JSIAM Letters Vol.7 (2015) pp.49–52 ©2015 Japan Society for Industrial and Applied Mathematics

Combinatorial relaxation algorithm for the entire sequence of the maximum degree of minors in mixed polynomial matrices

Shun Sato[1]

[1] Graduate School of Information Science and Technology, the University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan

E-mail *shun_sato@mist.i.u-tokyo.ac.jp*

Received December 26, 2014, Accepted March 28, 2015

Abstract

Iwata–Takamatsu (2013) showed that the maximum degree of minors in mixed polynomial matrices for a specified order can be computed by combinatorial relaxation type algorithm. In this letter, based on their algorithm, we propose an efficient combinatorial relaxation algorithm for computing the entire sequence of the maximum degree of minors. In our previous work, we dealt with a similar problem for rational function matrices, where the efficiency derived from the discrete concavity of valuated bimatroids. We follow the same line of discussion; but, technical details are different due to special characteristics of mixed matrices.

Keywords combinatorial relaxation, degree of minor, valuated bimatroid, discrete concavity, mixed matrix

Research Activity Group Discrete Systems

1. Introduction

The concept of mixed matrices [1] was introduced as a mathematical tool for description of physical systems. A mixed matrix contains two kinds of numbers as follows:

- Accurate Numbers (Fixed Constants)
  They frequently represent conservation laws, and are precise in values. These numbers should be treated numerically.
- Inaccurate Numbers (Independent Parameters)
  They often denote physical characteristics and are not precise in values. Since they can be assumed to be independent, these numbers should be treated combinatorially as nonzero parameters.

Let $A(x)$ be a mixed polynomial matrix over certain fields (which we will define later). R and C denote the row-set and column-set of A, respectively. $\deg f$ denotes the degree of a polynomial function f. We define the maximum degree of minors of order k as

$$\delta_k(A) := \max_{I\subseteq R, J\subseteq C} \{\deg\det A[I,J] \mid |I| = |J| = k\},$$

where $A[I,J]$ denotes the submatrix of A with $I \subseteq R$ and $J \subseteq C$. We set $\delta_0(A) = 0$. The entire sequence of maximum degree of minors $\{\delta_k(A)\}_{k=0}^{\mathrm{rank}\,A}$ can be used for computing the Smith–McMillan form at infinity [1], which is often used in control theory. Thus we aim to develop an efficient algorithm for computing $\{\delta_k(A)\}_{k=0}^{\mathrm{rank}\,A}$.

The maximum degree of minors can be defined on rational function matrices, which include mixed polynomial matrices as special cases. Murota [2] showed that the maximum degree of minors on rational function matrices for a specified order can be computed by the general framework of "combinatorial relaxation" (Murota [3]). Our previous work [4] then showed an efficient combinatorial relaxation type algorithm for finding the entire sequence of the maximum degree of minors in rational function matrices. On the other hand, for mixed polynomial matrices, Iwata–Takamatsu [5] gave an algorithm for a specified order. In this letter, we extend the idea of [4] to mixed polynomial matrices and develop a combinatorial relaxation type algorithm for finding the entire sequence of the maximum degree of minors. The efficiency of the proposed algorithm is based on two theorems (Theorems 5 and 6) concerning "tightness" of combinatorial relaxations.

2. Preliminaries

2.1 Mixed Polynomial Matrices and $\delta_k(A)$'s

We define *mixed polynomial matrices* (see, e.g., [1]). In the definition below, fields K and F represent accurate and inaccurate numbers, respectively; the coefficients of $Q_{ij}(x)$ correspond to fixed constants, and the coefficients of $T_{ij}(x)$ to independent parameters.

Definition 1 (Mixed Polynomial Matrices) *Let K be a subfield of a field F. A polynomial matrix $M(x)$ is called a mixed polynomial matrix over (K,F) if $M(x)$ can be represented as $M(x) = Q(x) + T(x)$, where $Q(x)$ and $T(x)$ satisfy the following conditions:*

(MP-Q) *$Q(x)$ is a polynomial matrix over K.*

(MP-T) *$T(x)$ is a polynomial matrix over F, and the set $\mathcal{T}$ of nonzero coefficients of $T(x)$ is algebraically independent over K.*

A mixed polynomial matrix $M(x)$ is called a *layered mixed* (LM)*-polynomial matrix* if it is in the form

$$M(x) = \begin{bmatrix} Q(x) \\ T(x) \end{bmatrix}, \tag{1}$$

where $Q(x)$ and $T(x)$ satisfy **(MP-Q)** and **(MP-T)**.

Finding the maximum degree of minors in a mixed polynomial matrix can be reduced to a corresponding problem on an LM-polynomial matrix. Let us start with an $m \times n$ mixed polynomial matrix $\tilde{A}(x) = \tilde{Q}(x) + \tilde{T}(x)$ with $\tilde{R}$ and $\tilde{C}$ being the row-set and the column-set. Then, we can associate $\tilde{A}$ with an LM-polynomial matrix

$$A(x) = \left[\begin{array}{cc} \mathrm{diag}[x^{d_1}, \ldots, x^{d_m}] & \tilde{Q}(x) \\ \mathrm{diag}[t_1 x^{d_1}, \ldots, t_m x^{d_m}] & \tilde{T}(x) \end{array} \right], \qquad (2)$$

where $d_i := \max_{j \in C} \deg \tilde{Q}_{ij}$, and t_i is a new parameter for all $i \in \tilde{R}$. We define R and C as the row-set and column-set of A. We define $\delta_k^{\mathrm{LM}}(A)$ as follows:

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

where $R_Q \subseteq R$ and $R_T \subseteq R$ denote the row-subsets corresponding to $\tilde{Q}$ and $\tilde{T}$. Then, $\delta_k(\tilde{A}) = \delta_k^{\mathrm{LM}}(A) - \sum_{i=1}^{m} d_i$ holds for $k = 1, \ldots, r$ [6], and the problem with $\delta_k(\tilde{A})$'s reduces to $\delta_k^{\mathrm{LM}}(A)$'s ($r := \mathrm{rank}\, \tilde{A} = \mathrm{rank}\, A - m$).

2.2 Valuated Bimatroid

A valuated bimatroid is a triple (R, C, w), where R and C are disjoint finite sets and $w : 2^R \times 2^C \to \mathbb{R} \cup \{-\infty\}$ is a map satisfying a certain exchange axiom (see, e.g., [1,7]). The value $\rho \in \mathbb{Z}$ denotes the maximum value of $|I| = |J|$ such that $(I, J) \subseteq R \times C$ and $w(I, J) > -\infty$. We define $S_k \subseteq 2^R \times 2^C$ and $\delta_k \in \mathbb{R}$ as follows:

$$S_k = \{(I, J) \mid |I| = |J| = k,\ I \subseteq R,\ J \subseteq C\},$$
$$\delta_k = \max\{w(I, J) \mid (I, J) \in S_k\}.$$

Proposition 2 ([7]) *$\delta_{k-1} + \delta_{k+1} \le 2\delta_k$ holds for $k = 1, 2, \ldots, \rho - 1$.*

The set $\mathcal{M}_k$ is defined as $\{(I, J) \in S_k \mid w(I, J) = \delta_k\}$.

Proposition 3 ([7]) *For any $(I_k, J_k) \in \mathcal{M}_k$ with $1 \le k \le \rho - 1$, there exist $(I_l, J_l) \in \mathcal{M}_l$ $(0 \le l \le \rho, l \ne k)$ such that $I_{l-1} \subseteq I_l$ and $J_{l-1} \subseteq J_l$ $(1 \le l \le \rho)$.*

It is known that $w(I, J) := \deg \det \tilde{A}[I, J]$ defines a valuated bimatroid [1, 7]. Hence, $\{\delta_k(\tilde{A})\}$, as well as $\{\delta_k^{\mathrm{LM}}(A)\}$, is concave by Proposition 2, and the maximizers of w have a nested structure by Proposition 3.

2.3 Combinatorial Relaxation of $\delta_k^{\mathrm{LM}}(A)$

The description of this section is based on [5]. Let $A(x)$ be an LM-polynomial matrix defined by (2). We define $G(A) = (R \cup C, E(A), c)$ as a bipartite graph associated with $A(x)$, where the arc set $E(A)$ and the weight $c : E(A) \to \mathbb{Z}$ are defined as follows:

$$E(A) := \{(i, j) \mid i \in R,\ j \in C,\ A_{ij}(x) \ne 0\}, \qquad (3)$$
$$c(i, j) := \deg A_{ij} \quad ((i, j) \in E(A)). \qquad (4)$$

For a matching M on the bipartite graph $G(A)$, we define $\partial^+ M := \partial M \cap R$ and $\partial^- M := \partial M \cap C$, where ∂M denotes the set of incident vertices of M. Then, let $\hat{\delta}_k^{\mathrm{LM}}(A)$ be the weight of a maximum weight $(m + k)$-matching M such that $R_Q \subseteq \partial^+ M$ in $G(A)$, i.e.,

$$\hat{\delta}_k^{\mathrm{LM}}(A) := \max\left\{ \sum_{e \in M} c(e) \,\middle|\, M : \text{ a matching in } G(A),\ |M| = m + k,\ R_Q \subseteq \partial^+ M \right\}.$$

If there is no $(m + k)$-matching in $G(A)$, we put $\hat{\delta}_k^{\mathrm{LM}}(A) = -\infty$. Then, $\hat{\delta}_k^{\mathrm{LM}}(A)$'s play the role of a combinatorial relaxation of $\delta_k^{\mathrm{LM}}(A)$'s, and $\delta_k^{\mathrm{LM}}(A) \le \hat{\delta}_k^{\mathrm{LM}}(A)$ holds [2]. Moreover, for $R_Q \subseteq I \subseteq R$ and $J \subseteq C$ such that $|I| = |J|$, we define

$$\hat{w}(I, J) := \max\{c(M) \mid M : \text{ a matching in } G(A),\ \partial M = I \cup J\},$$

and for the other case, we define $\hat{w}(I, J) = -\infty$. This $\hat{w}(I, J)$, as well as $w(I, J)$, defines a valuated bimatroid. Therefore, Propositions 2 and 3 hold for $\hat{w}$.

We can test whether $\delta_k^{\mathrm{LM}}(A) = \hat{\delta}_k^{\mathrm{LM}}(A)$ (which we call "tight") holds or not without knowing $\delta_k^{\mathrm{LM}}(A)$ itself by utilizing the duality of linear programming as Proposition 4 below shows. The dual of the linear programming problems associated with the weighted bipartite matching problem discussed above is given as follows:

$$\begin{aligned} \mathrm{DLP}(A, k) : \min. & \sum_{i \in R} p_i + \sum_{j \in C} q_j + (m + k)t \\ \text{s.t. } & p_i + q_j + t \ge c_{ij} \ \ ((i, j) \in E(A)), \\ & p_i \ge 0 \ (i \in R_T), \quad q_j \ge 0 \ (j \in C). \end{aligned}$$

$\mathrm{DLP}(A, k)$ has an integral optimal solution, and the optimal value is equal to $\hat{\delta}_k^{\mathrm{LM}}(A)$. For a feasible solution (p, q, t), we define the active rows $I^* \subseteq R$, the active columns $J^* \subseteq C$, and the tight coefficient matrix A^* as

$$I^* = R_Q \cup \{i \in R_T \mid p_i > 0\}, \quad J^* = \{j \in C \mid q_j > 0\},$$
$$A^*_{ij} = \lim_{x \to \infty} x^{-p_i - q_j - t} A_{ij}(x). \qquad (5)$$

Note that the right-hand side of (5) is a bounded constant because of $p_i + q_j + t \ge c_{ij} = \deg A_{ij}$, and that computing the rank of A^* is relatively easy (but it needs the algorithm for the rank of an LM-matrix [1]).

Proposition 4 ([2]) *Let (p, q, t) be an optimal dual solution. The following three conditions are equivalent:*

- *$\delta_k^{\mathrm{LM}}(A) = \hat{\delta}_k^{\mathrm{LM}}(A)$ holds;*
- *There exist $I \supseteq I^*$ and $J \supseteq J^*$ such that it holds $\mathrm{rank}\, A^*[I, J] = |I| = |J| = m + k$;*
- *The following four conditions hold:*
  - (r1) $\mathrm{rank}\, A^*[R, C] \ge m + k$,
  - (r2) $\mathrm{rank}\, A^*[I^*, C] = |I^*|$,
  - (r3) $\mathrm{rank}\, A^*[R, J^*] = |J^*|$,
  - (r4) $\mathrm{rank}\, A^*[I^*, J^*] \ge |I^*| + |J^*| - (m + k)$.

3. Proposed Algorithm

In this section, we propose an algorithm to compute $\delta_k^{\mathrm{LM}}(A)$'s for an LM-polynomial matrix $A(x)$ defined by (2). For the sake of the algorithm description, let us here suppose that $A(x)$ is a Laurent polynomial matrix. Here, a rational function f is said to be a Laurent polynomial function if there exists an integer N such that $x^N f(x)$ is a polynomial function ($-\mathrm{ord}\, f$ denotes the minimum among such N's). We define

$$d_{\max} = \max_{i,j} \deg A_{ij}, \quad d_{\min} = \min_{i,j} \mathrm{ord}\, A_{ij}.$$

3.1 Theorems to Improve Efficiency

We show two theorems concerning tightness that form the basis of our algorithm. These theorems can be proved similarly as the corresponding theorems in [4].

Theorem 5 *Suppose that $\delta_k^{\mathrm{LM}}(A) = \hat{\delta}_k^{\mathrm{LM}}(A)$ holds and (p,q,t) is a common optimal dual solution of* $\mathrm{DLP}(A,k)$ *and* $\mathrm{DLP}(A,k+1)$. *Then, $\delta_{k+1}^{\mathrm{LM}}(A) = \hat{\delta}_{k+1}^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A) + t$ if and only if* $\operatorname{rank} A^* > m+k$, *where A^* is the tight coefficient matrix defined by* (5).

Theorem 5 allows us to check if $\delta_{k+1}^{\mathrm{LM}}(A) = \hat{\delta}_{k+1}^{\mathrm{LM}}(A)$ by computing $\operatorname{rank} A^*$ only. This value, $r^* := \operatorname{rank} A^*$, is always greater than or equal to $m+k$. Furthermore, when $r^* > m+k+1$, due to the next theorem, we obtain all of $\delta_{k+1}^{\mathrm{LM}}(A), \dots, \delta_{r^*-m}^{\mathrm{LM}}(A)$ at the same time, i.e., we can skip the computation of $\delta_{k+2}^{\mathrm{LM}}(A), \dots, \delta_{r^*-m}^{\mathrm{LM}}(A)$.

Theorem 6 *Under the assumptions of Theorem 5 and $m+k < r^*$, the following equalities and inequality hold*:

$$\delta_l^{\mathrm{LM}}(A) = \hat{\delta}_l^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A) + (l-k)t \quad (k < l \leq r^* - m),$$

$$\delta_l^{\mathrm{LM}}(A) < \delta_k^{\mathrm{LM}}(A) + (l-k)t \quad (l = r^* - m + 1).$$

3.2 The Outline of the Proposed Algorithm

The outline of the proposed algorithm is as follows.

Outline of the Proposed Algorithm

Step 0: Compute $\delta_0^{\mathrm{LM}}(A)$ and set $k := 0$.

Step 1: Find a common optimal dual solution (p,q,t) of $\mathrm{DLP}(A,k)$ and $\mathrm{DLP}(A,k+1)$. If $t < (m+k+1)d_{\min} - \delta_k^{\mathrm{LM}}(A)$ holds, then halt.

Step 2: Test for the tightness, i.e., whether $\delta_{k+1}^{\mathrm{LM}}(A) = \hat{\delta}_{k+1}^{\mathrm{LM}}(A)$ or not, by using (p,q,t) and the tight coefficient matrix A^* (Theorem 5). If the equality holds, go to Step 4. Otherwise, go to Step 3.

Step 3: Modify A to A' satisfying $\hat{\delta}_{k+1}^{\mathrm{LM}}(A') < \hat{\delta}_{k+1}^{\mathrm{LM}}(A)$ and $\delta_{k+1}^{\mathrm{LM}}(A') = \hat{\delta}_{k+1}^{\mathrm{LM}}(A)$, and go back to Step 1.

Step 4: Output $\delta_{k+1}^{\mathrm{LM}}(A), \dots, \delta_{r^*-m}^{\mathrm{LM}}(A)$ (Theorem 6), update $k := r^* - m$ and go back to Step 1.

For Step 0, the initialization, M_0 is defined as the unique matching on submatrix $\operatorname{diag}[x^{d_1}, \dots, x^{d_m}]$ of A ($\delta_0^{\mathrm{LM}}(A) = \sum_{i=1}^m d_i$). We set $I_0^* := R_Q$, $J_0^* := \partial^- M_0$ and $k := 0$. The other steps are discussed in Section 3.3–3.6.

3.3 Step 1: Construction of an Optimal Dual Solution

At every starting point of Step 1, the following conditions hold as a result of the last iteration:

(1) $\deg\det A[I_k^*, J_k^*] = \sum_{(i,j)\in M_k} \deg A_{ij} = \delta_k^{\mathrm{LM}}(A)$;

(2) $R_Q \subseteq I_k^* = \partial^+ M_k$, $J_k^* = \partial^- M_k$.

In actual computation, I_k^* and J_k^* are not necessarily stored because they can be obtained from M_k, which is an $(m+k)$-matching on $G(A)$ achieving $\hat{\delta}_k^{\mathrm{LM}}(A)$. They are explicitly introduced here for a better presentation.

An optimal dual solution (p,q,t) of $\mathrm{DLP}(A,k)$ can be constructed by solving the shortest paths problem on the auxiliary graph G_{M_k} (cf. [5]). In this step, we can adopt "reweighting" using the latest optimal dual variable (see, e.g., [8]).

Lemma 7 is an immediate corollary of [4, Lemma 2].

Lemma 7 *Let (p,q,t) be the optimal dual solution of* $\mathrm{DLP}(A,k)$ *obtained in Step 1. Then, (p,q,t) is an optimal dual solution of* $\mathrm{DLP}(A,k+1)$.

Lemma 7 means that $\hat{\delta}_{k+1}^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A) + t$ holds. Since $\hat{\delta}_{k+1}^{\mathrm{LM}}(A) \geq \delta_{k+1}^{\mathrm{LM}}(A) \geq (m+k+1)d_{\min}$ holds for all integer $k < r$, $\hat{\delta}_{k+1}^{\mathrm{LM}}(A) = \delta_k^{\mathrm{LM}}(A) + t < (m+k+1)d_{\min}$ implies $k = r$. Therefore, if $t < (m+k+1)d_{\min} - \delta_k^{\mathrm{LM}}(A)$ holds, we can set $\operatorname{rank} A = m+k$ and halt.

3.4 Step 2: Test for Tightness

Since Lemma 7 means that (p,q,t) is a common optimal dual solution of $\mathrm{DLP}(A,k)$ and $\mathrm{DLP}(A,k+1)$, we can adopt Theorem 5 instead of Proposition 4 to test for the tightness. Here, we need to compute the rank of an LM-matrix A^*. We execute a slightly different version of the algorithm stated in [1] to ensure Theorem 8 below.

The rank of an LM-matrix can be computed by solving the independent matching problem on a bipartite graph $G = (R_T \cup C_Q, C; E_T \cup E_Q)$, where C_Q denotes the copy of the column-set C and the sets of arcs are defined as $E_T = \{(i,j) \mid i \in R_Q, j \in C, A_{ij}^* \neq 0\}$ and $E_Q = \{(j_Q, j) \mid j \in C\}$. We can solve this problem by utilizing augmenting paths in auxiliary graph $G_M = (V,E)$, where $V = R_T \cup C_Q \cup C$ and $E = E_T \cup E_Q \cup E^+ \cup M^\circ$. Here, E^+ represents the structure of the linear matroid with respect to $A^*[R_Q, C]$. The difference from [1] appears only in the step of initialization.

(i) We set $\mathit{base}[i] = j$ for $(i,j) \in M_k$ such that $i \in R_Q$, and also set $M^\circ := \{(j,i) \mid (i,j) \in M_k, i \in R_T\} \cup \{(j, j_Q) \mid i \in R_Q, \mathit{base}[i] = j\}$.

(ii) Then, we construct a constant matrix P by repeating the following procedure for $i = 1, 2, \dots, m$: we choose $(i,j) \in M_k \cap (R_Q \times C)$ in descending order of p_i, and conduct the row elimination for all rows taking A_{ij}^* as the pivot. At the end of the procedure, we obtain a constant matrix P such that

$$P = \begin{bmatrix} I_m & U \end{bmatrix} = SA^*[R_Q, C] \tag{6}$$

holds, where S is a nonsingular constant matrix which represents the row eliminations (U is a constant matrix created by the procedure).

After the procedure (i) and (ii), we execute the algorithm for the rank of LM-matrices [1]. At the end of this algorithm, P satisfies

$$\operatorname{rank}\begin{bmatrix} P \\ A^*[R_T, C] \end{bmatrix} = \text{term-rank}\begin{bmatrix} P \\ A^*[R_T, C] \end{bmatrix}, \tag{7}$$

where term-rank A is defined as the maximum matching on $G(A)$ (see, e.g., [1]).

3.5 Step 3: Matrix Modification

At the beginning of Step 3, the relation (7) holds and we have a constant matrix S satisfying (6). Then, we define $\tilde{S}(x)$ and $S(x)$ as follows:

$$\tilde{S}(x) = \begin{bmatrix} S(x) & O \\ O & I \end{bmatrix} = \operatorname{diag}(x; p_R)\begin{bmatrix} S & O \\ O & I \end{bmatrix}\operatorname{diag}(x; -p_R).$$

We modify the matrix $A(x)$ to $A'(x)$ as follows:

$$A'(x) = \tilde{S}(x)A(x) = \begin{bmatrix} S(x)A[R_Q, C](x) \\ A[R_T, C](x) \end{bmatrix}. \tag{8}$$

This modification makes sense, as stated in Theorem 8.

Theorem 8 *The matrix* $A'(x)$ *defined in* (8) *has the following four properties*: (1) $\delta_l^{\mathrm{LM}}(A') = \delta_l^{\mathrm{LM}}(A)$ *for* $l = 0, 1, \ldots, r$; (2) $\deg\det A'[I_k^*, J_k^*] = \delta_k^{\mathrm{LM}}(A')$; (3) $\hat{\delta}_k^{\mathrm{LM}}(A') = \delta_k^{\mathrm{LM}}(A')$; (4) $\hat{\delta}_{k+1}^{\mathrm{LM}}(A') < \hat{\delta}_{k+1}^{\mathrm{LM}}(A)$.

Proof (1) It is sufficient to show that $\tilde{S}(x)$ is biproper. This claim holds by construction of S.
(2) By construction of A', we obtain

$$\deg\det A'[R_Q \cup I_k^*, J_k^*] = \deg\det A[R_Q \cup I_k^*, J_k^*].$$

Since the right-hand side is equal to $\delta_k^{\mathrm{LM}}(A)$, this equality means that the property (2) holds.
(3) An optimal solution (p, q, t) of $\mathrm{DLP}(A, k)$ is a feasible solution of $\mathrm{DLP}(A', k)$. Here, A^* and A'^* denote the tight coefficient matrices of A and A' with respect to (p, q, t). Then, the following equality holds:

$$A'^* = \begin{bmatrix} S & O \\ O & I \end{bmatrix} A^*.$$

This means that $\text{term-rank}\, A'^*[I_k^*, J_k^*] = m + k$ holds. Hence, by [5, Lemma 5], (p, q, t) is an optimal solution of $\mathrm{DLP}(A', k)$. Moreover, (7) and Proposition 4 mean that $\hat{\delta}_k^{\mathrm{LM}}(A') = \delta_k^{\mathrm{LM}}(A')$.
(4) We show $\text{term-rank}\, A'^*[R, C] = \operatorname{rank} A'^*[R, C] = m + k$ in the discussion above. Hence, (p, q, t) is not optimal in DLP $(A', k+1)$ by [5, Lemma 5]. **(QED)**

3.6 Step 4: Outputs and Updates

Recall that the task of Step 4 is to output $\delta_l^{\mathrm{LM}}(A)$'s $(l = k+1, \ldots, r^* - m)$ in view of Theorem 6, and then we go back to Step 1. But in order to start the process of Step 1, we need the corresponding matching $M = M_{r^*-m}$ such that

$$\sum_{(i,j)\in M} \deg A_{ij} = \deg\det A[\partial^+ M, \partial^- M] = \delta_{r^*-m}^{\mathrm{LM}}(A)$$

holds. The key ingredient for obtaining this is the computation of $I_{r^*-m}^* = \partial^+ M_{r^*-m}$ and $J_{r^*-m}^* = \partial^- M_{r^*-m}$, which can be obtained simultaneously in the calculation of $\operatorname{rank} A^*$ described in Section 3.4 as $I_{r^*-m}^* = \partial M^\circ \cap R$, $J_{r^*-m}^* = \partial M^\circ \cap C$. Then, M_{r^*-m} is a maximum weight bipartite matching on $G = (I_{r^*-m}^* \cup J_{r^*-m}^*, E, \gamma)$, where $E = E(A) \cap (I_{r^*-m}^* \times J_{r^*-m}^*)$. We can obtain M_{r^*-m} efficiently by augmenting paths.

4. Complexity Analysis

The time complexity of the proposed algorithm can be analyzed as in Theorem 9, following the same line of discussion in Iwata–Takamatsu [5]. This theorem can be proved similarly as [4, Theorem 4].

Theorem 9 *The proposed algorithm runs in* $O(dnr(n + m^2 + dm^{\omega-1}r))$ *time, where* $d := d_{\max} - d_{\min}$.

This complexity of the proposed algorithm is almost the same as in the naive algorithm, which simply repeats the algorithm [5] for a specified order (even a little bit worse than the naive one in special cases, e.g., very low rank case $m^{3-\omega} > dr$). However, since the step of modification is seldom executed [5, Proposition 1], the time complexity of the computation for $\operatorname{rank} A^*$ is crucial. As stated in Proposition 10 below, The number of iterations of Step 2 is only $(O(\sqrt{dr}) + s)$, whereas the naive one computes the rank $4(r + s)$ times, where s denotes the number of modifications. Hence, the proposed algorithm might be faster if d and s are sufficiently small.

Proposition 10 *Let* $A(x)$ *be an LM-Laurent polynomial matrix in the form of* (2). *Then, the proposed algorithm executes Step 2 at most* $(f(d, r) + s)$ *times, where* s *denotes the number of modifications and* f *is defined as*

$$f(d, r) = \begin{cases} \left\lfloor \frac{-1+\sqrt{8dr+1}}{2} \right\rfloor & (r \geq 2d - 1) \\ r & (r < 2d - 1) \end{cases}. \tag{9}$$

Proof If t denotes the number of iterations of Step 4, Step 2 is executed $(t + s)$ times. By the latter part of Theorem 6, t is same as the number of k's such that $\delta_{k+1}^{\mathrm{LM}}(A) - \delta_k^{\mathrm{LM}}(A) \neq \delta_k^{\mathrm{LM}}(A) - \delta_{k-1}^{\mathrm{LM}}(A)$ holds, which is less than $f(d, r)$. Here, $f(d, r)$ is upper bound of the maximum number of $n(\{a_k\})$ among $\{a_k\} \in X_d^r$, where $X_d^r \subseteq \mathbb{Z}^{r+1}$ is the set of sequences $\{a_k\}$ such that $a_0 = 0$, $a_r \geq 0$ and $a_{k+1} - a_k \leq a_k - a_{k-1} \leq d$ $(k = 1, \ldots, r-1)$, and $n(\{a_k\})$ is the number of k's such that $a_{k+1} - a_k \neq a_k - a_{k-1}$. To see this, we define N as an upper bound of the maximum number of $n(\{a_k\})$ among $\{a_k\} \in X_d^r$ and the sequence $\{b_k^\gamma\}$ for an integer γ as follows: $b_k^\gamma = dk$ for $k = 0, 1, \ldots, \gamma-1$ and $b_k^\gamma = d(\gamma-1) + \sum_{i=\gamma}^k (d - (i - \gamma + 1))$ for $k = \gamma, \ldots, r$. Then, for any $\{a_k\} \in X_d^r$, there exists $\{b_k^\gamma\} \in X_d^r$ such that $n(\{a_k\}) = n(\{b_k^\gamma\})$ for $\gamma = r + 1 - n(\{a_k\})$. Hence, we only have to deal with $\{b_k^\gamma\}$'s and we need to consider whether $b_r^\gamma \geq 0$ or not for some γ. Therefore, N is equal to the maximum value of $r - \gamma + 1$ such that $b_r^\gamma = d(\gamma - 1) + \sum_{i=\gamma}^r (d - (i - \gamma + 1)) \geq 0$. By introducing $\alpha := r - \gamma + 1$, we obtain the equivalent problem: maximize α subject to $\alpha^2 + \alpha - 2dr \leq 0$ and $1 \leq \alpha \leq r$. It means that $N = f(d, r)$. **(QED)**

Acknowledgments

The author is grateful to Kazuo Murota for valuable comments and also for anonymous reviewer's comments.

References

[1] K. Murota, Matrices and Matroids for Systems Analysis, Springer, Berlin, 2000.

[2] K. Murota, Combinatorial relaxation algorithm for the maximum degree of subdeterminants: computing Smith–McMillan form at infinity and structural indices in Kronecker form, Appl. Algebra Engrg. Comm. Comput., **6** (1995), 251–273.

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

[4] S. Sato, Combinatorial relaxation algorithm for the entire sequence of the maximum degree of minors, METR2014-23, (2014), `http://www.keisu.t.u-tokyo.ac.jp/research/techrep/data/2014/METR14-23.pdf`.

[5] S. Iwata and M. Takamatsu, Computing the maximum degree of minors in mixed polynomial matrices via combinatorial relaxation, Algorithmica, **66** (2013), 346–368.

[6] K. Murota, On the degree of mixed polynomial matrices, SIAM J. Matrix Anal. Appl., **20** (1999), 196–227.

[7] K. Murota, Finding optimal minors of valuated bimatroids, Appl. Math. Lett., **8** (1995), 37–42.

[8] N. Tomizawa, On some techniques useful for solution of transportation network problems, Networks, **1** (1971), 173–194.