RIMS-1721

Computing the Maximum Degree of Minors in Mixed Polynomial Matrices via Combinatorial Relaxation

By

Satoru IWATA and Mizuyo TAKAMATSU

May 2011

京都大学　数理解析研究所

RESEARCH INSTITUTE FOR MATHEMATICAL SCIENCES

KYOTO UNIVERSITY, Kyoto, Japan

Computing the Maximum Degree of Minors in Mixed Polynomial Matrices via Combinatorial Relaxation*

Satoru Iwata† Mizuyo Takamatsu‡

May 6, 2011

Abstract

Mixed polynomial matrices are polynomial matrices with two kinds of nonzero coefficients: fixed constants that account for conservation laws and independent parameters that represent physical characteristics. The computation of their maximum degrees of minors is known to be reducible to valuated independent assignment problems, which can be solved by polynomial numbers of additions, subtractions, and multiplications of rational functions. However, these arithmetic operations on rational functions are much more expensive than those on constants.

In this paper, we present a new algorithm of combinatorial relaxation type. The algorithm finds a combinatorial estimate of the maximum degree by solving a weighted bipartite matching problem, and checks if the estimate is equal to the true value by solving independent matching problems. The algorithm mainly relies on fast combinatorial algorithms and performs numerical computation only when necessary. In addition, it requires no arithmetic operations on rational functions. As a byproduct, this method yields a new algorithm for solving a linear valuated independent assignment problem.

1 Introduction

Let $A(s) = (A_{ij}(s))$ be a rational function matrix with $A_{ij}(s)$ being a rational function in s. The maximum degree $\delta_k(A)$ of minors of order k is defined by

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

where $\deg f$ denotes the degree of a rational function $f(s)$, and $A[I, J]$ denotes the submatrix with row set I and column set J. This $\delta_k(A)$ determines the Smith-McMillan form at infinity of a rational function matrix [27], which is used in decoupling and disturbance rejection of linear

*A preliminary version of this paper is to appear in Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization.

†Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606-8502, Japan. Supported by a Grant-in-Aid for Scientific Research from the Japan Society for Promotion of Science. E-mail: iwata@kurims.kyoto-u.ac.jp

‡Department of Information and System Engineering, Chuo University, Tokyo 112-8551, Japan. Supported by a Grant-in-Aid for Scientific Research from the Japan Society for Promotion of Science. E-mail: takamatsu@ise.chuo-u.ac.jp

time-invariant systems, and the Kronecker canonical form of a matrix pencil [4, 7, 26], which is used in analysis of linear DAEs with constant coefficients. Thus, the computation of $\delta_k(A)$ is a fundamental and important procedure in dynamical systems analysis.

The notion of mixed polynomial matrices [15, 22] was introduced as a mathematical tool for faithful description of dynamical systems such as electric circuits, mechanical systems, and chemical plants. A mixed polynomial matrix is a polynomial matrix that consists of two kinds of coefficients as follows.

Accurate Numbers (Fixed Constants) Numbers that account for conservation laws are precise in values. These numbers should be treated numerically.

Inaccurate Numbers (Independent Parameters) Numbers that represent physical characteristics are not precise in values. These numbers should be treated combinatorially as nonzero parameters without reference to their nominal values. Since each such nonzero entry often comes from a single physical device, the parameters are assumed to be independent.

For example, physical characteristics in engineering systems are not precise in values because of measurement noise, while exact numbers do arise in conservation laws such as Kirchhoff's conservation laws in electric circuits, or the law of conservation of mass, energy, or momentum and the principle of action and reaction in mechanical systems. Thus, it is natural to distinguish inaccurate numbers from accurate numbers in the description of dynamical systems.

In [21], Murota reduces the computation of $\delta_k(A)$ for a mixed polynomial matrix $A(s)$ to solving a *valuated independent assignment problem*, for which he presents in [19, 20] algorithms that perform polynomial numbers of additions, subtractions, and multiplications of rational functions. However, these arithmetic operations on rational functions are much more expensive than those on constants.

In this paper, we present an algorithm for computing $\delta_k(A)$, based on the framework of "combinatorial relaxation." The outline of the algorithm is as follows.

Phase 1 Construct a relaxed problem by discarding numerical information and extracting zero/nonzero structure in $A(s)$. The solution is regarded as an estimate of $\delta_k(A)$.

Phase 2 Check whether the obtained estimate is equal to the true value of $\delta_k(A)$, or not. If it is, return the estimate and halt.

Phase 3 Modify the relaxation so that the invalid solution is eliminated, and find a solution to the modified relaxed problem. Then go back to Phase 2.

In this algorithm, we solve a weighted bipartite matching problem in Phase 1 and independent matching problems in Phase 2. We remark that our algorithm does not need symbolic operations on rational functions.

This framework of combinatorial relaxation algorithm is somewhat analogous to the idea of relaxation and cutting plane in integer programming. In contrast to integer programming, where hard combinatorial problems are relaxed to linear programs, here we relax a linear algebraic problem to an efficiently solvable combinatorial problem. Then the algorithm checks the validity of the obtained solution and modifies the relaxation if necessary just like adding a cutting plane. This is where the name "combinatorial relaxation" comes from. See [22, §7.1] for more details.

We now summarize previous works based on the framework of combinatorial relaxation. The combinatorial relaxation approach was invented by Murota [16] for computing the Newton polygon of the Puiseux series solutions to determinantal equations. This approach was further applied to the computation of the degree of $\det A(s)$ in [17] and $\delta_k(A)$ in [9, 10, 11, 18] for a polynomial matrix $A(s)$. In computational efficiency of the algorithms, it is crucial to solve the relaxed problem efficiently and not to invoke the modification of the relaxation many times. The result in [11] shows practical efficiency of the combinatorial relaxation through computational experiments.

Let us have a closer look at the algorithms [9, 10, 11, 18] for $\delta_k(A)$. In [18], Murota presented a combinatorial relaxation algorithm for general rational function matrices using biproper equivalence transformations in the modification of the relaxed problem. The primal-dual version was presented in [11]. These algorithms need to transform a polynomial matrix by row/column operations, which possibly increase the number of terms in some entries. To avoid this phenomenon in the case of matrix pencils, Iwata [9] presented another combinatorial relaxation algorithm, which uses only strict equivalence transformations.

Another approach presented in [10] uses a mixed polynomial matrix as a combinatorial relaxation. Given a mixed polynomial matrix, one can compute the maximum degree of minors by solving a valuated independent assignment problem. When specific values are assigned into the independent parameters, the maximum degree of minors may change. The combinatorial relaxation algorithm in [10] computes the resulting exact value.

In this paper, we extend the combinatorial relaxation framework to mixed polynomial matrices. In contrast to the previous work [10], our goal is to compute the maximum degree of minors in a given mixed polynomial matrix efficiently. Our combinatorial relaxation is the assignment problem obtained by replacing accurate numbers by independent parameters. Our algorithm adopts a different way of matrix modification from the previous algorithms [10, 11, 18], which enables us to evaluate the complexity by the number of basic arithmetic operations. For an $m \times n$ mixed polynomial matrix with $m \leq n$, the algorithm runs in $O(m^{\omega+1} n d_{\max}^2)$ time, where ω is the matrix multiplication exponent and $d_{\max}$ is the maximum degree of an entry.

We compare this time complexity with that of the previous algorithm of Murota [21] based on the valuated independent assignment. The bottleneck in that algorithm is to transform an $m \times n$ polynomial matrix into an upper triangular matrix in each iteration. This can be done in $O\tilde{~}(m^{\omega} n d_{\max})$ time, where $O\tilde{~}$ indicates that we ignore $\log(m d_{\max})$ factors, by using Bareiss' fraction-free Gaussian elimination approach [1, 2, 25] and an $O(d \log d \log \log d)$ time algorithm in [3] for multiplying polynomials of degree d. Since the number of iterations is k, Murota's algorithm runs in $O\tilde{~}(k m^{\omega} n d_{\max})$ time. Thus the worst-case complexity of our algorithm is comparable to that of the previous one.

However, our combinatorial relaxation algorithm terminates without invoking the modification of the relaxation unless there is an unlucky numerical cancellation. Consequently, in most cases, it runs in $O(m^{\omega} n d_{\max})$ time, which is much faster than the previous algorithm.

One application of our combinatorial relaxation algorithm is to compute the Kronecker canonical form of a *mixed matrix pencil*, which is a mixed polynomial matrix with $d_{\max} = 1$. For an $n \times n$ regular mixed matrix pencil, our algorithm enables us to compute the Kronecker canonical form in $O(n^{\omega+2})$ time. This time complexity is better than the previous algorithm given in [12], which makes use of another characterization based on expanded matrices.

Another application is to compute the optimal value of a linear valuated independent assignment problem. The optimal value of this problem coincides with the degree of the determinant of an associated mixed polynomial matrix. Thus we can make use of our combinatorial relaxation algorithm. This means that we find the optimal value of a linear valuated independent assignment problem by solving a sequence of independent matching problems.

Combinatorial relaxation approach exploits the combinatorial structures of polynomial matrices and exhibits a connection between combinatorial optimization and matrix computation. While recent works of Mucha and Sankowski [13, 14], Sankowski [24], and Harvey [8] utilize matrix computation to solve matching problems, this paper adopts the opposite direction, that is, we utilize matching problems for matrix computation.

The organization of this paper is as follows. Section 2 is devoted to preliminaries on rational function matrices, the independent matching problem, and mixed matrix theory. We present a combinatorial relaxation algorithm in Section 3, and analyze its complexity in Section 4. In Section 5, we apply the combinatorial relaxation approach to the linear valuated independent assignment problem.

2 Preliminaries

We provide preliminaries on rational functions (Section 2.1), independent matching problems (Section 2.2), and mixed polynomial matrices (Section 2.3). Then we explain how to reduce the computation of the rank of a layered mixed matrix to an independent matching problem in Section 2.4.

2.1 Rational Function Matrices

We denote the degree of a polynomial $g(s)$ by $\deg g$, where $\deg 0 = -\infty$ by convention. For a rational function $f(s) = g(s)/h(s)$ with polynomials $g(s)$ and $h(s)$, its degree is defined by $\deg f = \deg g - \deg h$. A rational function $f(s)$ is called *proper* if $\deg f \leq 0$, and *strictly proper* if $\deg f < 0$. We call a rational function matrix (*strictly*) *proper* if its entries are (strictly) proper rational functions. A square proper rational function matrix is called *biproper* if it is invertible and its inverse is a proper rational function matrix. A proper rational function matrix is biproper if and only if its determinant is a nonzero constant. It is known that $\delta_k(A)$ is invariant under biproper equivalence transformations, i.e.,

$$\delta_k(A) = \delta_k(\tilde{A}) \quad (k = 1, \ldots, \operatorname{rank} A(s))$$

if $\tilde{A}(s) = U(s)A(s)V(s)$ with biproper matrices $U(s)$ and $V(s)$.

A rational function matrix $Z(s)$ is called a *Laurent polynomial matrix* if $s^K Z(s)$ is a polynomial matrix for some integer K. In our algorithm, we make use of biproper Laurent polynomial matrices in the phase of matrix modification.

2.2 Independent Matching Problem

A *matroid* is a pair $\mathbf{M} = (V, \mathcal{I})$ of a finite set V and a collection $\mathcal{I}$ of subsets of V such that

(I-1) $\emptyset \in \mathcal{I}$,

(I-2) $I \subseteq J \in \mathcal{I} \Rightarrow I \in \mathcal{I}$,

(I-3) $I, J \in \mathcal{I}, |I| < |J| \Rightarrow I \cup \{v\} \in \mathcal{I}$ for some $v \in J \setminus I$.

The set V is called the *ground set*, $I \in \mathcal{I}$ is an *independent set*, and $\mathcal{I}$ is the *family of independent sets*. The following problem is an extension of the matching problem.

> **[Independent Matching Problem (IMP)]**
> Given a bipartite graph $G = (V^+, V^-; E)$ with vertex sets V^+, V^- and edge set E, and a pair of matroids $\mathbf{M}^+ = (V^+, \mathcal{I}^+)$ and $\mathbf{M}^- = (V^-, \mathcal{I}^-)$, find a matching $M \subseteq E$ that maximizes $|M|$ subject to
>
> $$\partial^+ M \in \mathcal{I}^+, \quad \partial^- M \in \mathcal{I}^-, \tag{1}$$
>
> where $\partial^+ M$ and $\partial^- M$ denote the set of vertices in V^+ and V^- incident to M, respectively.

A matching $M \subseteq E$ satisfying (1) is called an *independent matching*.

2.3 Mixed Matrices and Mixed Polynomial Matrices

A *generic matrix* is a matrix in which each nonzero entry is an independent parameter. A matrix A is called a *mixed matrix* if A is given by $A = Q + T$ with a constant matrix Q and a generic matrix T. A *layered mixed matrix* (or an *LM-matrix* for short) is defined to be a mixed matrix such that Q and T have disjoint nonzero rows. An LM-matrix A is expressed by $A = \binom{Q}{T}$.

A polynomial matrix $A(s)$ is called a *mixed polynomial matrix* if $A(s)$ is given by $A(s) = Q(s) + T(s)$ with a pair of polynomial matrices $Q(s) = \sum_{k=0}^{N} s^k Q_k$ and $T(s) = \sum_{k=0}^{N} s^k T_k$ that satisfy the following two conditions.

(MP-Q) Q_k $(k = 0, 1, \ldots, N)$ are constant matrices.

(MP-T) T_k $(k = 0, 1, \ldots, N)$ are generic matrices.

A *layered mixed polynomial matrix* (or an *LM-polynomial matrix* for short) is defined to be a mixed polynomial matrix such that $Q(s)$ and $T(s)$ have disjoint nonzero rows. An LM-polynomial matrix $A(s)$ is expressed by $A(s) = \binom{Q(s)}{T(s)}$.

The matrix $A(s) = \binom{Q(s)}{T(s)}$ is called an *LM-Laurent polynomial matrix* if $s^K A(s)$ is an LM-polynomial matrix for some integer K. We denote the row set and the column set of $A(s)$ by R and C, and the row sets of $Q(s)$ and $T(s)$ by R_Q and R_T. We also set $m_Q = |R_Q|$, $m_T = |R_T|$, and $n = |C|$ for convenience. The (i, j) entry of $Q(s)$ and $T(s)$ is denoted by $Q_{ij}(s)$ and $T_{ij}(s)$, respectively. We use these notations throughout this paper for LM-Laurent polynomial matrices as well as LM-matrices. For an LM-Laurent polynomial matrix $A(s) = \binom{Q(s)}{T(s)}$, let us define

$$\delta_k^{\mathrm{LM}}(A) = \{\deg \det A[R_Q \cup I, J] \mid I \subseteq R_T, J \subseteq C, |I| = k, |J| = m_Q + k\},$$

where $0 \leq k \leq \min(m_T, n - m_Q)$. Note that $\delta_k^{\mathrm{LM}}(A)$ designates the maximum degree of minors of order $m_Q + k$ with row set containing R_Q.

We denote an $m \times m$ diagonal matrix with the (i, i) entry being a_i by $\mathrm{diag}[a_1, \ldots, a_m]$. Let $\tilde{A}(s) = \tilde{Q}(s) + \tilde{T}(s)$ be an $m \times n$ mixed polynomial matrix with row set R and column set C. We construct a $(2m) \times (m + n)$ LM-polynomial matrix

$$A(s) = \begin{pmatrix} \mathrm{diag}[s^{d_1}, \ldots, s^{d_m}] & \tilde{Q}(s) \\ -\mathrm{diag}[t_1 s^{d_1}, \ldots, t_m s^{d_m}] & \tilde{T}(s) \end{pmatrix}, \tag{2}$$

where $t_i\,(i = 1, \ldots, m)$ are independent parameters and $d_i = \max_{j \in C} \deg \tilde{Q}_{ij}(s)$ for $i \in R$. A mixed polynomial matrix $\tilde{A}(s)$ and its associated LM-polynomial matrix $A(s)$ have the following relation, which implies that the value of $\delta_k(\tilde{A})$ is obtained from $\delta_k^{\mathrm{LM}}(A)$.

Lemma 2.1 ([22, Lemma 6.2.6])**.** Let $\tilde{A}(s)$ be an $m \times n$ mixed polynomial matrix and $A(s)$ the associated LM-polynomial matrix defined by (2). Then it holds that

$$\delta_k(\tilde{A}) = \delta_k^{\mathrm{LM}}(A) - \sum_{i=1}^{m} d_i.$$

2.4 Rank of LM-matrices

The computation of the rank of an LM-matrix $A = \binom{Q}{T}$ is reduced to solving an independent matching problem [23] as follows. See [22, §4.2.4] for details.

Let $C_Q = \{j_Q \mid j \in C\}$ be a copy of the column set C of A. Consider a bipartite graph $G = (V^+, V^-; E_T \cup E_Q)$ with $V^+ = R_T \cup C_Q$, $V^- = C$,

$$E_T = \{(i, j) \mid i \in R_T, j \in C, T_{ij} \neq 0\} \quad \text{and} \quad E_Q = \{(j_Q, j) \mid j \in C\}.$$

Let $\mathbf{M}^+ = (V^+, \mathcal{I}^+)$ be a matroid defined by

$$\mathcal{I}^+ = \{I^+ \subseteq V^+ \mid \mathrm{rank}\, Q[R_Q, I^+ \cap C_Q] = |I^+ \cap C_Q|\},$$

and $\mathbf{M}^-$ be a free matroid. Then $\mathrm{rank}\, A$ has the following property.

Theorem 2.2 ([22, Theorem 4.2.18])**.** Let A be an LM-matrix. Then the rank of A is equal to the maximum size of an independent matching in the problem defined above, i.e.,

$$\mathrm{rank}\, A = \max\{|M| \mid M : \text{independent matching}\}.$$

We describe the algorithm [22] for computing $\mathrm{rank}\, A$ as follows. Let us denote the reorientation of $a \in E_T \cup E_Q$ by a°. With reference to G and an independent matching M, we construct an auxiliary graph $G_M = (\tilde{V}, \tilde{E})$ with $\tilde{V} = V^+ \cup V^-$ and $\tilde{E} = E_T \cup E_Q \cup E^+ \cup M^\circ$, where

$$
\begin{aligned}
&E^+ = \{(i,j) \mid i \in \partial^+ M \cap C_Q, j \in C_Q \setminus \partial^+ M, \partial^+ M \setminus \{i\} \cup \{j\} \in \mathcal{I}^+\}, \\
&M^\circ = \{a^\circ \mid a \in M\}.
\end{aligned}
$$

Let I_0 be any subset of C satisfying $\operatorname{rank} Q[R_Q, I_0] = \operatorname{rank} Q$. We determine an initial independent matching by using I_0.

Algorithm for the Rank of an LM-matrix

Step 1 $M \leftarrow \{(j_Q, j) \mid j \in I_0\}$. Then we transform Q by row operations so that $Q[R_Q, \partial^+ M \cap C_Q]$ is in the form of $\binom{D}{O}$ with D being a diagonal matrix.

Step 2 If there exists in $G_M = (\tilde{V}, \tilde{E})$ a directed path from $S^+ = R_T \setminus \partial^+ M$ to $S^- = C \setminus \partial^- M$, then go to Step 3. Otherwise output $\operatorname{rank} A = |M|$ and halt.

Step 3 Let P be a shortest path from S^+ to S^-. $M \leftarrow (M \setminus \{a \in M \mid a^\circ \in P \cap M^\circ\}) \cup (P \cap E_T) \cup (P \cap E_Q)$. Then we transform Q by row operations so that $Q[R_Q, \partial^+ M \cap C_Q]$ is in the form of $\binom{D}{O}$ with D being a diagonal matrix. Go to Step 2.

Let $\tilde{A} = \binom{\tilde{Q}}{\tilde{T}}$ be an LM-matrix obtained at the termination of this algorithm. Then we have

$$
\begin{pmatrix} \tilde{Q} \\ \tilde{T} \end{pmatrix} = \begin{pmatrix} U & O \\ O & I \end{pmatrix} \begin{pmatrix} Q \\ T \end{pmatrix} \tag{3}
$$

for some nonsingular constant matrix U, because Q is transformed into $\tilde{Q}$ by row operations in Steps 1 and 3.

The structure of an LM-matrix $\tilde{A} = (\tilde{A}_{ij})$ is represented by a bipartite graph $G(\tilde{A}) = (R_Q \cup R_T, C; E(\tilde{A}))$ with $E(\tilde{A}) = \{(i,j) \mid i \in R_Q \cup R_T, j \in C, \tilde{A}_{ij} \neq 0\}$. The maximum size of a matching in $G(\tilde{A})$ is called the *term-rank* of $\tilde{A}$, denoted by $\operatorname{term-rank} \tilde{A}$. Now $\tilde{A}$ has the following property, which forms the basis of the algorithm to compute the combinatorial canonical form (CCF) of an LM-matrix [22, §4.4].

Lemma 2.3. Let $\tilde{A}$ be an LM-matrix obtained at the termination of Algorithm for the Rank of an LM-matrix. Then we have $\operatorname{rank} \tilde{A} = \operatorname{term-rank} \tilde{A}$.

Proof. For $\tilde{A} = \binom{\tilde{Q}}{\tilde{T}}$, we denote the row sets of $\tilde{Q}$ and $\tilde{T}$ by R_Q and R_T, and the column set of $\tilde{A}$ by C. We may assume that $\tilde{A}$ is of full row rank, because the rows consisting only of zeros can be ignored. Let W be the set of vertices reachable from S^+ in G_M at the termination of the algorithm. Since there is no directed path from S^+ to S^-, it holds that $S^- \subseteq C \setminus W$. Hence $C \cap W \subseteq \partial^- M$ follows from $S^- = C \setminus \partial^- M$.

We put $B = \{j \in C \mid j_Q \in \partial^+ M \cap C_Q\}$. Then $\tilde{A}[R_Q, B]$ is expressed as $\tilde{A}[R_Q, B] = \begin{pmatrix} D_1 & O \\ O & D_2 \end{pmatrix}$, where D_1 and D_2 are diagonal matrices with column sets $B_1 \subseteq C \setminus W$ and $B_2 \subseteq C \cap W$. The row sets of D_1 and D_2 is denoted by R_1 and R_2. If $\tilde{A}[R_2, (C \setminus W) \setminus B_1] \neq O$,

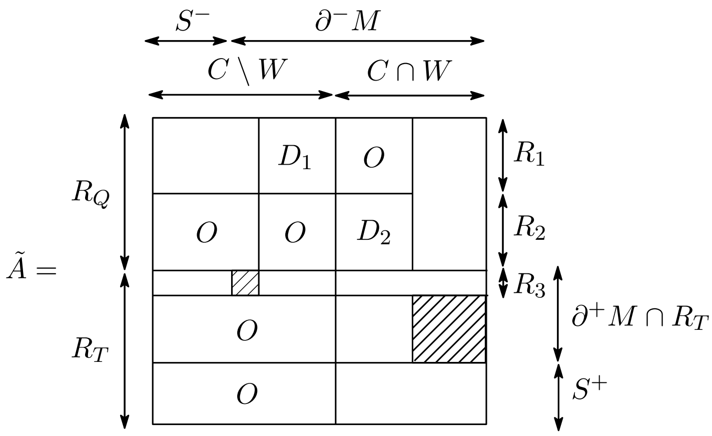

Figure 1: An LM-matrix obtained at the termination of Algorithm for the Rank of an LM-matrix, where shaded squares represent nonsingular matrices.

there exists an edge (i_Q, j_Q) in E^+ with $i \in B_2$ and $j \in (C \setminus W) \setminus B_1$. Then j is reachable from S^+ through $i \in C \cap W$, i_Q, and j_Q, which contradicts the definition of W. Hence it holds that $\tilde{A}[R_2, (C \setminus W) \setminus B_1] = O$.

Let us define $R_3 = \{i \in \partial^+ M \cap R_T \mid \exists j \in (C \setminus W) \setminus B_1 \text{ such that } (i,j) \in M\}$. Then we have

$$\operatorname{rank} \tilde{A} = |M| = |R_1| + |R_2| + |R_3| + |(\partial^+ M \cap R_T) \setminus R_3|.$$

By the definition of W, it holds that $\tilde{A}[S^+, C \setminus W] = O$ and $\tilde{A}[(\partial^+ M \cap R_T) \setminus R_3, C \setminus W] = O$. Thus, $\tilde{A}$ is in the form shown in Figure 1. Consider a bipartite $G(\tilde{A}) = (R_Q \cup R_T, C; E(\tilde{A}))$. Since $(R_1 \cup R_3, C \cap W)$ is a cover in $G(\tilde{A})$, it holds that

$$\text{term-rank}\, \tilde{A} \leq |R_1 \cup R_3| + |C \cap W| = |R_1| + |R_3| + |R_2| + |(\partial^+ M \cap R_T) \setminus R_3|,$$

where the first step is due to König-Egerváry theorem. Thus we obtain

$$\operatorname{rank} \tilde{A} \leq \text{term-rank}\, \tilde{A} \leq \operatorname{rank} \tilde{A},$$

which implies $\operatorname{rank} \tilde{A} = \text{term-rank}\, \tilde{A}$. □

3 Combinatorial Relaxation Algorithm

In this section, we present a combinatorial relaxation algorithm to compute

$$\delta_k^{\mathrm{LM}}(A) = \max_{I,J} \{\deg \det A[R_Q \cup I, J] \mid I \subseteq R_T, J \subseteq C, |I| = k, |J| = m_Q + k\}$$

for an LM-polynomial matrix $A(s) = \binom{Q(s)}{T(s)}$. We assume that $Q(s)$ is of full row rank. Note that this assumption is valid for the associated LM-polynomial matrix defined by (2). Since an LM-polynomial matrix is transformed into an LM-Laurent polynomial matrix in the phase of matrix modification, we hereafter deal with an LM-Laurent polynomial matrix.

We describe the outline of the proposed algorithm in Section 3.1. Sections 3.2–3.5 are devoted to the details of each phase in the algorithm.

3.1 Combinatorial Relaxation

Let us construct a bipartite graph $G(A) = (R_Q \cup R_T, C; E(A))$ with $E(A) = \{(i,j) \mid i \in R_Q \cup R_T, j \in C, A_{ij}(s) \neq 0\}$. The weight $c(e)$ of an edge $e = (i,j)$ is given by $c(e) = c_{ij} = \deg A_{ij}(s)$. We remark that $c(e)$ is integer for any $e \in E(A)$ if $A(s)$ is an LM-Laurent polynomial matrix. The maximum weight of a matching in $G(A)$, denoted by $\hat{\delta}_k^{\mathrm{LM}}(A)$, is an upper bound on $\delta_k^{\mathrm{LM}}(A)$. We adopt $\hat{\delta}_k^{\mathrm{LM}}(A)$ as an estimate of $\delta_k^{\mathrm{LM}}(A)$.

Consider the following linear program ($\mathrm{PLP}(A,k)$):

$$
\begin{array}{rl}
\text{maximize} & \displaystyle\sum_{e \in E(A)} c(e)\xi(e) \\
\text{subject to} & \displaystyle\sum_{\partial e \ni i} \xi(e) = 1 \quad (\forall i \in R_Q), \\
& \displaystyle\sum_{\partial e \ni i} \xi(e) \leq 1 \quad (\forall i \in R_T), \\
& \displaystyle\sum_{\partial e \ni j} \xi(e) \leq 1 \quad (\forall j \in C), \\
& \displaystyle\sum_{e \in E(A)} \xi(e) = m_Q + k, \\
& \xi(e) \geq 0 \quad (\forall e \in E(A)),
\end{array}
$$

where ∂e denotes the set of vertices incident to e.

The first constraint represents that M must satisfy $\partial M \supseteq R_Q$, where ∂M denotes the vertices incident to edges in M. By the total unimodularity of the coefficient matrix, $\mathrm{PLP}(A,k)$ has an integral optimal solution with $\xi(e) \in \{0,1\}$ for any $e \in E(A)$. This optimal solution corresponds to the maximum weight matching M in $G(A)$, and its optimal value $c(M) = \sum_{e \in M} c(e)$ is equal to $\hat{\delta}_k^{\mathrm{LM}}(A)$. The dual program ($\mathrm{DLP}(A,k)$) is expressed as follows:

$$
\begin{array}{rl}
\text{minimize} & \displaystyle\sum_{i \in R} p_i + \sum_{j \in C} q_j + (m_Q + k)t \\
\text{subject to} & p_i + q_j + t \geq c(e) \quad (\forall e = (i,j) \in E(A)), \\
& p_i \geq 0 \quad (\forall i \in R_T), \\
& q_j \geq 0 \quad (\forall j \in C).
\end{array}
$$

Then $\mathrm{DLP}(A,k)$ has an integral optimal solution, because the coefficient matrix is totally unimodular and $c(e)$ is integer for any $e \in E(A)$.

The outline of the combinatorial relaxation algorithm to compute $\delta_k^{\mathrm{LM}}(A)$ is summarized as follows.

Outline of Algorithm for Computing $\delta_k^{\mathrm{LM}}(A)$

Phase 1 Find a maximum weight matching M such that $\partial M \supseteq R_Q$ and $|M| = m_Q + k$ in $G(A)$. Then the maximum weight $\hat{\delta}_k^{\mathrm{LM}}(A)$ is regarded as an estimate of $\delta_k^{\mathrm{LM}}(A)$. Construct an optimal solution (p,q,t) of $\mathrm{DLP}(A,k)$ from M.

Phase 2 Test whether $\hat{\delta}_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A)$ or not by using (p, q, t). If equality holds, return $\hat{\delta}_k^{\mathrm{LM}}(A)$ and halt.

Phase 3 Modify $A(s)$ to another matrix $\tilde{A}(s)$ such that $\hat{\delta}_k^{\mathrm{LM}}(\tilde{A}) \leq \hat{\delta}_k^{\mathrm{LM}}(A) - 1$ and $\delta_k^{\mathrm{LM}}(\tilde{A}) = \delta_k^{\mathrm{LM}}(A)$. Construct an optimal solution $(\tilde{p}, \tilde{q}, \tilde{t})$ of $\mathrm{DLP}(\tilde{A}, k)$ from (p, q, t), and go back to Phase 2.

In Phase 1, we find a maximum weight matching M by using an efficient combinatorial algorithm. An optimal solution (p, q, t) can be obtained by using shortest path distance in an associated auxiliary graph. In Phase 2, we check whether the upper estimate $\hat{\delta}_k^{\mathrm{LM}}(A)$ coincides with $\delta_k^{\mathrm{LM}}(A)$ by computing the ranks of LM-matrices. If it does not, we transform $A(s)$ into $\tilde{A}(s)$ so that the upper estimate decreases in Phase 3. We repeat this procedure until the upper estimate coincides with $\delta_k^{\mathrm{LM}}(A)$. Since the upper estimate decreases at each step, starting with $\hat{\delta}_k^{\mathrm{LM}}(A)$ in the initial step, the number of iterations is at most $\hat{\delta}_k^{\mathrm{LM}}(A)$. The procedure in each phase is explained in detail below.

3.2 Construction of an Initial Dual Optimal Solution

For the LM-polynomial matrix $A(s)$ defined by (2), $G(A) = (R_Q \cup R_T, C; E(A))$ has a matching $M_0 = \{(i, i) \mid i \in R_Q\}$, which corresponds to $s^{d_1}, \ldots, s^{d_m}$. By applying augmenting path type algorithms to M_0, we find a maximum weight matching M with $\partial M \supseteq R_Q$ and $|M| = m_Q + k$ in $G(A)$. Then, we construct an optimal solution (p, q, t) of $\mathrm{DLP}(A, k)$ from M as follows.

Consider an auxiliary graph $\check{G}_M = (\check{V}, \check{E})$ with

$$\check{V} = R_Q \cup R_T \cup C \cup \{u^+\} \cup \{u^-\} \quad \text{and} \quad \check{E} = E_{\mathrm{c}} \cup M^\circ \cup W^+ \cup W^-,$$

where u^+ and u^- are new vertices and

$$\begin{aligned} E_{\mathrm{c}} &= \{(i, j) \mid (i, j) \in E(A)\} \quad \text{(copy of } E(A)), \\ M^\circ &= \{a^\circ \mid a \in M\}, \\ W^+ &= \{(u^+, i) \mid i \in R_T \setminus \partial M\}, \\ W^- &= \{(j, u^-) \mid j \in C \setminus \partial M\} \cup \{(u^-, j) \mid j \in C\}. \end{aligned}$$

We define the arc length $\gamma : \check{E} \to \mathbb{Z}$ by

$$\gamma(a) = \begin{cases} -c(a) & (a \in E_{\mathrm{c}}) \\ c(a^\circ) & (a \in M^\circ) \\ 0 & (a \in W^+ \cup W^-) \end{cases}.$$

Let $\varphi(i)$ be a shortest distance from u^+ to $i \in \check{V}$ with respect to the arc length γ in $\check{G}_M$. If there exists no path from u^+ to i, then we put $\varphi(i) = \infty$. We define

$$p_i = \varphi(i) \ (\forall i \in R_Q \cup R_T), \quad q_j = \varphi(u^-) - \varphi(j) \ (\forall j \in C), \quad t = -\varphi(u^-). \tag{4}$$

Then (p, q, t) is an optimal dual solution of $\mathrm{DLP}(A, k)$ as follows.

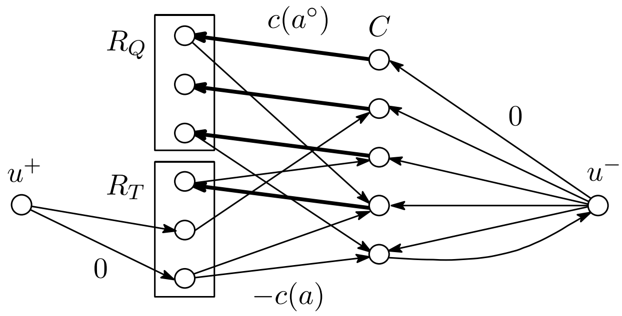

Figure 2: An auxiliary graph $\check{G} = (\check{V}, \check{E})$, where heavy lines show edges in a matching.

Lemma 3.1. Let M be a maximum weight matching with $\partial M \supseteq R_Q$ and $|M| = m_Q + k$ in $G(A)$. We define (p, q, t) by the auxiliary graph $\check{G}_M$ and (4). Then (p, q, t) is an optimal dual solution of $\mathrm{DLP}(A, k)$. Moreover, if a weight function c is integer valued, then (p, q, t) is an integral solution.

Proof. Let us assume that there exists a shortest path P from u^+ to $i \in R_T$ with negative distance. Then, the matching $\hat{M} = M \setminus \{a \in M \mid a^\circ \in P \cap M^\circ\} \cup (P \cap E_{\mathrm{c}})$ satisfies $c(\hat{M}) > c(M)$, which contradicts the optimality of M. Hence we have $\varphi(i) \geq 0$ for each $i \in R_T$. This implies the second constraint in $\mathrm{DLP}(A, k)$. Moreover, it holds that

$$\varphi(i) = 0 \quad (\forall i \in R_T \setminus \partial M), \tag{5}$$

because there exists an arc (u^+, i) for each $i \in R_T \setminus \partial M$.

Since there exists an arc (u^-, j) for each $j \in C$, we have $\varphi(j) \leq \varphi(u^-)$, which implies the third constraint in $\mathrm{DLP}(A, k)$. Similarly, we obtain $\varphi(j) \leq \varphi(i) - c(e)$ for each $e = (i, j) \in E(A)$. Hence the first constraint in $\mathrm{DLP}(A, k)$ holds by (4).

Finally, we check the optimality of (p, q, t). There exist both arcs (i, j) and (j, i) for any (i, j) satisfying $i = u^-$ and $j \in C \setminus \partial M$ and $(i, j) \in M$. Hence we have

$$\varphi(j) = \varphi(u^-) \quad (\forall j \in C \setminus \partial M), \tag{6}$$

$$\varphi(j) = \varphi(i) - c(e) \quad (\forall e = (i, j) \in M). \tag{7}$$

It follows from (4)–(6) that

$$\begin{aligned}
\sum_{i \in R} p_i + \sum_{j \in C} q_j + (m_Q + k)t &= \sum_{i \in R} \varphi(i) + \sum_{j \in C} (\varphi(u^-) - \varphi(j)) - (m_Q + k)\varphi(u^-) \\
&= \sum_{i \in R \cap \partial M} \varphi(i) + \sum_{j \in C \cap \partial M} (\varphi(u^-) - \varphi(j)) - (m_Q + k)\varphi(u^-) \\
&= \sum_{(i,j) \in M} (\varphi(i) - \varphi(j)) + |C \cap \partial M|\varphi(u^-) - (m_Q + k)\varphi(u^-) \\
&= c(M),
\end{aligned}$$

where the last step is due to (7) and $|C \cap \partial M| = |M| = m_Q + k$. Thus we obtain

$$\sum_{i \in R} p_i + \sum_{j \in C} q_j + (m_Q + k)t = c(M),$$

which implies that (p, q, t) is optimal.

If a weight function c is integer valued, (p, q, t) is integral by the construction rule. $\square$

3.3 Test for Tightness

We describe a necessary and sufficient condition for $\hat{\delta}_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A)$. For an integral feasible solution (p, q, t) of DLP(A, k), let us put

$$I^* = R_Q \cup \{i \in R_T \mid p_i > 0\} \quad \text{and} \quad J^* = \{j \in C \mid q_j > 0\}.$$

We call I^* and J^* *active rows* and *active columns*, respectively. The *tight coefficient matrix* $A^* = (A^*_{ij})$ is defined by

$$A^*_{ij} = (\text{the coefficient of } s^{p_i+q_j+t} \text{ in } A_{ij}(s)).$$

Note that A^* is an LM-matrix and $A(s) = (A_{ij}(s))$ is expressed by

$$A_{ij}(s) = s^{p_i+q_j+t}(A^*_{ij} + A^{\infty}_{ij}(s)) \tag{8}$$

with a strictly proper matrix $A^{\infty}(s) = (A^{\infty}_{ij}(s))$.

The following lemma gives a necessary and sufficient condition for $\hat{\delta}_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A)$, which is immediately derived from [18, Theorem 7].

Lemma 3.2. Let (p, q, t) be an optimal dual solution, I^* and J^* the active rows and columns, and A^* the tight coefficient matrix. Then the following three conditions (a)–(c) are equivalent.

(a) $\hat{\delta}_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A)$ holds.

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

(c) The following four conditions are satisfied:

- **(r1)** $\operatorname{rank} A^*[R, C] \geq m_Q + k$,
- **(r2)** $\operatorname{rank} A^*[I^*, C] = |I^*|$,
- **(r3)** $\operatorname{rank} A^*[R, J^*] = |J^*|$,
- **(r4)** $\operatorname{rank} A^*[I^*, J^*] \geq |I^*| + |J^*| - (m_Q + k)$.

Lemma 3.2 implies that we can check $\hat{\delta}_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A)$ efficiently by computing the ranks of four LM-matrices $A^*[R, C]$, $A^*[I^*, C]$, $A^*[R, J^*]$, and $A^*[I^*, J^*]$. This can be done by solving the corresponding independent matching problems described in Section 2.4. The optimality condition for (p, q, t) is given by the following variant, which is also derived from [18].

Lemma 3.3. Let (p, q, t) be a dual feasible solution, I^* and J^* the active rows and columns, and A^* the tight coefficient matrix. Then (p, q, t) is optimal if and only if the following four conditions are satisfied:

(t1) term-rank $A^*[R, C] \geq m_Q + k$,

(t2) term-rank $A^*[I^*, C] = |I^*|$,

(t3) term-rank $A^*[R, J^*] = |J^*|$,

(t4) term-rank $A^*[I^*, J^*] \geq |I^*| + |J^*| - (m_Q + k)$.

3.4 Matrix Modification

Let $A(s)$ be an LM-Laurent polynomial matrix such that $\delta_k^{\mathrm{LM}}(A) < \hat{\delta}_k^{\mathrm{LM}}(A)$. We describe the rule of modifying $A(s)$ into another LM-Laurent polynomial matrix $\tilde{A}(s)$. Since $\delta_k^{\mathrm{LM}}(A) < \hat{\delta}_k^{\mathrm{LM}}(A)$, it follows from Lemma 3.2 that at least one of conditions (r1)–(r4) is violated. In the test for tightness in Phase 2, we transform the tight coefficient matrix $A^* = \binom{Q^*}{T^*}$ into another LM-matrix $\tilde{A}^* = \binom{\tilde{Q}^*}{\tilde{T}^*}$ by applying Algorithm for the Rank of an LM-matrix described in Section 2.4. When Algorithm for the Rank of an LM-matrix terminates, it finds a nonsingular constant matrix U such that

$$\begin{pmatrix} \tilde{Q}^* \\ \tilde{T}^* \end{pmatrix} = \begin{pmatrix} U & O \\ O & I \end{pmatrix} \begin{pmatrix} Q^* \\ T^* \end{pmatrix}.$$

Let (p, q, t) be an optimal solution of $\mathrm{DLP}(A, k)$. With the use of U and (p, q, t), we transform $A(s)$ into another LM-Laurent polynomial matrix $\tilde{A}(s)$ defined by

$$\tilde{A}(s) = \mathrm{diag}[s^{p_1}, \ldots, s^{p_m}] \begin{pmatrix} U & O \\ O & I \end{pmatrix} \mathrm{diag}[s^{-p_1}, \ldots, s^{-p_m}] A(s). \tag{9}$$

In order to show that $\delta_k^{\mathrm{LM}}(A)$ is invariant under the transformation given in (9), we make use of the following lemma, which is immediately derived from [18, Lemma 11].

Lemma 3.4. Let $A(s)$ and $\tilde{U}(s)$ be rational function matrices, and $\tilde{A}(s) = \tilde{U}(s)A(s)$. Then $\delta_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(\tilde{A})$ if $\tilde{U}[R_Q, R_T] = O$, $\det \tilde{U}[R_Q, R_Q]$ is nonzero constant, and $\tilde{U}[R_T, R_T]$ is biproper.

The following lemma asserts $\delta_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(\tilde{A})$.

Lemma 3.5. Let $A(s)$ be an LM-Laurent polynomial matrix and $\tilde{A}(s)$ the LM-Laurent polynomial matrix defined by (9). Then we have $\delta_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(\tilde{A})$.

Proof. For $\tilde{U}(s) = \mathrm{diag}[s^{p_1}, \ldots, s^{p_m}] \begin{pmatrix} U & O \\ O & I \end{pmatrix} \mathrm{diag}[s^{-p_1}, \ldots, s^{-p_m}]$, we denote the row/column sets by $R_Q \cup R_T$. Then $\tilde{U}[R_Q, R_T] = O$ holds and $\tilde{U}[R_T, R_T] = I$ is biproper. In addition, $\det \tilde{U}[R_Q, R_Q] = \det U$ is nonzero constant, because U is a nonsingular constant matrix. Hence we obtain $\delta_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(\tilde{A})$ by Lemma 3.4. □

Let $\tilde{c}_{ij}$ denote the degree of the (i, j) entry of $\tilde{A}(s)$. We now prove that an optimal solution (p, q, t) of $\mathrm{DLP}(A, k)$ is feasible for $\mathrm{DLP}(\tilde{A}, k)$ but not optimal.

Lemma 3.6. Let $A(s)$ be an LM-Laurent polynomial matrix with $\delta_k^{\mathrm{LM}}(A) < \hat{\delta}_k^{\mathrm{LM}}(A)$, and $\tilde{A}(s)$ the LM-Laurent polynomial matrix defined by (9). Then an optimal solution (p, q, t) of $\mathrm{DLP}(A, k)$ is feasible for $\mathrm{DLP}(\tilde{A}, k)$ but not optimal.

Proof. We put

$$F(s) = s^{-t}\, \mathrm{diag}[s^{-p_1}, \ldots, s^{-p_m}] \tilde{A}(s)\, \mathrm{diag}[s^{-q_1}, \ldots, s^{-q_n}].$$

The degree of the (i,j) entry of $F(s)$ satisfies $\deg F_{ij}(s) = \tilde{c}_{ij} - p_i - q_j - t$. By (8) and (9), $F(s)$ is transformed as

$$
\begin{aligned}
F(s) =& s^{-t} \begin{pmatrix} U & O \\ O & I \end{pmatrix} \mathrm{diag}[s^{-p_1}, \ldots, s^{-p_m}] A(s) \,\mathrm{diag}[s^{-q_1}, \ldots, s^{-q_n}] \\
=& \begin{pmatrix} U & O \\ O & I \end{pmatrix} (A^* + A^{\infty}(s)),
\end{aligned}
$$

where $A^{\infty}(s)$ denotes a strictly proper matrix. This indicates $\deg F_{ij}(s) \leq 0$, because U and A^* are constant matrices. Thus we obtain $\tilde{c}_{ij} - p_i - q_j - t \leq 0$, which implies that (p,q,t) is feasible for $\mathrm{DLP}(\tilde{A}, k)$.

We show that (p,q,t) is not optimal for $\mathrm{DLP}(\tilde{A}, k)$. Let us set $\tilde{A}^*_{(p,q,t)} = \begin{pmatrix} U & O \\ O & I \end{pmatrix} A^*$. Then $\tilde{A}^*_{(p,q,t)}$ is the tight coefficient matrix of $\tilde{A}(s)$ with respect to (p,q,t). By $\delta_k^{\mathrm{LM}}(A) < \hat{\delta}_k^{\mathrm{LM}}(A)$ and Lemma 3.2, at least one of conditions (r1)–(r4) is not fulfilled. Let us assume that A^* violates (r2) and $\tilde{A}^*_{(p,q,t)}$ is obtained by applying Algorithm for the Rank of an LM-matrix to $A^*[I^*, C]$. Then it follows from Lemma 2.3 that $\mathrm{rank}\, \tilde{A}^*_{(p,q,t)}[I^*, C] = \text{term-rank}\, \tilde{A}^*_{(p,q,t)}[I^*, C]$. Hence we have

$$
\text{term-rank}\, \tilde{A}^*_{(p,q,t)}[I^*, C] = \mathrm{rank}\, \tilde{A}^*_{(p,q,t)}[I^*, C] = \mathrm{rank}\, A^*[I^*, C] < |I^*|.
$$

Thus $\tilde{A}^*_{(p,q,t)}$ violates (t2), which implies that (p,q,t) is not optimal for $\mathrm{DLP}(\tilde{A}, k)$ by Lemma 3.3. If (r1), (r3), or (r4) is violated, we can prove that (p,q,t) is not optimal for $\mathrm{DLP}(\tilde{A}, k)$ in a similar way. □

3.5 Dual Updates

Let (p,q,t) be an optimal solution of $\mathrm{DLP}(A, k)$. By Lemma 3.6, (p,q,t) is feasible for $\mathrm{DLP}(\tilde{A}, k)$. For (p,q,t) and another feasible solution (p', q', t') of $\mathrm{DLP}(\tilde{A}, k)$, we consider the amount of change Δ in the value of the dual objective function defined by

$$
\Delta = \left(\sum_{i \in R} p'_i + \sum_{j \in C} q'_j + (m_Q + k)t' \right) - \left(\sum_{i \in R} p_i + \sum_{j \in C} q_j + (m_Q + k)t \right).
$$

With the use of (p,q,t), we construct a feasible solution (p', q', t') which satisfies $\Delta < 0$. By repeating this procedure, we find an optimal dual solution of $\mathrm{DLP}(\tilde{A}, k)$.

Let $G^* = (R, C; E^*)$ be a bipartite graph with $E^* = \{(i,j) \in E(\tilde{A}) \mid p_i + q_j + t = \tilde{c}_{ij}\}$. Since (p,q,t) is not optimal for $\mathrm{DLP}(\tilde{A}, k)$ by Lemma 3.6, at least one of conditions (t1)–(t4) for $\tilde{A}^*_{(p,q,t)}$ is violated. For each case, we construct another feasible solution (p', q', t') with $\Delta < 0$ as follows.

Case1: (t1) is violated

Since the maximum size of a matching in $G^* = (R, C; E^*)$ is strictly less than $m_Q + k$, G^* has a cover W with $|W| < m_Q + k$. We now define

$$p'_i = \begin{cases} p_i + 1 & (i \in R \cap W) \\ p_i & (i \in R \setminus W) \end{cases}, \qquad q'_j = \begin{cases} q_j + 1 & (j \in C \cap W) \\ q_j & (j \in C \setminus W) \end{cases}, \qquad t' = t - 1.$$

Then it holds that

$$\Delta = |R \cap W| + |C \cap W| - (m_Q + k) = |W| - (m_Q + k) < 0.$$

We show that (p', q', t') is feasible for $\mathrm{DLP}(\tilde{A}, k)$. First, consider an edge $(i, j) \in E(\tilde{A})$ satisfying $i \in W$ or $j \in W$. Since we have $(p'_i + q'_j + t') - (p_i + q_j + t) \geq 0$, it holds that $p'_i + q'_j + t' \geq p_i + q_j + t \geq \tilde{c}_{ij}$ by the feasibility of (p, q, t) for $\mathrm{DLP}(\tilde{A}, k)$. Next, consider an edge $(i, j) \in E(\tilde{A})$ satisfying $i \notin W$ and $j \notin W$. It follows from $(i, j) \notin E^*$ that $p_i + q_j + t > \tilde{c}_{ij}$. Hence $p'_i + q'_j + t' = p_i + q_j + t - 1 > \tilde{c}_{ij} - 1$ holds. Thus, (p', q', t') is feasible for $\mathrm{DLP}(\tilde{A}, k)$.

Case2: (t2) is violated

Since the maximum size of a matching in $G^*[I^* \cup C] = (I^*, C; E^*[I^* \cup C])$ is strictly less than $|I^*|$, $G^*[I^*, C]$ has a cover $W \subseteq I^* \cup C$ with $|W| < |I^*|$. We now define

$$p'_i = \begin{cases} p_i & (i \in (I^* \cap W) \cup (R \setminus I^*)) \\ p_i - 1 & (i \in I^* \setminus W) \end{cases}, \qquad q'_j = \begin{cases} q_j + 1 & (j \in C \cap W) \\ q_j & (j \in C \setminus W) \end{cases}, \qquad t' = t.$$

Then it holds that

$$\Delta = -|I^* \setminus W| + |C \cap W| = |W| - |I^*| < 0.$$

We can show that (p', q', t') is feasible for $\mathrm{DLP}(\tilde{A}, k)$ in a similar way to Case 1.

Case3: (t3) is violated

Since the maximum size of a matching in $G^*[R \cup J^*] = (R, J^*; E^*[R \cup J^*])$ is strictly less than $|J^*|$, $G^*[R, J^*]$ has a cover $W \subseteq R \cup J^*$ with $|W| < |J^*|$. We now define

$$p'_i = \begin{cases} p_i + 1 & (i \in R \cap W) \\ p_i & (i \in R \setminus W) \end{cases}, \qquad q'_j = \begin{cases} q_j & (j \in (J^* \cap W) \cup (C \setminus J^*)) \\ q_j - 1 & (j \in J^* \setminus W) \end{cases}, \qquad t' = t.$$

Then it holds that

$$\Delta = |R \cap W| - |J^* \setminus W| = |W| - |J^*| < 0.$$

We can show that (p', q', t') is feasible for $\mathrm{DLP}(\tilde{A}, k)$ in a similar way to Case 1.

Case4: (t4) is violated

Since the maximum size of a matching in $G^*[I^* \cup J^*] = (I^*, J^*; E^*[I^* \cup J^*])$ is strictly less than $|I^*| + |J^*| - (m_Q + k)$, $G^*[I^*, J^*]$ has a cover $W \subseteq I^* \cup J^*$ with $|W| < |I^*| + |J^*| - (m_Q + k)$. We now define

$$p'_i = \begin{cases} p_i - 1 & (i \in I^* \setminus W) \\ p_i & (i \in (I^* \cap W) \cup (R \setminus I^*)) \end{cases}, \quad q'_j = \begin{cases} q_j - 1 & (j \in J^* \setminus W) \\ q_j & (j \in (J^* \cap W) \cup (C \setminus J^*)) \end{cases}, \quad t' = t + 1.$$

Then it holds that

$$\Delta = -|I^* \setminus W| - |J^* \setminus W| + (m_Q + k) = |W| - (|I^*| + |J^*| - (m_Q + k)) < 0.$$

We can show that (p', q', t') is feasible for $\mathrm{DLP}(\tilde{A}, k)$ in a similar way to Case 1.

4 Complexity Analysis

This section is devoted to complexity analysis of our combinatorial relaxation algorithm.

4.1 Worst Case Analysis

We analyze the complexity of our combinatorial relaxation algorithm, which is now described as follows.

Algorithm for Computing $\delta_k^{\mathrm{LM}}(A)$

Step 1 Find a maximum weight matching M in $G(A)$ by using an efficient combinatorial algorithm.

Step 2 Construct an optimal solution (p, q, t) of $\mathrm{DLP}(A, k)$ from M.

Step 3 Apply Algorithm for the Rank of an LM-matrix to $A^*[R, C]$, $A^*[I^*, C]$, $A^*[R, J^*]$, and $A^*[I^*, J^*]$. If (r1)–(r4) hold, then return $\hat{\delta}_k^{\mathrm{LM}}(A)$ and halt.

Step 4 Modify $A(s)$ to another matrix $\tilde{A}(s)$ defined by (9).

Step 5 Construct an optimal solution $(\tilde{p}, \tilde{q}, \tilde{t})$ of $\mathrm{DLP}(\tilde{A}, k)$ by performing the procedure given in Section 3.5. Go back to Step 3.

This algorithm is dominated by the computation of $\tilde{A}(s)$ in Step 4. We discuss its time and space complexities in the following.

For an integral optimal solution (p, q, t) of $\mathrm{DLP}(A, k)$, we denote $\mathrm{diag}[s^{p_1}, \ldots, s^{p_m}]$ and $\mathrm{diag}[s^{q_1}, \ldots, s^{q_n}]$ by $V_{\mathrm{r}}(s)$ and $V_{\mathrm{c}}(s)$, respectively. By the definition of the tight coefficient matrix A^*, the LM-Laurent polynomial matrix $A(s)$ is expressed as

$$A(s) = s^t V_{\mathrm{r}}(s) \left(A^* + \frac{1}{s} A_1 + \frac{1}{s^2} A_2 + \cdots + \frac{1}{s^l} A_l \right) V_{\mathrm{c}}(s)$$

for some integer l, where A_i denotes a constant matrix for $i = 1, 2, \ldots, l$. It follows from (9) that

$$\tilde{A}(s) = s^t V_{\mathrm{r}}(s) \begin{pmatrix} U & O \\ O & I \end{pmatrix} \left(A^* + \frac{1}{s} A_1 + \frac{1}{s^2} A_2 + \cdots + \frac{1}{s^l} A_l \right) V_{\mathrm{c}}(s). \tag{10}$$

Thus, it suffices to perform constant matrix multiplications $\begin{pmatrix} U & O \\ O & I \end{pmatrix} A^*, \begin{pmatrix} U & O \\ O & I \end{pmatrix} A_1, \ldots, \begin{pmatrix} U & O \\ O & I \end{pmatrix} A_l$. In (10), each entry of $\tilde{A}(s)$ is normalized with the aid of the dual variable (p, q, t). This leads to the following lemma, which states that we may assume that l is at most $\hat{\delta}_k^{\mathrm{LM}}(A)$.

Lemma 4.1. Let $A(s) = \binom{Q(s)}{T(s)}$ be an LM-polynomial matrix such that $Q(s)$ is of full row rank, and (p, q, t) an optimal dual solution of $\mathrm{DLP}(A, k)$. Then $\delta_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(\bar{A})$ holds for

$$\bar{A}(s) = s^t V_{\mathrm{r}}(s) \left(A^* + \frac{1}{s} A_1 + \frac{1}{s^2} A_2 + \cdots + \frac{1}{s^{\hat{\delta}_k^{\mathrm{LM}}(A)}} A_{\hat{\delta}_k^{\mathrm{LM}}(A)} \right) V_{\mathrm{c}}(s),$$

which is obtained by ignoring the terms $\dfrac{1}{s^d} A_d$ with $d > \hat{\delta}_k^{\mathrm{LM}}(A)$.

Proof. The initial value of the dual objective function is $\hat{\delta}_k^{\mathrm{LM}}(A)$ and we showed in Section 3.5 that $\Delta < 0$ holds at each step of updating a dual feasible solution. This implies that a dual feasible solution is updated at most $\hat{\delta}_k^{\mathrm{LM}}(A)$ times throughout the algorithm. At each update step, $p_i + q_j + t$ decreases by at most one by the construction rule given in Section 3.5. Hence, for any i and j, $p_i + q_j + t$ decreases by at most $\hat{\delta}_k^{\mathrm{LM}}(A)$ throughout the algorithm. Thus we may ignore the terms $\dfrac{1}{s^d} A_d$ with $d > \hat{\delta}_k^{\mathrm{LM}}(A)$. □

Recall the notation $m_Q = |R_Q|$. By Lemma 4.1, the time and space complexities of the algorithm are as follows.

Theorem 4.2. Let $A(s) = \binom{Q(s)}{T(s)}$ be an $m \times n$ LM-polynomial matrix such that $Q(s)$ is of full row rank, and $d_{\max}$ the maximum degree of an entry in $A(s)$. Then Algorithm for Computing $\delta_k^{\mathrm{LM}}(A)$ runs in $O((m_Q + k)^2 m_Q^{\omega - 1} n d_{\max}^2)$ time and $O((m_Q + k) m n d_{\max})$ space, where $\omega < 2.38$ is the matrix multiplication exponent.

Proof. Let us set $D = \hat{\delta}_k^{\mathrm{LM}}(A)$. At each matrix modification step, we execute constant matrix multiplications $(D+1)$ times by Lemma 4.1, which costs $O(m_Q^{\omega-1} n D)$ time. Since a dual feasible solution is updated at most D times, the time complexity of the algorithm is $O(m_Q^{\omega-1} n D^2)$. The space complexity is $O(mnD)$, because we need to store constant matrices A^*, $A_1, \ldots, A_D$. By $D = O((m_Q + k) d_{\max})$, the algorithm requires $O((m_Q + k)^2 m_Q^{\omega-1} n d_{\max}^2)$ time and $O((m_Q + k) m n d_{\max})$ space. □

Let r denote the rank of the LM-polynomial matrix $A(s)$. In order to obtain the Smith-McMillan form at infinity or the Kronecker canonical form, we need to compute $\delta_k^{\mathrm{LM}}(A)$ for $k = 1, \ldots, r - m_Q$. A straightforward approach requires r applications of the algorithm. The following lemma, which is derived from [11, Lemma 3.4], bounds the number of possible modifications when k ranges from 1 to $r - m_Q$.

Lemma 4.3. Let $A(s) = \binom{Q(s)}{T(s)}$ be an LM-polynomial matrix of rank r such that $Q(s)$ is of full row rank, and $d_{\max}$ the maximum degree of an entry in $A(s)$. Then, the total number of modifications for $k = 1, \ldots, r - m_Q$ is bounded by $rd_{\max}$.

Lemma 4.3 leads to the following theorem.

Theorem 4.4. Let $A(s) = \binom{Q(s)}{T(s)}$ be an $m \times n$ LM-polynomial matrix such that $Q(s)$ is of full row rank, r the rank of $A(s)$, and $d_{\max}$ the maximum degree of an entry in $A(s)$. We can obtain $\delta_k^{\mathrm{LM}}(A)$ for all k from 1 to $r - m_Q$ in $O(r^2 m_Q^{\omega-1} n d_{\max}^2)$ time.

Proof. Let us set $D = \hat{\delta}_{r-m_Q}^{\mathrm{LM}}(A) = O(rd_{\max})$. In a similar way to the proof of Theorem 4.2, each matrix modification step costs $O(m_Q^{\omega-1} nD)$ time. Since Lemma 4.3 asserts that the total number of modifications for $k = 1, \ldots, r - m_Q$ is bounded by $rd_{\max}$, we obtain $\delta_k^{\mathrm{LM}}(A)$ for all k in $O(r^2 m_Q^{\omega-1} n d_{\max}^2)$ time. $\square$

Let $\tilde{A}(s)$ be an $m \times n$ mixed polynomial matrix of rank $\tilde{r}$, and $A(s)$ the associated LM-polynomial matrix of rank r defined by (2). By $m_Q = O(m)$ and $r = O(m)$, we can compute $\delta_k(\tilde{A})$ for the fixed parameter k as well as for all k from 1 to $\tilde{r}$ in $O(m^{\omega+1} n d_{\max}^2)$ time.

4.2 Probabilistic Analysis

Let η be a prime. For an LM-polynomial matrix $A(s) = \binom{Q(s)}{T(s)}$, assuming that each nonzero coefficient of $Q(s)$ is chosen uniformly at random from the nonzero integers between $-\eta + 1$ and $\eta - 1$, we analyze the probability of the event that the initial estimate $\hat{\delta}_k^{\mathrm{LM}}(A)$ coincides with the true value of $\delta_k^{\mathrm{LM}}(A)$ when η is sufficiently large.

Let (p, q, t) be an optimal dual solution of $\mathrm{DLP}(A, k)$, I^* and J^* the active rows and columns, and A^* the tight coefficient matrix. In the analysis, we use the following lemma.

Lemma 4.5 ([18, Lemma 4])**.** For $I \subseteq R$ and $J \subseteq C$ such that $|I| = |J| = m_Q + k$, we have $\hat{\delta}_k^{\mathrm{LM}}(A[I, J]) = \hat{\delta}_k^{\mathrm{LM}}(A)$ if and only if $I \supseteq I^*$, $J \supseteq J^*$, and

$$\text{term-rank}\, A^*[I, J] = |I| = |J| = m_Q + k. \tag{11}$$

In the computation of $\hat{\delta}_k^{\mathrm{LM}}(A)$ by using a matching algorithm, we find a pair of $I \supseteq I^*$ and $J \supseteq J^*$ satisfying (11). If $\text{term-rank}\, A^*[I, J] = \operatorname{rank} A^*[I, J]$ holds, then $\hat{\delta}_k^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A)$ also holds by Lemma 3.2. In order to analyze the probability that $\text{term-rank}\, A^*[I, J] = \operatorname{rank} A^*[I, J]$, we consider the following problem.

Problem 4.6. Let B be a $k \times k$ constant matrix and η a prime. We assume that the positions of nonzero entries in B are fixed, $\text{term-rank}\, B = k$, and each nonzero entry is chosen uniformly at random from the nonzero integers between $-\eta + 1$ and $\eta - 1$. What is the probability that $\operatorname{rank} B = k$ when η is sufficiently large relative to k?

The row set and column set of $B = (B_{ij})$ are denoted by $\bar{R}$ and $\bar{C}$. Consider a bipartite graph $G(B) = (\bar{R}, \bar{C}; \bar{E})$ with $\bar{E} = \{(i, j) \mid i \in \bar{R}, j \in \bar{C}, B_{ij} \neq 0\}$. If $G(B)$ has only one perfect matching, $\text{term-rank}\, B = \operatorname{rank} B$ always holds, because there is no numerical cancellation.

Note that

$$\Pr\{\operatorname{rank} B \neq k\} = \Pr\{\det B = 0\} \leq \Pr\left\{\det B \equiv 0 \pmod{\eta}\right\}. \tag{12}$$

Suppose we are given a nonzero integer between $-\eta+1$ and $\eta-1$ to each nonzero entry in the rows except the last one. By the Laplace expansion, we obtain

$$\det B = \sum_{1\leq j\leq k} c_j B_{kj},$$

where $c_j = \pm \det B[\bar{R} \setminus \{k\}, \bar{C} \setminus \{j\}]$ is a constant. Consider $L = \{j \mid c_j \not\equiv 0 \pmod{\eta}, B_{kj} \neq 0\}$ and denote each entry of L by $i_1, i_2, \ldots, i_l$. Moreover, we define the probability P_l by

$$P_l = \Pr\left\{ \sum_{j\in L} c_j B_{kj} \equiv 0 \pmod{\eta} \;\middle|\; |L| = l \right\}.$$

Then, it follows from (12) that

$$\begin{aligned} \Pr\{\operatorname{rank} B \neq k\} &\leq \Pr\left\{\det B \equiv 0 \pmod{\eta}\right\} \\ &= \Pr\{|L| = 0\} + \sum_{l=2}^{k} \Pr\{|L| = l\} \times P_l \\ &\leq \Pr\{\operatorname{rank} B' \neq k-1\} + \sum_{l=2}^{k} P_l, \end{aligned} \tag{13}$$

where B' is a $(k-1) \times (k-1)$ constant matrix.

In the analysis of P_l, we use the following fundamental fact.

Lemma 4.7. Let η be a prime. Given integers α with $\alpha \not\equiv 0 \pmod{\eta}$ and β, there exist at most two integers z which satisfy $-\eta+1 \leq z \leq \eta-1$ and $\alpha z + \beta \equiv 0 \pmod{\eta}$.

We assign nonzero integers between $-\eta+1$ and $\eta-1$ to B_{kj} for $j \in L \setminus \{i_l\}$. Then B_{ki_l} needs to satisfy

$$-\eta+1 \leq B_{ki_l} \leq \eta-1, \quad B_{ki_l} \neq 0, \quad c_l B_{ki_l} \equiv -\sum_{j\in L\setminus\{i_l\}} c_j B_{kj} \pmod{\eta}. \tag{14}$$

By Lemma 4.7, the probability of choosing B_{ki_l} which satisfies (14) is at most $\frac{1}{\eta-1}$. This implies that

$$P_l \leq \frac{1}{\eta-1} \Pr\left\{ \sum_{j\in L\setminus\{i_l\}} c_j B_{kj} \not\equiv 0 \pmod{\eta} \right\} = \frac{1}{\eta-1}(1 - P_{l-1}) \leq \frac{1}{\eta-1}.$$

Hence, it follows from (13) that

$$\Pr\{\operatorname{rank} B \neq k\} \leq \Pr\{\operatorname{rank} B' \neq k-1\} + \frac{k-1}{\eta-1}.$$

Since $\Pr\{\operatorname{rank} B'' \neq 2\} \leq \frac{1}{\eta-1}$ for a 2×2 constant matrix B'', this implies

$$\Pr\{\operatorname{rank} B \neq k\} \leq \frac{1}{\eta-1} + \frac{2}{\eta-1} + \cdots + \frac{k-1}{\eta-1} = \frac{k(k-1)}{2(\eta-1)}.$$

The upper bound in the right-hand side converges to zero when η grows more rapidly than k^2. Thus we obtain the following proposition.

Proposition 4.8. Let B be a $k \times k$ constant matrix and η a sufficiently large prime relative to k. We assume that the positions of nonzero entries in B are fixed, $\operatorname{term-rank} B = k$, and each nonzero entry is chosen uniformly at random from the nonzero integers between $-\eta + 1$ and $\eta - 1$. Then $\operatorname{rank} B = \operatorname{term-rank} B$ holds with high probability.

By Proposition 4.8, an LM-matrix A^* satisfies $\operatorname{term-rank} A^*[I, J] = \operatorname{rank} A^*[I, J]$ with high probability. This means that the combinatorial relaxation algorithm terminates in many cases by showing that an initial estimate coincides with the true value. If the algorithm finds the true value at the first iteration, we do not transform the given LM-polynomial matrix into another one. Thus, it runs in $O(n^{2+\frac{1}{4-\omega}})$ time by using Gabow and Xu's algorithm [6] when we check the validity of an estimate.

Otherwise, the number of matrix modifications is at most $\hat{\delta}_k^{\mathrm{LM}}(A) - \delta_k^{\mathrm{LM}}(A)$. In the proof of Theorem 4.2, this value is bounded by $O((m_Q + k)d_{\max})$. In most cases, however, the difference is so small that it can be regarded as a constant. Thus the algorithm effectively runs in $O((m_Q + k)m_Q^{\omega-1} n d_{\max})$ time, which is also much faster than suggested by Theorem 4.2.

5 Application to Linear Valuated Independent Assignment

As a generalization of matroids, Dress and Wenzel [5] introduced *valuated matroids*. A valuated matroid $\mathbf{M} = (V, \mathcal{B}, \omega)$ is a triple of a ground set V, a base family $\mathcal{B} \subseteq 2^V$, and a function $\omega : \mathcal{B} \to \mathbb{R}$ that satisfy the following axiom (VM).

(VM) For any $B, B' \in \mathcal{B}$ and $u \in B \setminus B'$, there exists $v \in B' \setminus B$ such that $B \setminus \{u\} \cup \{v\} \in \mathcal{B}$, $B' \cup \{u\} \setminus \{v\} \in \mathcal{B}$, and $\omega(B) + \omega(B') \leq \omega(B \setminus \{u\} \cup \{v\}) + \omega(B' \cup \{u\} \setminus \{v\})$.

The function ω is called a *valuation*.

Murota [19, 20] introduced the *valuated independent assignment problem* as a generalization of the independent matching problem.

[Valuated Independent Assignment Problem (VIAP)]
Given a bipartite graph $G = (V^+, V^-; E)$ with vertex sets V^+, V^- and edge set E, a pair of valuated matroids $\mathbf{M}^+ = (V^+, \mathcal{B}^+, \omega^+)$ and $\mathbf{M}^- = (V^-, \mathcal{B}^-, \omega^-)$, and a weight function $w : E \to \mathbb{R}$, find a matching $M \subseteq E$ that maximizes

$$\Omega(M) := w(M) + \omega^+(\partial^+ M) + \omega^-(\partial^- M)$$

subject to $\partial^+ M \in \mathcal{B}^+$ and $\partial^- M \in \mathcal{B}^-$.

Let $\mathbf{M}^+$ and $\mathbf{M}^-$ be linear valuated matroids represented by polynomial matrices $Q_+(s)$ and $Q_-(s)$, respectively. For a bipartite graph $G = (V^+, V^-; E)$, let $T(s) = \sum_{k=0}^{N} s^k T_k$ be a polynomial matrix which satisfies (MP-T), $E = \{(i,j) \mid T_{ij}(s) \neq 0\}$, and $\deg T_{ij}(s) = w(e)$ for $e = (i,j) \in E$. Then the optimal value of $\Omega(M)$ is equal to the degree of the determinant of the mixed polynomial matrix

$$A(s) = \begin{pmatrix} O & Q_+(s)^\top & O \\ Q_-(s) & O & I \\ O & I & T(s) \end{pmatrix}.$$

We obtain $\deg \det A(s)$ by using our combinatorial relaxation algorithm. This means that we can find the optimal value of the linear valuated independent assignment problem by solving a sequence of independent matching problems.

Acknowledgement

The authors are grateful to Kazuo Murota for helpful comments on the manuscript.

References

[1] E. H. Bareiss, *Sylvester's identity and multistep integer-preserving Gaussian elimination*, Mathematics of Computation, 22 (1968), pp. 565–578.

[2] E. H. Bareiss, *Computational solutions of matrix problems over an integral domain*, IMA Journal of Applied Mathematics, 10 (1972), pp 68–104.

[3] D. G. Cantor and E. Kaltofen, *On fast multiplication of polynomials over arbitrary algebras*, Acta Informatica, 28 (1991), pp. 693–701.

[4] J. Demmel and B. Kågström, *Accurate solutions of ill-posed problems in control theory*, SIAM Journal on Matrix Analysis and Applications, 9 (1988), pp. 126–145.

[5] A. W. M. Dress and W. Wenzel, *Valuated matroids*, Advances in Mathematics, 93 (1992), pp. 214–250.

[6] H. N. Gabow and Y. Xu, *Efficient theoretic and practical algorithms for linear matroid intersection problems*, J. Comput. Syst. Sci., 53 (1996), pp. 129–147.

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

[8] N. J. A. Harvey, *Algebraic algorithms for matching and matroid problems*, SIAM Journal on Computing, 39 (2009), pp. 679–702.

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

[10] S. Iwata and K. Murota, *Combinatorial relaxation algorithm for mixed polynomial matrices*, Mathematical Programming, 90 (2001), pp. 353–371.

[11] S. Iwata, K. Murota, and I. Sakuta, *Primal-dual combinatorial relaxation algorithms for the maximum degree of subdeterminants*, SIAM Journal on Scientific Computing, 17 (1996), pp. 993–1012.

[12] S. Iwata and M. Takamatsu, *On the Kronecker canonical form of mixed matrix pencils*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 44–71.

[13] M. Mucha and P. Sankowski, *Maximum matchings via Gaussian elimination*, Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 248–255, 2004.

[14] M. Mucha and P. Sankowski, *Maximum matchings in planar graphs via Gaussian elimination*, Algorithmica, 45 (2006), pp. 3–20.

[15] K. Murota, *Systems Analysis by Graphs and Matroids — Structural Solvability and Controllability*, Springer-Verlag, Berlin, 1987.

[16] K. Murota, *Computing Puiseux-series solutions to determinantal equations via combinatorial relaxation*, SIAM Journal on Computing, 19 (1990), pp. 1132–1161.

[17] K. Murota, *Computing the degree of determinants via combinatorial relaxation*, SIAM Journal on Computing, 24 (1995), pp. 765–796.

[18] K. Murota, *Combinatorial relaxation algorithm for the maximum degree of subdeterminants: Computing Smith-McMillan form at infinity and structural indices in Kronecker form*, Applicable Algebra in Engineering, Communication and Computing, 6 (1995), pp. 251–273.

[19] K. Murota, *Valuated matroid intersection, I: optimality criteria*, SIAM Journal on Discrete Mathematics, 9 (1996), pp. 545–561.

[20] K. Murota, *Valuated matroid intersection, II: algorithms*, SIAM Journal on Discrete Mathematics, 9 (1996), pp. 562–576.

[21] K. Murota, *On the degree of mixed polynomial matrices*, SIAM Journal on Matrix Analysis and Applications, 20 (1999), pp. 196–227.

[22] K. Murota, *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin, 2000.

[23] K. Murota and M. Iri, *Structural solvability of systems of equations — A mathematical formulation for distinguishing accurate and inaccurate numbers in structural analysis of systems*, Japan Journal of Industrial and Applied Mathematics, 2 (1985), pp. 247–271.

[24] P. Sankowski, *Maximum weight bipartite matching in matrix multiplication time*, Theoretical Computer Science, 410 (2009), pp. 4480–4488.

[25] A. Storjohann, *Algorithms for Matrix Canonical Forms*, Ph.D. thesis, ETH Zürich, 2000.

[26] J. S. Thorp, *The singular pencil of a linear dynamical system*, International Journal of Control, 18 (1973), pp. 577–596.

[27] G. C. Verghese and T. Kailath, *Rational matrix structure*, IEEE Transactions on Automatic Control, AC-26 (1981), pp. 434–439.