S of order $|I^\sharp|$ such that $S_{\tau(l)\tau(i)} = 0$ if $l < i$ and that $S \cdot A^\sharp[I^\sharp, C]$ has $(|I^\sharp| - \operatorname{rank} A^\sharp[I^\sharp, C])$ zero row vectors. Let $H \subseteq I^\sharp$ be the singleton set corresponding to a zero row vector in $S \cdot A^\sharp[I^\sharp, C]$. Then construct U as in (13). Let V be the unit matrix of order n.

When (r3) is violated: In checking the condition (r3) by the Gaussian column elimination with row pivoting, applied to the matrix $A^\sharp[R, J^\sharp]$ with columns rearranged by σ according to the magnitude of γ_j, we obtain a nonsingular matrix S of order $|J^\sharp|$ such that $S_{\sigma(j)\sigma(l)} = 0$ if $l < j$ and that $A^\sharp[R, J^\sharp] \cdot S$ has $(|J^\sharp| - \operatorname{rank} A^\sharp[R, J^\sharp])$ zero column vectors. Construct the constant matrix V of order n in a similar (transposed) way as we constructed U in the case of (r2), namely

$$V[C, H] = S[C, H],\ V[H, C - H] = O,\ V[C - H, C - H] = I_{|C-H|}. \tag{14}$$

Let U be the unit matrix of order m.

When (r4) is violated: In the course of checking the condition (r4) by means of the Gaussian row elimination with column pivoting, applied to the matrix $A^\sharp[I^\sharp, J^\sharp]$ with rows rearranged by τ according to the magnitude of ρ_i, we obtain a nonsingular matrix S of order $|I^\sharp|$ such that $S_{\tau(l)\tau(i)} = 0$ if $l < i$ and that $S \cdot A^\sharp[I^\sharp, J^\sharp]$ has $(|I^\sharp| - \operatorname{rank} A^\sharp[I^\sharp, J^\sharp])$ zero row vectors. Let H be a subset of $I^\sharp$ corresponding to $(k - |J^\sharp| + 1)$ zero row vectors in $S \cdot A^\sharp[I^\sharp, J^\sharp]$. Then construct the constant matrix U in the same way as in (13). Let V be the unit matrix of order n.

After modifying the matrix $A(x)$ to $A'(x)$ by (10) with $U(x)$ and $V(x)$ thus constructed, the set of potentials is updated as follows. Put $c'_{ij} = \deg A'_{ij}(x)$ similarly to (1). It follows from (10), (11) and (12) that

$$\mathrm{diag}\,(x;-\rho)\cdot A'(x)\cdot\mathrm{diag}\,(x;-\gamma)=U\cdot\mathrm{diag}\,(x;-\rho)\cdot A(x)\cdot\mathrm{diag}\,(x;-\gamma)\cdot V,$$

which, together with (2), implies

$$\rho_i+\gamma_j\geq c'_{ij}\quad\text{for } i\in R\quad\text{and } j\in C. \tag{15}$$

We update the set of potentials to improve the estimate exploiting the fact that (15) above holds with strict inequality for some $i \in R$ and $j \in C$. Roughly speaking, we reduce the potentials of such rows or columns without violating the admissibility. More specific description is given below according to which condition of the four is violated.

When (r1) is violated:

$$\rho'_i = \begin{cases} \max\{\max\limits_{j\in\tilde{J}}(c'_{ij}-\gamma_j),\ \max\limits_{j\in(C-\tilde{J})} c'_{ij}-\gamma_{\sigma(k)}\} & \text{if } i\in\tilde{I}; \\ \rho_i & \text{otherwise}; \end{cases} \tag{16}$$

$$\gamma'_j = \max_{i\in R}(c'_{ij}-\rho'_i). \tag{17}$$

When (r2) is violated:

$$\rho'_i = \begin{cases} \max\{\max\limits_{j\in\tilde{J}}(c'_{ij}-\gamma_j),\ \max\limits_{j\in(C-\tilde{J})} c'_{ij}-\gamma_{\sigma(k)}\} & \text{if } i\in I^\sharp; \\ \rho_i & \text{otherwise}; \end{cases} \tag{18}$$

$$\gamma'_j = \max_{i\in R}(c'_{ij}-\rho'_i). \tag{19}$$

When (r3) is violated:

$$\gamma'_j = \begin{cases} \max\{\max\limits_{i\in\tilde{I}}(c'_{ij}-\rho_i),\ \max\limits_{i\in(R-\tilde{I})} c'_{ij}-\rho_{\tau(k)}\} & \text{if } j\in J^\sharp; \\ \gamma_j & \text{otherwise}; \end{cases} \tag{20}$$

$$\rho'_i = \max_{j\in C}(c'_{ij}-\gamma'_j). \tag{21}$$

When (r4) is violated:

$$\rho'_i = \begin{cases} \rho_i-\beta & \text{if } i\in H; \\ \rho_i & \text{otherwise,} \end{cases} \qquad \gamma'_j = \begin{cases} \gamma_j & \text{if } j\in J^\sharp; \\ \gamma_j+\beta & \text{otherwise,} \end{cases} \tag{22}$$

where

$$\beta=\min\{\rho_{\tau(|I^\sharp|)}-\rho_{\tau(|I^\sharp|+1)},\quad \gamma_{\sigma(|J^\sharp|)}-\gamma_{\sigma(|J^\sharp|+1)},\quad \min_{h\in H,j\in J^\sharp}(\rho_h+\gamma_j-c'_{hj})\}. \tag{23}$$

Exceptionally, the first term is excluded from the minimization when $I^\sharp = R$ and the second is excluded when $J^\sharp = C$.

As to the admissibility of the updated set of potentials, the following lemma holds.

補題 11.2.2. The updated set of potentials (ρ', γ') is admissible with respect to the modified matrix A'. That is, $\rho'_i + \gamma'_j \geq c'_{ij}$ for $i \in R$ and $j \in C$.

The following theorem shows that the estimate is improved by the modification described above.

定理 11.2.3. The following inequality holds:

$$\widehat{\delta}_k(A'; \rho', \gamma') < \widehat{\delta}_k(A; \rho, \gamma). \tag{24}$$

11.2.5 Algorithm description

We have so far fixed the size k of a submatrix. It is necessary to compute δ_k for $k = 1, 2, \ldots, r$ for obtaining the Smith-McMillan form at infinity.

Before describing the whole algorithm, we should remark how we can detect $\delta_k = -\infty$. For a rational function

$$f(x) = \frac{a_1 x^{d_1} + a_2 x^{d_2} + \cdots + a_\ell x^{d_\ell}}{q(x)} \quad (d_1 > d_2 > \cdots > d_\ell),$$

we define min-deg $f(x) \equiv d_\ell - \deg q(x)$ with respect to its expression. Put $d_{\min} = \min_{i,j}$ min-deg $A_{ij}(x)$ while $d_{\max} = \max_{i,j} \deg A_{ij}(x)$. If δ_k is finite, it must satisfy $\delta_k \geq k \cdot d_{\min}$. Hence we can conclude that $\delta_k = -\infty$ when $\widehat{\delta}_k(A; \rho, \gamma) < k \cdot d_{\min}$ for some admissible (ρ, γ).

We now summarize the algorithm for computing δ_k for $k = 1, 2, \ldots, r (= \operatorname{rank} A)$ as follows. It is not necessary to input r, since it is computed in the procedure. **Algorithm for computing $\delta_k(A)$** $(k = 1, 2, \ldots)$

Step 0: Compute $d_{\min}$. Set $k := 1$. Find a set of initial potentials (ρ, γ) which satisfies $\rho_i + \gamma_j \geq c_{ij}$ for $i \in R$ and $j \in C$.

Step 1: Compute

$$\widehat{\delta}_k := \sum_{i=1}^{k} \rho_{\tau(i)} + \sum_{j=1}^{k} \gamma_{\sigma(j)},$$

where τ and σ are permutations such that $\rho_{\tau(1)} \geq \rho_{\tau(2)} \geq \cdots \geq \rho_{\tau(m)}$ and $\gamma_{\sigma(1)} \geq \gamma_{\sigma(2)} \geq \cdots \geq \gamma_{\sigma(n)}$. If $\widehat{\delta}_k < k \cdot d_{\min}$, then halt (at this point $(k-1)$ equals the rank r).

Step 2: Put

$$I^\sharp := \{i \in R \mid \rho_i > \rho_{\tau(k+1)}\}, \qquad \tilde{I} := \{i \in R \mid \rho_i \geq \rho_{\tau(k)}\};$$

$$J^\sharp := \{j \in C \mid \gamma_j > \gamma_{\sigma(k+1)}\}, \qquad \tilde{J} := \{j \in C \mid \gamma_j \geq \gamma_{\sigma(k)}\};$$

where exceptionally $I^\sharp = R$ and $J^\sharp = C$ in case $k = m$ and $k = n$, respectively. Put

$$A^\sharp_{ij} := \begin{cases} \lim\limits_{x \to \infty} x^{-\rho_i - \gamma_j} A_{ij}(x) & \text{if } i \in \tilde{I} \text{ and } j \in \tilde{J}; \\ 0 & \text{otherwise.} \end{cases}$$

If the four conditions (r1)–(r4) in Proposition 11.2.1. are satisfied, then output $\widehat{\delta}_k$. If at least one condition of the four is violated, go to Step 3. If $k = \min(m, n)$, then halt. Otherwise $k := k + 1$ and go to Step 1.

Step 3: Modify the matrix $A(x)$ and the potentials as described in (16)–(22), according to which of (r1)–(r4) is violated. Go to Step 1.

In the algorithm described above, we may start with an arbitrary set of admissible potentials. It is desirable, however, to adopt reasonably small integers as the initial potentials.

One of the natural choices is to put

$$\rho_i = \max_{j \in C} c_{ij} \quad \text{for } i \in R, \quad \gamma_j = 0 \quad \text{for } j \in C. \tag{25}$$

In this case, $\gamma_j = 0$ for $j \in C$ throughout the computation. Hence ρ_i for $i \in R$ always coincide with the row degrees. Since $J^\sharp = \emptyset$ and $\tilde{J} = C$, the four rank conditions in the test for tightness (see Proposition 11.2.1.) are reduced to a couple of conditions (r1) and (r2). This algorithm will be called the row-degree method, and is an extension of the well-known algorithm for $\delta_m(A)$ of an $m \times n$ row-full rank matrix $A(x)$ based on the concept of row properness (Wolovich [23]).

Another natural choice is, of course, the symmetric (transposed) version of the above, i.e.,

$$\rho_i = 0 \quad \text{for } i \in R, \quad \gamma_j = \max_{i \in R} c_{ij} \quad \text{for } j \in C. \tag{26}$$

This algorithm, however, does not behave in a symmetric manner to the row-degree method. This is because the symmetry between rows and columns is broken in the way of modification. In fact, any condition of the four may possibly be violated. Thus this algorithm is not the "column-degree method," and the primal-dual framework is not a simple extension of the Wolovich algorithm.

11.3 Experimental evaluation

11.3.1 Method

We make experiments on Sun SPARC Station 10 to evaluate the three combinatorial relaxation type algorithms, the primal-dual method (PD) with initial potentials given by (26), the row-degree method (RD) and the matching method (M). All of these are implemented in C language. For comparison we also realize the elimination method (E) in Maple (a computer algebraic system).

Our program restricts input matrices to Laurant polynomial matrices of which each entry is a polynomial in x and $1/x$.

We provide two types of sample problems, regular pencils (P) and band matrices (B). A regular pencil $xE - F$ used here is generated randomly in the following way:

$$E_{ij} = \begin{cases} g & \text{with probability } \frac{1-p}{2K} \quad \text{for } g = \pm 1, \pm 2, \ldots, \pm K; \\ 0 & \text{with probability } p, \end{cases}$$

$$F_{ij} = \begin{cases} h & \text{with probability } \frac{1-q}{2K} \quad \text{for } h = \pm 1, \pm 2, \ldots, \pm K; \\ 0 & \text{with probability } q. \end{cases}$$

We put K=1, p=0.0625 and q=0.5.

On the other hand, we provide band matrices in the following way. First we put

$$X = \mathrm{diag}\,(x; d), \quad d = (d_1, d_2, \ldots d_n),$$

where d_i is generated randomly as follows:

$$d_i = \ell \quad \text{with probability } \frac{1}{2L+1} \quad \text{for } \ell = 0, \pm 1, \pm 2, \ldots, \pm L.$$

With a lower bidiagonal matrix

$$P = \begin{pmatrix} 1 & & & \\ \frac{1}{x} & 1 & & \\ & \ddots & \ddots & \\ & & \frac{1}{x} & 1 \end{pmatrix}$$

and an upper bidiagonal matrix

$$Q = \begin{pmatrix} 1 & \frac{1}{x} & & \\ & \ddots & \ddots & \\ & & 1 & \frac{1}{x} \\ & & & 1 \end{pmatrix}$$

we construct a pentadiagonal (band) matrix $A(x) = P^2 X Q^2$. We put $L = 10$. Obviously, both P and Q are biproper, and hence the Smith-McMillan form at infinity of $A(x)$ is the same as that of X, which is easily obtained by sorting $(d_1, \ldots, d_n)$. Therefore, we know $\delta_k(A)$ in advance and we use this type of matrices as sample data to encertain that our program returns correct answers.

We measure CPU time and the number of modifications (for PD, RD, and M) to evaluate each algorithm.

11.3.2 Results

Before we report experimental results, we should note that the time complexity is $\mathrm{V}(rmn \max(m, n))$ if $(d_{\max} - d_{\min})$ is bounded by a constant. In fact, we have $(d_{\max} - d_{\min}) \leq 1$ with the pencils and $(d_{\max} - d_{\min}) \leq 2L + 2$ with the band matrices. Therefore the time complexity is $\mathrm{V}(rmn \max(m, n))$.

表 1 Statistics over 10 pencils of each size. CPU time in seconds and the number of modifications show the mean values of 10 iterations ($K = 1, p = 0.0625, q = 0.5$; Sun SPARC Station 10).

Algorithms		n				
		8	16	32	64	128
PD	CPU time (s)	6.67×10^{-3}	6.17×10^{-2}	6.72×10^{-1}	7.73	—
	Stand. Dev.	8.61×10^{-3}	8.05×10^{-3}	1.77×10^{-2}	1.20×10^{-1}	—
	# Modifications	0.3	0.9	1.0	0.6	—
RD	CPU time (s)	1.17×10^{-2}	5.67×10^{-2}	5.70×10^{-1}	6.91	9.87×10
	Stand. Dev.	8.05×10^{-3}	8.61×10^{-3}	2.19×10^{-2}	1.04×10^{-1}	5.33×10^{-1}
	# Modifications	0.7	1.1	0.8	0.7	0.1
M	CPU time (s)	1.17×10^{-2}	6.83×10^{-2}	6.62×10^{-1}	7.61	1.04×10^{2}
	Stand. Dev.	8.05×10^{-3}	9.46×10^{-3}	3.04×10^{-2}	1.41×10^{-1}	4.56×10^{-1}
	# Modifications	0.1	0.3	0.2	0.1	0.0
E	CPU time (s)	3.23	4.41×10	1.10×10^{3}	—	—
	Stand. Dev.	4.83×10^{-1}	4.94	2.11×10	—	—

Tables 1 and 2 show the results for the pencils and the band matrices, respectively. Algorithm (PD) failed for $n = 128$ because of a memory shortage. The experiments with algorithm (E) for large n were given up because it would have taken many days.

It is observed in our experiment that for $n \times n$ matrix A, approximately $\mathrm{V}(n^{3.4})$ time is needed by each of combinatorial relaxation algorithm, while the worst case time complexity is $\mathrm{V}(n^4)$ since $\operatorname{rank} A = n$ both in problem type (P) and in (B). On the other hand, the elimination method takes approximately $\mathrm{V}(n^4)$ time in our experiment, whereas the number of operations on rational functions is $\mathrm{V}(n^3)$.

表 2 Statistics over 10 band matrices of each size. CPU time in seconds and the number of modifications show the mean values of 10 iterations ($L = 10$; Sun SPARC Station 10).

Algorithms		n				
		8	16	32	64	128
PD	CPU time (s)	1.50×10^{-2}	1.03×10^{-1}	9.92×10^{-1}	1.19×10	—
	Stand. Dev.	5.27×10^{-3}	1.31×10^{-2}	3.54×10^{-2}	3.22×10^{-1}	—
	# Modifications	6.5	12.1	15.9	18.1	—
RD	CPU time (s)	1.50×10^{-2}	9.83×10^{-2}	1.01	1.11×10	1.41×10^{2}
	Stand. Dev.	5.27×10^{-3}	1.46×10^{-2}	4.92×10^{-2}	2.03×10^{-1}	7.38×10^{-1}
	# Modifications	5.5	11.9	15.7	18.1	18.9
M	CPU time (s)	1.50×10^{-2}	1.13×10^{-1}	9.37×10^{-1}	1.04×10	1.35×10^{2}
	Stand. Dev.	1.23×10^{-2}	1.89×10^{-2}	2.81×10^{-2}	2.57×10^{-1}	1.81
	# Modifications	3.6	6.8	8.5	10.2	11.6
E	CPU time (s)	2.92	4.52×10	7.65×10^{2}	1.28×10^{4}	—
	Stand. Dev.	1.12	4.13×10	6.94×10^{2}	1.27×10^{4}	—

In the above experiment, no distinction in computing time is recognized among algorithms (PD), (RD), and (M). Since algorithm (RD) lacks symmetry, it is natural that this algorithm takes longer time than the other algorithms for a matrix with a special structure. For example, let us consider

$$A(x) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & x & x^2 & \cdots & x^{n-1} \\ 1 & x^2 & x^4 & \cdots & x^{2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x^{n-1} & x^{2(n-1)} & \cdots & x^{(n-1)^2} \end{pmatrix}. \tag{1}$$

表 3 CPU time in seconds and the number of modifications for the matrix $A(x)$ of (1) with n=16.

Algorithms	CPU time (s)	# Modifications
PD	1.67	139
RD	2.03	156
M	8.33×10^{-2}	0

The results for the matrix with n=16 are shown on Table 3.

11.4 Conclusion

In this paper, we have proposed a primal-dual framework of combinatorial relaxation algorithms for computing the maximum degree of subdeterminants. Furthermore, we have realized combinatorial relaxation algorithms in computer program and made experiments. It turns out that combinatorial relaxation algorithms are effective for computing the maximum degree of subdeterminants and the index of nilpotency.

参考文献

[1]R. K. AHUJA, T. L. MAGNANTI, J. B. ORLIN, *Network Flows — Theory, Algorithms and Applications*, Prentice Hall, 1993.

[2]K. E. BRENAN, S. L. CAMPBELL, AND L. R. PETZOLD, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, 1989.

[3]R. A. BRUALDI AND H. J. RYSER, *Combinatorial Matrix Theory*, Cambridge University Press, 1991.

[4]P. BUJAKIEWICZ, *Maximum Weighted Matching for High Index Differential Algebraic Equations*, Doctor Thesis, Delft University of Technology, 1994.

[5]C. COMMAULT AND J.-M. DION, *Structure at infinity of linear multivariable systems: a geometric approach*, IEEE Trans. Automat. Control, AC-27 (1982), pp. 693–696.

[6]C. COMMAULT, J.-M. DION AND A. PEREZ, *Disturbance rejection for structured systems*, IEEE Trans. Automat. Control, AC-36 (1991), pp. 884–887.

[7]I. DUFF AND C. W. GEAR, *Computing the structural index*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 594–603.

[8]R. GANI AND I. T. CAMERON, *Modelling for dynamic simulation of chemical processes: the index problem*, Chem. Engin. Sci., 47 (1992), pp. 1311–1315.

[9]C. W. GEAR, *Differential-algebraic equation index transformations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 39–47.

[10]C. W. GEAR, *Differential algebraic equations, indices, and integral algebraic equations*, SIAM J. Numer. Anal., 27 (1990), pp. 1527–1534.

[11]E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II*, Springer-Verlag, 1991.

[12]M. L. J. HAUTUS, *The formal Laplace transform for smooth linear systems*, in Mathematical Systems Theory (eds. G. Marchesini and S. K. Mitter), Lecture Notes in Economics and Mathematical Systems, 131, Springer-Verlag, 1976, pp. 29–47.

[13]K. MUROTA, *Systems Analysis by Graphs and Matroids — Structural Solvability and Controllability*, Algorithms and Combinatorics 3, Springer-Verlag, 1987.

[14]K. MUROTA, *Computing Puiseux-series solutions to determinantal equations via combinatorial relaxation*, SIAM J. Comput., 19 (1990), pp. 1132–1161.

[15]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 Engin. Comm. Comput., to appear.

[16]K. MUROTA, *Computing the degree of determinants via combinatorial relaxation*, SIAM J. Comput., 24 (1995), No. 4, to appear.

[17]K. MUROTA AND J. W. VAN DER WOUDE, *Structure at infinity of structured descriptor systems and its applications*, SIAM J. Control Optim., 29 (1991), pp. 878–894.

[18]M. NEWMAN, *Integral Matrices*, Academic Press, 1972.

[19]C. C. PANTELIDES, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 213–231.

[20]C. H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, 1982.

[21]J. UNGAR, A. KRÖNER AND W. MARQUARDT, *Structural analysis of differential-algebraic equation systems: Theory and application*, Comput. Chem. Engin., to appear.

[22]G. C. VERGHESE AND T. KAILATH, *Rational matrix structure*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 434–439.

[23]W. A. WOLOVICH, *Linear Multivariable Systems*, Springer-Verlag, 1974.

[24]J. W. VAN DER WOUDE, *On the structure at infinity of a structured system*, Linear Algebra Appl., 148 (1991), pp. 145–169.