Japan Journal of Industrial and Applied Mathematics (2020) 37:677–696
https://doi.org/10.1007/s13160-020-00413-3

ORIGINAL PAPER

On a weighted linear matroid intersection algorithm by Deg-Det computation

Hiroki Furue[1] · **Hiroshi Hirai**[1]

Received: 30 August 2019 / Revised: 13 February 2020 / Published online: 3 March 2020

Abstract
In this paper, we address the weighted linear matroid intersection problem from computation of the degree of the determinant of a symbolic matrix. We show that a generic algorithm computing the degree of noncommutative determinants, proposed by the second author, becomes an $O(mn^3 \log n)$ time algorithm for the weighted linear matroid intersection problem, where two matroids are given by column vectors of $n \times m$ matrices A, B. We reveal that our algorithm is viewed as a "nonstandard" implementation of Frank's weight splitting algorithm for linear matroids. This gives a linear algebraic reasoning to Frank's algorithm. Although our algorithm is slower than existing algorithms in the worst case estimate, it has a notable feature. Contrary to existing algorithms, our algorithm works on different matroids represented by another "sparse" matrices A^0, B^0, which skips unnecessary Gaussian eliminations for constructing residual graphs.

Keywords Combinatorial optimization · Polynomial time algorithm · Weighted matroid intersection · The degree of determinant · Weight splitting

Mathematics Subject Classification 68W40

1 Introduction

Several basic combinatorial optimization problems have linear algebraic formulations. It is classically known [2] that the maximum cardinality of a matching in a bipartite graph $G = (U, V; E)$ with color classes $U = [n], V = [n']$ is equal to the rank of the matrix $A = \sum_{e \in E} A_e x_e$, where x_e $(e \in E)$ are variables and A_e is an $n \times n'$

✉ Hiroki Furue
hiroki_furue@mist.i.u-tokyo.ac.jp

Hiroshi Hirai
hirai@mist.i.u-tokyo.ac.jp

1 Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan

matrix with $(A_e)_{ij} := 1$ if $e = ij$ and zero otherwise. Such a rank interpretation is known for the linear matroid intersection, nonbipartite matching, and linear matroid matching problems; see [13].

The degree of the determinant of a polynomial (or rational) matrix is a weighted counter part of rank, and can formulate weighted versions of combinatorial optimization problems. The maximum weight perfect matching problem in a bipartite graph $G = ([n], [n]; E)$ with integer weights c_e ($e \in E$) corresponds to computing the degree $\deg_t \det A(t)$ of the determinant of the (rational) matrix $A(t) := \sum_{e \in E} A_e x_e t^{c_e}$. Again, the weighted linear matroid intersection, nonbipartite matching, and linear matroid matching problems admit such formulations.

Inspired by the recent advance [6, 10] of a noncommutative approach to symbolic rank computation, the second author [8] introduced the problem of computing the degree $\deg_t \mathrm{Det} A(t)$ of the *Dieudonné determinant* $\mathrm{Det} A(t)$ of a matrix $A(t) = \sum_i A_i(t) x_i$, where x_i are pairwise noncommutative variables and $A_i(t)$ is a rational matrix with commuting variable t. He established a general min-max formula for $\deg_t \mathrm{Det} A(t)$, presented a conceptually simple and generic algorithm, referred here to as **Deg-Det**, for computing $\deg_t \mathrm{Det} A(t)$, and showed that $\deg_t \det A(t) = \deg_t \mathrm{Det} A(t)$ holds if $A(t)$ corresponds to an instance of the weighted linear matroid intersection problem. In particular, **Deg-Det** gives rise to a pseudo-polynomial time algorithm for the weighted linear matroid intersection problem. In the first version of the paper [8], the second author asked (i) whether **Deg-Det** can be a (strongly) polynomial time algorithm for the weighted linear matroid intersection, and (ii) how **Deg-Det** is related to the existing algorithms for this problem. He pointed out some connection of **Deg-Det** to the primal-dual algorithm by Lawler [15] but the precise relation was not clear.

The main contribution of this paper is to answer the questions (i) and (ii):

- We show that **Deg-Det** becomes an $O(nm^3 \log n)$ time algorithm for the weighted linear matroid intersection problem, where the two matroids are represented and given by two $n \times m$ matrices A, B. This answers affirmatively the first question.
- For the second question, we reveal the relation between our algorithm and the *weight splitting algorithm* by Frank [4]. This gives a linear algebraic reasoning to Frank's algorithm.

  We show that the behavior of our algorithm is precisely the same as that of a slightly modified version of Frank's algorithm. However our algorithm is rather different from the standard implementation of Frank's algorithm for linear matroids. This relationship was unexpected and nontrivial for us, since the two algorithms look quite different.

Although our algorithm is slower than the standard $O(mn^3)$-time implementation of Frank's algorithm in the worst case estimate, it has a notable feature. Frank's algorithm works on a subgraph $\bar{G}_X$ of the residual graph G_X for a common independent set X, where G_X is determined by Gaussian elimination for A, B and $\bar{G}_X$ is determined by a splitting of the weight. On the other hand, our algorithm does not compute the residual graph G_X but computes a non-redundant subgraph G_X^0 of $\bar{G}_X$, which is the residual graph of different matroids represented by another "sparse" matrices A^0, B^0. Consequently, our algorithm applies fewer elimination operations than the standard one, which will be a practical advantage.

Related work. The essence of **Deg-Det** comes from the *combinatorial relaxation algorithm* by Murota [11], which is an algorithm computing the degree of the (ordinary) determinant of a polynomial/rational matrix; see [12, Sect. 7.1].

Several algorithms have been proposed for the general weighted matroid intersection problem under the independence oracle model; see e.g., [16, Sect. 41.3] and the references therein. For linear matroids given by two $n \times m$ matrices, the current fastest algorithms (as far as we know) are an $O(mn^{\omega})$-time implementation of Frank's algorithm using fast matrix multiplication and an $O(nm^{\frac{7-\omega}{5-\omega}} \log^{\frac{\omega-1}{5-\omega}} n \log mC)$-time algorithm by Gabow and Xu [5], where C is the maximum absolute value of weights c_i and $\omega \in [2, 2.37]$ denotes the exponent of the time complexity of matrix multiplication. Huang, Kakimura, and Kamiyama [9] gave an $O(nm \log n_* + Cmn_*^{\omega-1})$-time algorithm, where n_* is the maximum size of a common independent set. This algorithm is currently fastest for the case of small C.

For unweighted linear matroid intersection, Cunningham [1] showed that the classical Edmonds' algorithm runs in $O(mn^2 \log n)$ time. Harvey [7] gave a randomized $O(mn^{\omega-1})$-time algorithm. His algorithm also treats the problem as the rank computation of a matrix with variables x_i, and uses random substitution of the variables and fast matrix multiplication.

Organization. The rest of this paper is organized as follows. In Sect. 2, we introduce algorithm **Deg-Det**, and describe basics of the unweighted (linear) matroid intersection problem from a linear algebraic viewpoint; our algorithm treats the unweighed problem as a subproblem. In Sect. 3, we first formulate the weighted linear matroid intersection problem as the degree of the determinant of a rational matrix A, and show that **Deg-Det** computes $\deg_t \det A$ correctly. Then we present our algorithm by specializing **Deg-Det**, analyze its time complexity, and reveal its relationship to Frank's algorithm.

In this paper, we deal with linear matroids represented over the field of rationals but our augment and algorithm work on an arbitrary field.

2 Preliminaries

2.1 Notation

Let $\mathbb{Q}$ and $\mathbb{Z}$ denote the sets of rationals and integers, respectively. Let $\mathbf{0} \in \mathbb{Q}^n$ denote the zero vector. For $I \subseteq [n] := \{1, 2, ..., n\}$, let $\mathbf{1}_I \in \mathbb{Q}^n$ denote the characteristic vector of I, that is, $(\mathbf{1}_I)_k := 1$ if $k \in I$ and 0 otherwise. Here, $\mathbf{1}_{[n]}$ is simply denoted by $\mathbf{1}$.

For a polynomial $p = \sum_{i=0}^{k} a_i t^i \in \mathbb{Q}[t]$ with $a_k \neq 0$, the degree $\deg_t p$ with respect to t is defined as k. The degree $\deg_t p/q$ of a rational function $p/q \in \mathbb{Q}(t)$ with polynomials $p, q \in \mathbb{Q}[t]$ is defined as $\deg_t p - \deg_t q$. The degree of zero polynomial is defined as $-\infty$.

A rational function p/q is called *proper* if $\deg_t p/q \leq 0$. A rational matrix $Q \in \mathbb{Q}(t)^{n\times m}$ is called proper if each entry of Q is proper. For a proper rational matrix $Q \in \mathbb{Q}(t)^{n\times m}$, there is a unique matrix over $\mathbb{Q}$, denoted by Q^0, such that

$$Q = Q^0 + t^{-1}Q',$$

where Q' is some proper matrix.

For an integer vector $\alpha \in \mathbb{Z}^n$, let (t^α) denote the $n \times n$ diagonal matrix having diagonals $t^{\alpha_1}, t^{\alpha_2}, \ldots, t^{\alpha_n}$ in order, that is,

$$(t^\alpha) = \begin{pmatrix} t^{\alpha_1} & & & \\ & t^{\alpha_2} & & \\ & & \ddots & \\ & & & t^{\alpha_n} \end{pmatrix}.$$

For a matrix $A \in \mathbb{Q}^{n\times m}$ and $J \subseteq [m]$, let $A[J]$ denote the submatrix of A consisting of the j-th columns for $j \in J$. Additionally, for $I \subseteq [n]$, let $A[I, J]$ denote the submatrix of A consisting of the (i, j)-entries for $i \in I, j \in J$.

2.2 Algorithm Deg-Det

Given $n \times n$ rational matrices $M_1, M_2, \ldots, M_m \in \mathbb{Q}(t)^{n\times n}$, consider the following matrix

$$M := M_1x_1 + M_2x_2 + \cdots + M_mx_m \quad \in \mathbb{Q}(t, x_1, x_2, \ldots, x_m),$$

where $x_1, x_2, \ldots, x_m$ are variables and M is regarded as a multivariate rational matrix with (pairwise commutative) variables $t, x_1, x_2, \ldots, x_m$. We address the computation of the degree of the determinant of M with respect to t.

Consider the following optimization problem:

$$\begin{aligned} \text{(P)} \quad \text{Max.} \ & \deg_t \det P + \deg_t \det Q \\ \text{s.t.} \ & PMQ : \ \text{proper}, \\ & P, Q \in \mathbb{Q}(t)^{n\times n} : \ \text{nonsingular}. \end{aligned}$$

This problem gives an upper bound of $\deg_t \det M$. Indeed, if PMQ is proper, then $\deg_t \det PMQ \leq 0$, and $\deg_t \det M \leq -\deg_t \det P - \deg_t \det Q$. In fact, it is shown [8] that the optimal value of (P) is interpreted as the negative of the degree of the *Dieudonné determinant* of M for the case where $x_1, x_2, \ldots, x_m$ are pairwise noncommutative variables.

The following algorithm for (P) is due to [8], which is viewed as a simplification of the *combinatorial relaxation algorithm* by Murota [11]; see also [12, Sect. 7.1].

Algorithm: Deg-Det
Input: $M = M_1x_1 + M_2x_2 + \cdots + M_mx_m$, where $M_i \in \mathbb{Q}(t)^{n\times n}$ for $i \in [m]$.

Output: An upper bound of $\deg_t \det M$ (the negative of the optimal value of (P)).
0: Let $P := t^{-d}I$ and $Q := I$, where d is the maximum degree of entries in M. Let $D^* := nd$.
1: Solve the following problem:

$$(\mathrm{P}^0) \quad \mathrm{Max.}\, r+s \quad \text{s.t.}\, K(PMQ)^0 L \text{ has an } r\times s \text{ zero submatrix}, \quad K, L \in \mathbb{Q}^{n\times n} : \text{ nonsingular},$$

and obtain optimal matrices K, L; recall the notation $(\cdot)^0$ in Sect. 2.1.
2: If the optimal value $r+s$ is at most n, then stop and output D^*.
3: Let I and J be the sets of row and column indices, respectively, of the $r\times s$ zero submatrix of $K(PMQ)^0L$. Find the maximum integer $\kappa (\geq 1)$ such that $(t^{\kappa \mathbf{1}_I})KPMQL(t^{-\kappa \mathbf{1}_{[n]\setminus J}})$ is proper.
If κ is unbounded, then output $-\infty$. Otherwise, let $P \leftarrow (t^{\kappa \mathbf{1}_I})KP$, $Q \leftarrow QL(t^{-\kappa \mathbf{1}_{[n]\setminus J}})$ and $D^* \leftarrow D^* - \kappa(r+s-n)$. Go to step 1.

Observe that in each iteration (P, Q) is a feasible solution of (P), and D^* equals $-\deg_t \det P - \deg_t \det Q$. Thus, (P) gives an upper bound of $\deg_t \det M$. We are interested in the case where the algorithm outputs $\deg_t \det M$ correctly.

Lemma 2.1 ([8]) *In step* 2 *of* **Deg-Det**, *the following holds*:

(1) *If* $r+s>n$, *then* $(PMQ)^0$ *is singular over* $\mathbb{Q}(x_1, x_2, \ldots, x_m)$.
(2) *If* $(PMQ)^0$ *is nonsingular, then* $D^* = \deg_t \det M$.

Proof (1). It is obvious that any $n\times n$ matrix is singular if it has an $r\times s$ zero submatrix with $r+s>n$.

(2). PMQ is written as $(PMQ)^0 + t^{-1}N$ for some proper N. If $(PMQ)^0$ is nonsingular, then $\deg_t \det PMQ = \deg_t \det (PMQ)^0 = 0$, and hence $\deg_t \det M = -\deg_t \det P - \deg_t \det Q = D^*$. □

2.3 Algebraic formulation for linear matroid intersection

Let $A = (a_1\ a_2\ \cdots\ a_m)$ be an $n\times m$ matrix over $\mathbb{Q}$. Let $\mathbf{M}(A) = ([m], \mathcal{I}(A))$ denote the linear matroid represented by A. Specifically, the ground set of the matroid $\mathbf{M}(A)$ is the set $[m]$ of the column indices, and the family $\mathcal{I}(A)$ of independent sets of $\mathbf{M}(A)$ consists of all subsets $X \subseteq [m]$ such that the corresponding column vectors a_i $(i\in X)$ are linearly independent. Let $\rho_A : 2^{[m]} \to \mathbb{Z}$ denote the rank function of $\mathbf{M}(A)$, that is, $\rho_A(X) := \max\{|Y| \mid Y \in \mathcal{I}(A), Y \subseteq X\}$. A minimal (linearly) dependent subset is called a circuit. See, e.g., [16, Chapter 39] for basics on matroids.

Suppose that we are given another $n \times m$ matrix $B = (b_1\ b_2\ \cdots\ b_m) \in \mathbb{Q}^{n\times m}$. Let $\mathbf{M}(B) = ([m], \mathcal{I}(B))$ be the corresponding linear matroid. A common independent set of $\mathbf{M}(A)$ and $\mathbf{M}(B)$ is a subset $X \subseteq [m]$ such that X is independent for both $\mathbf{M}(A)$ and $\mathbf{M}(B)$. The linear matroid intersection problem is to find a common independent set of the maximum cardinality. To formulate this problem linear algebraically, define an $n \times n$ matrix $M = M(A, B)$ over $\mathbb{Q}(x_1, x_2, \dots, x_m)$ by

$$M := \sum_{i=1}^{m} a_i b_i^\top x_i,$$

where $x_1, x_2, \dots, x_m$ are variables. The following is the matroid intersection theorem and its linear algebraic sharpening.

Theorem 2.2 ([3]; see also [13, 17]) *The following quantities are equal*:

(1) *The maximum cardinality of a common independent set of* $\mathbf{M}(A)$ *and* $\mathbf{M}(B)$.
(2) The minimum of $\rho_A(J) + \rho_B([m]\setminus J)$ over $J \subseteq [m]$.
(1') rank$M$.
(2') $2n$ *minus the maximum of* $r + s$ *such that* KML *has an* $r \times s$ *zero submatrix for some nonsingular matrices* $K, L \in \mathbb{Q}^{n\times n}$.

Sketch of Proof (1) = (2) is nothing but the matroid intersection theorem.

(1) = (1′). A $k \times k$ submatrix M' of M is represented by $M' = A'DB'^\top$, where A', B' are $k \times m$ submatrices of A, B, and D is the diagonal matrix with diagonals $x_1, x_2, \dots, x_m$ (in order). From Binet-Cauchy formula, we see that $\det M' \neq 0$ if and only if there is a k-element subset $X \subseteq [m]$ such that $\det A'[X] \det B'[X] \neq 0$. Thus, rank$M \geq k$ if and only if there is a common independent set of cardinality k.

(2) ≥ (2′). Take a basis $u_1, u_2, \dots, u_r$ of the orthogonal complement of the vector space spanned by $\{a_i \mid i \in J\}$, and extend it to a basis $u_1, u_2, \dots, u_n$ of $\mathbb{Q}^n$, where $r = n - \rho_A(J)$. Similarly, take a basis $v_1, v_2, \dots, v_n$ of $\mathbb{Q}^n$ that contains a basis $v_1, v_2, \dots, v_s$ of the orthogonal complement of the vector space spanned by $\{b_i \mid i \in [m]\setminus J\}$, where $s = n - \rho_B([m]\setminus J)$. Then $u_k^\top a_i b_i^\top v_\ell = 0$ for all $k \in [r]$, $\ell \in [s]$, and $i \in [m]$. This means that KML has an $r \times s$ zero submatrix for $K = (u_1\ u_2\ \cdots\ u_n)^\top$ and $L = (v_1\ v_2\ \cdots\ v_n)$.

(2′) ≥ (1′). If KML has an $r \times s$ zero submatrix, then rank$M$ = rank$KML \leq n - r + n - s$. □

Let us briefly explain Edmonds' algorithm to obtain a common independent set of the maximum cardinality. For any common independent set X, the auxiliary (di)graph $G_X = G_X(A, B)$ is defined as follows. The set $V(G_X)$ of nodes of G_X is equal to the ground set $[m]$ of the matroids, and the set $E(G_X)$ of arcs is given by: $(i, j) \in E(G_X)$ if and only if one of the following holds:

- $i \in X$, $j \notin X$, and i, j belong to a circuit of $\mathbf{M}(A)$.
- $i \notin X$, $j \in X$, and i, j belong to a circuit of $\mathbf{M}(B)$.

Let $S_X = S_X(A)$ denote the subset of nodes $i \in E\backslash X$ such that $X \cup \{i\}$ is independent in $\mathbf{M}(A)$, and $T_X = T_X(B)$ denote the subset of nodes $i \in E\backslash X$ such that $X \cup \{i\}$ is independent in $\mathbf{M}(B)$. See Fig. 1 for G_X, S_X, and T_X.

Lemma 2.3 ([3]) *Let X be a common independent set, and let R be the set of nodes reachable from S_X in G_X.*

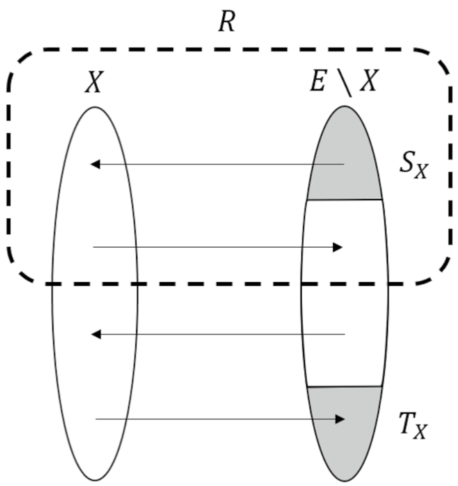
(1) *Suppose that $R \cap T_X \neq \emptyset$. For a shortest path P from S_X to T_X, the set $X \triangle V(P)$ is a common independent set with $|X \triangle V(P)| = |X| + 1$.*
(2) *Suppose that $R \cap T_X = \emptyset$. Then X is a maximum common independent set and R attains* $\min_{J \subseteq [m]} \rho_A(J) + \rho_B([m]\backslash J)$.

Here $\triangle$ denotes the symmetric difference. According to this lemma, Edmonds' algorithm is as follows:

- Find a shortest path P in G_X from S_X to T_X (by BFS).
- If it exists, then replace X by $X \triangle V(P)$, and repeat. Otherwise, X is a common independent set of the maximum cardinality.

In our case, the auxiliary graph G_X and optimal matrices K, L in (2′) are naturally obtained by applying elementary row operation to matrices A, B as follows. Since X is a common independent set, both $A[X]$ and $B[X]$ have column full rank $|X|$. Therefore, by multiplying nonsingular matrices K and L to A and B from left, respectively, we can make A and B diagonal in the position X, that is, for some injective maps $\sigma_A, \sigma_B : X \to [n]$, it holds $(KA)_{\sigma_A(i)i} = (LB)_{\sigma_B(i)i} = 1$ for $i \in X$ and other elements are zero. Incorporating permutation matrices in K, L, we can assume $\sigma_A = \sigma_B = \sigma$. Such matrices KA and LB are said to be *X-diagonal*. Notice

Fig. 1 The auxiliary graph G_X



that these operations do not change the matroids $\mathbf{M}(A)$ and $\mathbf{M}(B)$. See Fig. 2, where the columns and rows are permuted appropriately.

Then the auxiliary graph G_X is constructed from the nonzero patterns of KA and LB as follows. S_X (resp. T_X) consists of nodes i with $(KA)_{ki} \neq 0$ (resp. $(LB)_{ki} \neq 0$) for some $k \in [n] \backslash \sigma(X)$, where $\sigma(X) = \{j \in [n] \mid \exists i \in X, \sigma(i) = j\}$. Additionally, for $i \in X$, arc (i, j) (resp. (j, i)) exists if and only if $j \notin S_X$ and $(KA)_{\sigma(i)j} \neq 0$ (resp. $j \notin T_X$ and $(LB)_{\sigma(i)j} \neq 0$).

Moreover, in the case where $R \cap T_X = \emptyset$, the matrices $K, L^\top$ attain the maximum in (2′). Indeed, define I^*, J^*, I and J by

$$I^* := [n] \backslash \sigma(X), \tag{2.1}$$

$$J^* := [n] \backslash \sigma(X), \tag{2.2}$$

$$I := \sigma(R \cap X) \cup I^*, \tag{2.3}$$

$$J := \sigma(X \backslash R) \cup J^*. \tag{2.4}$$

Then the submatrix $(KML^\top)[I, J]$ is an $(n - |X \backslash R|) \times (n - |R \cap X|)$ zero submatrix, where $|X| = 2n - (n - |X \backslash R| + n - |R \cap X|)$. See Fig. 3.

3 Algorithm

In this section, we consider the weighted linear matroid intersection problem. In Sect. 3.1, we formulate the problem as the computation of the degree of the determinant of a rational matrix associated with given two linear matroids and

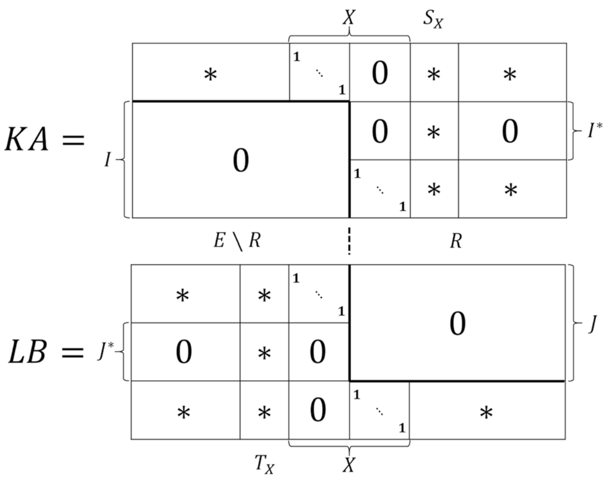


Fig. 2 Matrices A, B after elimination

$$KML^{\top} = KAD(LB)^{\top} =$$

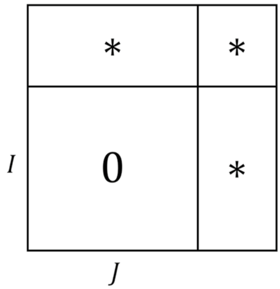


Fig. 3 $KML^{\top}$ has zero submatrix $KML^{\top}[I, J]$, where D is the diagonal matrix with diagonals $x_1, x_2, \ldots, x_n$

a weight. In Sect. 3.2, we specialize **Deg-Det** to present our algorithm for the weighted linear matroid intersection problem. Its time complexity is analyzed in Sect. 3.3, and its relation to Frank's algorithm is discussed in Sect. 3.4.

3.1 Algebraic formulation of weighted linear matroid intersection

Let A, B be $n \times m$ matrices over $\mathbb{Q}$ as in Sect. 2.3, and let $\mathbf{M}(A)$ and $\mathbf{M}(B)$ be the associated linear matroids on $[m]$. We assume that both A and B have no zero columns. In addition to A, B, we are further given integer weights $c_i \in \mathbb{Z}$ for $i \in [m]$. The goal of the weighted linear matroid intersection problem is to maximize the weight $c(X) := \sum_{i \in X} c_i$ over all common independent sets X.

Here we consider a restricted situation when the maximum is taken over all common independent sets of cardinality n. In this case, the maximum weight is interpreted as the degree of the determinant of the following $n \times n$ rational matrix M defined by

$$M := \sum_{i=1}^{m} a_i b_i^{\top} x_i t^{c_i}.$$

Lemma 3.1 *Suppose that A and B have row full rank. The $\deg_t \det M$ is equal to the maximum of the weight $c(X)$ over all common independent sets X of cardinality n.*

Proof As in the proof of Theorem 2.2, by Binet-Cauchy formula applied to M, we obtain $\det M = \sum_{X \subseteq [m] : |X| = n} \det A[X] \det B[X] t^{c(X)} \prod_{i \in X} x_i$, and

$$\deg_t \det M = \max\{c(X) \mid X \subseteq [m] \,:\, \det A[X] \det B[X] \neq 0\}.$$

□

Lemma 3.2 ([8]) *For the setting $M_i := a_i b_i^{\top} t^{c_i}$ ($i \in [m]$), the algorithm* **Deg-Det** *outputs* $\deg_t \det M$.

Proof Consider step 2 of **Deg-Det**. Here $(PM_iQ)^0$ is written as $a_i^0 b_i^{0\top}$ for some $a_i^0, b_i^0 \in \mathbb{Q}^n$; see (3.2) and (3.3) in the next subsection. In particular, $(PMQ)^0 = \sum_{i=1}^m a_i^0 b_i^{0\top} x_i$. Therefore, by Theorem 2.2, $(PMQ)^0$ is nonsingular if and only if the optimal value $r + s$ of (P^0) is at most n. Thus, if the algorithm terminates, then $(PMQ)^0$ is nonsingular and $D^* = \deg_t \det M$ by Lemma 2.1. □

3.2 Algorithm description

Here we present our algorithm by specializing **Deg-Det**. The basic idea is to apply Edmonds' algorithm to solve the problem (P^0) for $(PMQ)^0 = \sum_{i=1}^m (PM_iQ)^0 x_i$, where PMQ is proper. We first consider the case where P and Q are diagonal matrices represented as $P = (t^\alpha)$ and $Q = (t^\beta)$ for some $\alpha, \beta \in \mathbb{Z}^n$. In this case, $(PMQ)^0$ is explicitly written as follows. Observe that the properness of PMQ is equivalent to

$$\alpha_k + \beta_\ell + c_i \le 0 \quad (i \in [m], k, \ell \in [n] : (a_i)_k (b_i)_\ell \ne 0). \tag{3.1}$$

For $i \in [m]$, define $a_i^0, b_i^0 \in \mathbb{Q}^n$ by

$$(a_i^0)_k := \begin{cases} (a_i)_k & \text{if } \exists \ell \in [n], (a_i)_k (b_i)_\ell \ne 0, \alpha_k + \beta_\ell + c_i = 0, \\ 0 & \text{otherwise}, \end{cases} \tag{3.2}$$

$$(b_i^0)_\ell := \begin{cases} (b_i)_\ell & \text{if } \exists k \in [n], (a_i)_k (b_i)_\ell \ne 0, \alpha_k + \beta_\ell + c_i = 0, \\ 0 & \text{otherwise}. \end{cases} \tag{3.3}$$

Then $(PM_iQ)^0 = a_i^0 b_i^{0\top}$. Namely we have

$$(PMQ)^0 = \sum_{i=1}^m a_i^0 b_i^{0\top} x_i.$$

Therefore the step 1 of **Deg-Det** can be executed by solving the unweighted linear matroid intersection problem for two matroids $\mathbf{M}(A^0)$ and $\mathbf{M}(B^0)$, where the matrices A^0, B^0 are defined by

$$A^0 := (a_1^0\ a_2^0\ \cdots\ a_m^0),\ B^0 := (b_1^0\ b_2^0\ \cdots\ b_m^0).$$

The matrices A^0, B^0 have the following structure.

Lemma 3.3 *If $(a_i^0)_k \ne 0$ and $\alpha_{k'} = \alpha_k$, then $(a_i^0)_{k'} = (a_i)_{k'}$. If $(a_i^0)_k \ne 0$ and $\alpha_{k'} > \alpha_k$, then $(a_i^0)_{k'} = (a_i)_{k'} = 0$. The same properties holds for B^0 with β.*

Proof The former claim is immediate from the definition (3.2). For the latter claim, suppose to the contrary that $(a_i^0)_k$ and $(a_i^0)_{k'}$ are nonzero and $\alpha_{k'} > \alpha_k$. Then for some ℓ, ℓ', $(b_i^0)_\ell$ and $(b_i^0)_{\ell'}$ are nonzero with $\alpha_k + \beta_\ell + c_i = \alpha_{k'} + \beta_{\ell'} + c_i = 0$ by the definition (3.2). Then, $(a_i)_{k'}(b_i)_\ell \ne 0$ and $\alpha_{k'} + \beta_\ell + c_i > \alpha_k + \beta_\ell + c_i = 0$. This contradicts (3.1). □

Suppose that we are given a common independent set X of $\mathbf{M}(A^0)$ and $\mathbf{M}(B^0)$. According to Edmonds' algorithm (given after Lemma 2.3), construct the residual graph $G_X^0 := G_X(A^0, B^0)$ with node sets $S_X^0 := S_X(A^0)$ and $T_X^0 := T_X(B^0)$. Then we can increase X or obtain K, L that are optimal to the problem (P^0) (as was explained in the end of Sect. 2.3).

A key observation here is that K and L are commuted with (t^α) and (t^β), respectively:

$$K(t^\alpha) = (t^\alpha)K, \; L(t^\beta) = (t^\beta)L. \tag{3.4}$$

Indeed, by Lemma 3.3, if $(a_i^0)_k$ and $(a_i^0)_{k'}$ are nonzero, then $\alpha_k = \alpha_{k'}$ holds. Therefore, each elementary row operation for A^0 is done between rows k, k' with $\alpha_k = \alpha_{k'}$. Consequently, the elimination matrix K is a block diagonal matrix in which the rows (columns) k, k' in the same block have the same $\alpha_k = \alpha_{k'}$. Then we can see the commutation (3.4) as

$$\begin{aligned} K(t^\alpha) &= \begin{pmatrix} K_1 & & & \\ & K_2 & & \\ & & \ddots & \\ & & & K_k \end{pmatrix} \begin{pmatrix} t^{\bar{\alpha}_1} I & & & \\ & t^{\bar{\alpha}_2} I & & \\ & & \ddots & \\ & & & t^{\bar{\alpha}_k} I \end{pmatrix} \\ &= \begin{pmatrix} t^{\bar{\alpha}_1} I & & & \\ & t^{\bar{\alpha}_2} I & & \\ & & \ddots & \\ & & & t^{\bar{\alpha}_k} I \end{pmatrix} \begin{pmatrix} K_1 & & & \\ & K_2 & & \\ & & \ddots & \\ & & & K_k \end{pmatrix} = (t^\alpha)K, \end{aligned} \tag{3.5}$$

where $\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_k$ are distinct values of $\alpha_1, \alpha_2, \dots, \alpha_n$.

Therefore the update in step 3 of **Deg-Det** is done as $P \leftarrow (t^{\alpha+\kappa\mathbf{1}_I})K$, $Q \leftarrow L(t^{\beta-\kappa\mathbf{1}_{[n]\setminus J}})$. Instead of doing such update, we update A, B as $A \leftarrow KA$, $B \leftarrow L^\top B$, which keeps $\deg_t \det M$, and update α, β as $\alpha \leftarrow \alpha + \kappa\mathbf{1}_I$, $\beta \leftarrow \beta - \kappa\mathbf{1}_{[n]\setminus J}$. Then P, Q are always of the form $(t^\alpha), (t^\beta)$, and can be treated as exponent vectors α, β, where $-\deg_t \det P - \deg_t \det Q = -\sum_{i=1}^n (\alpha_i + \beta_i)$. Now the algorithm is written, without explicit references to P, Q, K, L, as follows.

Algorithm: Deg-Det-WMI
Input: $n \times m$ matrices $A = (a_1\, a_2\, \cdots\, a_m)$, $B = (b_1\, b_2\, \cdots\, b_m)$, and weights $c_i \in \mathbb{Z}\, (i = 1, 2, \dots, m)$.
Output: $\deg_t \det M$ for $M := \sum_{i=1}^m a_i b_i^\top x_i t^{c_i}$.
0: $X = \emptyset$, $\alpha := -\max_i c_i \mathbf{1}$ and $\beta := \mathbf{0}$.
1: If $|X| = n$, then output $-\sum_{i=1}^n (\alpha_i + \beta_i)$ and stop. Otherwise, according to (3.2), (3.3), decompose A, B as $A = A^0 + A'$, $B = B^0 + B'$. Apply elementary row operations to A, B so that A^0, B^0 are X-diagonal forms.
2: From A^0, B^0, construct the residual graph G_X^0 and node sets S_X^0, T_X^0. Let R^0 be the set of nodes reachable from S_X^0 in G_X^0.

2-1. If $R^0 \cap T_X^0 \neq \emptyset$: Taking a shortest path P from S_X^0 to T_X^0, let $X \leftarrow X \triangle V(P)$, and go to step 1.

2-2. If $R^0 \cap T^0_X = \emptyset$: Then R^0 determines the zero submatrix $((t^\alpha)M(t^\beta))^0[I,J]$ of maximum size $|I| + |J|(> n)$ by (2.3) and (2.4); see also Figs. 2 and 3. Letting $\alpha \leftarrow \alpha + \kappa \mathbf{1}_I$, $\beta \leftarrow \beta - \kappa \mathbf{1}_{[n]\setminus J}$, increase κ from 0 until a nonzero entry appears in the zero submatrix. If $\kappa = \infty$ or $-\sum_{i=1}^n (\alpha_i + \beta_i) < n \min_i c_i$, then output $-\infty$ and stop. Otherwise go to step 1.

The step 2 in this algorithm is essentially Edmonds' algorithm to solve the unweighted matroid intersection problem for two matroids $\mathbf{M}(A^0)$, $\mathbf{M}(B^0)$ and an initial common independent set X. It turns out below that X is actually commonly independent for $\mathbf{M}(A^0)$ and $\mathbf{M}(B^0)$. Assuming this, it is clear that, in step 2-1, X increases and is a common independent set in the next step 1, and that, in step 2-2, X is a maximum common independent set and a maximum-size zero submatrix of $((t^\alpha)M(t^\beta))^0 = \sum_{i=1}^m a_i^0 b_i^{0\top} x_i$ is obtained accordingly. After the update of α, β, A^0 and B^0 are changed so that $A^0[[n]\setminus I, R^0]$ and $B^0[[n]\setminus J, E \setminus R^0]$ become zero blocks, and $A^0[I, E\setminus R^0]$ or $B^0[J, R^0]$ has nonzero entries; see Fig. 4 in Sect. 3.3. Other parts are unchanged. In particular, both $A^0[X]$ and $B^0[X]$ are lower triangular matrices (by row/column permutations). Therefore X keeps commonly independent for new matroids $\mathbf{M}(A^0)$ and $\mathbf{M}(B^0)$ in the next step 1. If $|X| = n$, then this is in the situation where $(PMQ)^0$ is nonsingular, and hence the algorithm correctly outputs $\deg_t \det M$ as $-\deg_t \det P - \deg_t \det Q = -\sum_{i=1}^n (\alpha_i + \beta_i)$. If the singularity of M is detected, e.g., $\deg_t \det M < n \min_{i \in [m]} c_i$, then it outputs $-\infty$.

Moreover, X is always a common independent set of $\mathbf{M}(A)$ and $\mathbf{M}(B)$ having the maximum weight among all common independent sets of cardinality $|X|$. Therefore **Deg-Det-WMI** can obtain a maximum weight independent set (of arbitrary cardinality) by adding the following procedure.

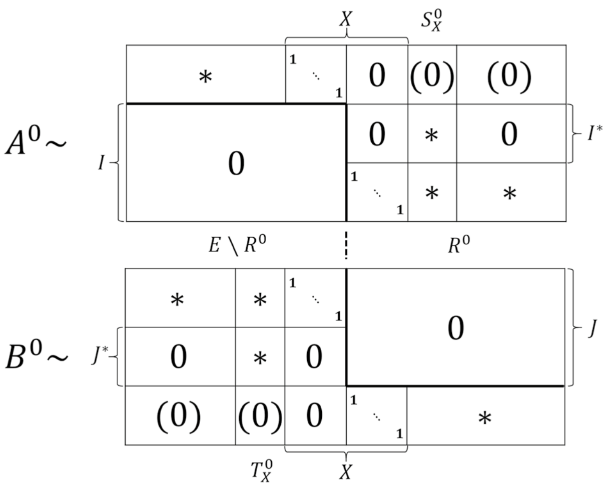


Fig. 4 Change of A^0, B^0

- After the update of X in step 2-1, for $k = |X|$, output $X_k := X$ as a maximum weight common independent set of cardinality k for $\mathbf{M}(A)$ and $\mathbf{M}(B)$.
- After the termination of the algorithm, output X^* from $X_0, X_1, \ldots, X_n$ having the maximum weight $c(X_i)$, where $X_0 := \emptyset$ and $c(X_k) := -\infty$ if X_k is undefined. Then X^* is a maximum weight common independent set for $\mathbf{M}(A)$ and $\mathbf{M}(B)$.

We show this fact by using the idea of *weight splitting* [4].

Lemma 3.4 *In step* 1, *define weight splitting* $c_i = c_i^1 + c_i^2$ *for each* $i \in [m]$ *by*

$$c_i^1 := c_i - c_i^2, \tag{3.6}$$

$$c_i^2 := -\max\{\beta_\ell \mid \ell \in [n] : (b_i)_\ell \neq 0\}. \tag{3.7}$$

Then X *is a common independent set of* $\mathbf{M}(A)$ *and* $\mathbf{M}(B)$ *such that* $c^1(X) = \max\{c^1(Y) \mid Y \in \mathcal{I}(A), |Y| = |X|\}$ *and* $c^2(X) = \max\{c^2(Y) \mid Y \in \mathcal{I}(B), |Y| = |X|\}$. *Thus* X *maximizes the weight* $c(X)$ *over all common independent sets of cardinality* $|X|$.

Proof We first verify that X is a common independent set of $\mathbf{M}(A)$ and $\mathbf{M}(B)$. We may assume $X = \{1, 2, \ldots, h\}$. Since X is commonly independent of $\mathbf{M}(A^0)$ and $\mathbf{M}(B^0)$, we can assume that $A^0[[h], X] = B^0[[h], X] = I$ in the X-diagonal forms. Then $I^* = J^* = \{h+1, \ldots, n\}$; recall (2.1) and (2.2). We can further assume that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_h$ and $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_h$. By Lemma 3.3, $A[X]$ and $B[X]$ are lower-triangular matrices with nonzero diagonals. Hence X is commonly independent for $\mathbf{M}(A)$ and $\mathbf{M}(B)$.

Next we make some observations to prove the statement. Observe from the definition (3.2) (3.3) (3.6) (3.7) and the properness (3.1) that

$$c_i^1 \leq -\alpha_k \quad (\forall k : (a_i)_k \neq 0), \tag{3.8}$$

$$c_i^2 \leq -\beta_\ell \quad (\forall \ell : (b_i)_\ell \neq 0), \tag{3.9}$$

and

$$c_i^1 = -\alpha_k, \quad c_i^2 = -\beta_\ell \quad (\forall k, \ell : (a_i^0)_k (b_i^0)_\ell \neq 0). \tag{3.10}$$

We also observe

$$\max_{k \in [n]} \alpha_k = \alpha_{k'} \ (\forall k' \in I^*), \quad \max_{\ell \in [n]} \beta_\ell = \beta_{\ell'} \ (\forall \ell' \in J^*). \tag{3.11}$$

This follows from the way of update $\alpha \leftarrow \alpha + \mathbf{1}_I$, $\beta \leftarrow \beta - \mathbf{1}_{[n]\setminus J}$ with the initialization $\alpha = -\max_i c_i \mathbf{1}$, $\beta = \mathbf{0}$ of the algorithm, and the fact that both $I^* \subseteq I$ and $J^* \subseteq J$ monotonically decrease.

Finally we prove that X maximizes both weights c^1 and c^2 for $\mathbf{M}(A)$ and $\mathbf{M}(B)$, respectively. It suffices to show

$$c^1(X) \geq c^1(X \cup \{i\}\backslash\{j\}) \quad (i \notin X, j \in X : X \cup \{i\}\backslash\{j\} \in \mathcal{I}(A)), \tag{3.12}$$

$$c^2(X) \geq c^2(X \cup \{i\}\backslash\{j\}) \quad (i \notin X, j \in X : X \cup \{i\}\backslash\{j\} \in \mathcal{I}(B)). \tag{3.13}$$

Indeed, this is the well-known optimality criterion of the maximum weight independent set problem on a matroid. Take i, j with $X \cup \{i\}\backslash\{j\} \in \mathcal{I}(A)$. If there is a nonzero element $(a_i)_{k^*} \neq 0$ for some $k^* \in I^*$, then by (3.8) and (3.11) it holds $c_i^1 \leq -\alpha_{k^*} \leq -\alpha_j = c_j^1$, where the equality follows from (3.10) and $(a_j^0)_j = 1$, and thus (3.12) holds. Suppose not. Let $k \in [h]$ be the smallest index such that $(a_i)_k \neq 0$. Then $c_i^1 \leq -\alpha_k$. Now $A[[h], X]$ is lower triangular. Additionally, by Lemma 3.3 and (3.11), $A[I^*, X] = A^0[I^*, X]$ is a zero matrix. Therefore, it must hold $j \geq k$ for i, j to belong to a circuit in $X \cup \{i\}$. Hence, $c_j^1 = -\alpha_j \geq -\alpha_k \geq c_i^1$. Thus (3.12) holds. (3.13) is similarly shown. □

3.3 Analysis

We analyze the time complexity of **Deg-Det-WMI**. It is obvious that if $R^0 \cap T_X^0 \neq \emptyset$ (step 2-1) occurs, then X increases and hence the rank of $((t^\alpha)M(t^\beta))^0$ increases. Therefore the algorithm goes to step 2-1 at most n times. The main analysis concerns step 2-2, particularly, how nonzero entries appear, how they affect A^0, B^0, and G_X^0, and how many times these scenarios occur until $R^0 \cap T_X^0 \neq \emptyset$.

As κ becomes positive, the submatrix $((t^\alpha)M(t^\beta))^0[[n]\backslash I, [n] \setminus J]$ becomes a zero block, since the degree of each element of $(t^\alpha)M(t^\beta)[[n]\backslash I, [n] \setminus J]$ decreases. Accordingly, $A^0[[n]\backslash I, R^0]$ and $B^0[[n]\backslash J, E \setminus R^0]$ become zero blocks; see Fig. 4.

Then, in G_X^0, all arcs entering R^0 disappear. Namely increasing κ only removes arcs entering to R^0 and does not change the other parts.

Next we analyze the moment when a non-zero element appears in $((t^\alpha)M(t^\beta))^0[I, J]$. Then, in the next step 1, it holds

$$(a_i^0)_k (b_i^0)_\ell \neq 0$$

for some $i \in [m]$, $k \in I$, $\ell \in J$. In this case, a new nonzero element appears in the i-th column of A^0 or B^0.

(a-1) If $i \notin R^0$ and $i \in X$: In the next step 1, Gaussian elimination for A^0 (and A) makes the new nonzero element $(a_i^0)_k = (a_i)_k$ zero. Since $A^0[[n]\backslash I, R^0] = O$, this does not affect $A^0[R^0]$. Therefore R^0 is still reachable from S_X^0. There may appear nonzero elements in $A^0[I, E\backslash R^0]$, which will make R^0 or S_X^0 larger in the next step 2.

(a-2) If $i \notin R^0$ and $i \notin X$: By $(a_i^0)_k \neq 0$, if $k \in I^*$, then i is included to S_X^0. Otherwise there appears an arc in G_X^0 from $X \cap R^0$ to i. For the both cases, i is included to R^0. By $\ell \in J$, if $\ell \in J^*$, then i belongs to T_X^0. Otherwise there is an arc from i to $X\backslash R^0$. Thus, $R^0 \cap T_X^0$ becomes nonempty if $\ell \in J^*$, and $|X \cap R^0|$ increases if $\ell \in J\backslash J^*$.

(b-1) If $i \in R^0$ and $i \in X$: Similar to the analysis of (a-1) above, Gaussian elimination for B^0 makes $(b_i^0)_k = (b_i)_k$ zero, and R^0 and T_X^0 increase or do not change.
(b-2) If $i \in R^0$ and $i \notin X$: By $(b_i^0)_\ell \neq 0$, if $\ell \in J^*$, then i is included to T_X^0, and $R^0 \cap T_X^0 \neq \emptyset$. Otherwise there appears an arc from i to $X \backslash R^0$, and $|X \cap R^0|$ increases.

Therefore, if the case (a-2) or (b-2) occurs, then $T_X^0 \cap R^0 \neq \emptyset$ or $|X \cap R^0|$ increases. After $O(n)$ occurrences of the cases (a-2) and (b-2), $T_X^0 \cap R^0$ becomes nonempty and $|X|$ increases. When X is updated, Gaussian elimination constructs the X-diagonal forms of A^0, B^0 in $O(mn^2)$ time.

We analyze the occurrences of (a-1) and (b-1). When $(a_i^0)_k$ becomes nonzero for some $i \in X \backslash R^0, k \in I$, it is eliminated by the row operation, and $(a_i^0)_k = (a_i)_k$ never becomes nonzero. Therefore, (a-1) and (b-1) occur at most $O(n|X|)$ time until X is updated, where the row operation is executed in $O(m)$ time per each occurrence. The total time for the elimination is $O(nm|X|)$. The augmentation κ and the identification of the next nonzero elements are computed in $O(nm)$ time by searching nonzero elements in A, B, which is needed for each time one of (a-1), (a-2), (b-1), and (b-2) occurs. Thus, by the naive implementation, **Deg-Det-WMI** runs in $O(mn^4)$ time.

We improve this complexity to $O(mn^3 \log n)$ as follows. Observe first that κ is given by

$$\kappa = -\max\{c_i + \alpha_k + \beta_\ell \mid i \in [m], k \in I, \ell \in J : (a_i)_k (b_i)_\ell \neq 0\}.$$

The main idea is to sort indices $(i, k, \ell) \in [m] \times I \times J$ according to $c_i + \alpha_k + \beta_\ell$ and keep in a binary heap the potential indices that attain κ. Notice that even if $(a_i)_k (b_i)_\ell$ is zero in a moment, it will become nonzero by row operations in (a-1) and (b-1) and can appear in $((t^\alpha) M(t^\beta))^0[I, J]$ later. On the other hand, any index (i, k, ℓ) with $c_i + \alpha_k + \beta_\ell > 0$ keeps $(a_i)_k (b_i)_\ell = 0$ and is irrelevant until X is updated.

Suppose now that X, A^0, B^0, and G_X^0 were updated in step 1. By BFS for G_X^0, we determine the reachable set R^0 and the index sets I, J. We can sort $c_i + \beta_\ell$ $(i \in [m], \ell \in J)$ in $O(mn \log m)$ time, which is improved to $O(mn \log n)$ time as follows. By sorting c_i $(i \in [m])$ in $O(m \log m)$ time, we obtain $|J|$ sorted lists of $c_i + \beta_\ell$ $(i \in [m])$ for $\ell \in J$. By keeping the head elements of these sorted lists in a heap, the whole sorted list can be obtained in $O(nm \log |J|)$, as in the merge sort.

From the sorted list, we construct an array p such that the e-th entry $p[e]$ has all indices (i, ℓ) with e-th largest $c_i + \beta_\ell$ as a linked list. For each $k \in I$, let p_k denote the copy of the array p, where $p_k[e]$ also has the value $v_{k,e} := c_i + \alpha_k + \beta_\ell$ for indices (i, ℓ) in $p_k[e]$. By the *head index* of p_k (relative to α, β, I, J), we mean the minimum index e_k such that $p_k[e_k]$ has the value v_{k,e_k} less than 0 and an index (i, ℓ) with $\ell \in J$, where J will decrease later. Notice that if $p_k[e]$ has the value $v_{k,e} \geq 0$, then $(a_i)_k (b_i)_\ell = 0$ for all indices (i, ℓ) in $p_k[e]$. Construct a binary (max) heap consisting of the pointers to the head indices e_k for all $k \in I$, where the key is the value v_{k,e_k} of $p_k[e_k]$. In the construction of the heap, if the key v_{k,e_k} of a node is equal to the key $v_{k',e_{k'}}$ of its parent node, then the two nodes are combined as a single node and the corresponding pointers are also combined as a single list. Then, by referring to the root of the heap, we know all indices $(i, k, \ell) \in [m] \times I \times J$ having the maximum negative value. Increase κ to the negative of this value (i.e., $\alpha \leftarrow \alpha + \kappa \mathbf{1}_I$, $\beta \leftarrow \beta - \kappa \mathbf{1}_{[n]\setminus J}$). If the root has no index (i, k, ℓ) with $(a_i)_k (b_i)_\ell \neq 0$, then delete the root from the heap, update the head index of each p_k indicated by the (deleted) root, and add the pointers of new head indices to the heap. Suppose that the root has an index (i, k, ℓ) with $(a_i)_k (b_i)_\ell \neq 0$; then $\kappa = -c_i - \alpha_k - \beta_\ell$. If $i \in X$ then execute the row operation to make $(a_i)_k (b_i)_\ell$ zero. As mentioned, once $(a_i)_k (b_i)_\ell$ becomes zero by the row operation, it never becomes nonzero. Here $(a_{i'})_{k'} (b_{i'})_{\ell'}$ for another index (i', k', ℓ') in the root may become nonzero from zero, which is eliminated in the next if $i' \in X$. Therefore, together with doing such row operations, after looking the indices in the root at most twice, the root has no index (i, k, ℓ) with $i \in X$ and $(a_i)_k (b_i)_\ell \neq 0$. Suppose that there is (i, k, ℓ) with $i \notin X$ and $(a_i)_k (b_i)_\ell \neq 0$. Then G_X^0, R^0, I, and J are updated. In particular, I increases and J decreases. For each newly added $k \in I$, construct array p_k (from p), identify the head index of p_k, and add the pointer to the heap. In this way, until X increases, each index (i, k, ℓ) is referred to at most twice, and the heap is updated in $O(\log n)$ time per the reference. In total, $O(mn^2 \log n)$ time is required. Thus we have:

Theorem 3.5 *Algorithm* **Deg-Det-WMI** *runs in* $\mathrm{O}(mn^3 \log n)$ *time*.

3.4 Relation to Frank's algorithm

In this subsection, we reveal the relation between our algorithm **Deg-Det-WMI** and Frank's weight splitting algorithm [4]. We show that the common independent sets X obtained by **Deg-Det-WMI** are the same as the ones obtained by a slightly modified version of Frank's algorithm. This means in a sense that **Deg-Det-WMI** is a nonstandard specialization of Frank's algorithm to linear matroids.

Let us briefly explain Frank's algorithm; our presentation basically follows [14, Sect. 13.7]. His algorithm keeps a weight splitting $c_i = c_i^1 + c_i^2$ for each $i \in E$ and a common independent set X such that X is maximum for both c_i^1 and c_i^2 over all common independent sets of cardinality $|X|$.

0: $c_i^1 := c_i$, $c_i^2 := 0$ for $i \in E$ and $X := \emptyset$.
1: Applying elementary row operations to A, B, construct the residual graph G_X, and node sets S_X, T_X as in Sect. 2.2.
2: From the weight splitting $c = c^1 + c^2$, construct subgraph $\bar{G}_X$ of G_X and node subsets $\bar{S}_X \subseteq S_X$, $\bar{T}_X \subseteq T_X$ by: $\bar{G}_X$ consists of arcs ij with $i \in X \not\ni j$ and $c_i^1 = c_j^1$ or $i \notin X \ni j$ and $c_i^2 = c_j^2$, and

$$\bar{S}_X := \{i \in S_X \mid \forall j \in S_X, c_i^1 \geq c_j^1\}, \tag{3.14}$$

$$\bar{T}_X := \{i \in T_X \mid \forall j \in T_X, c_i^2 \geq c_j^2\}. \tag{3.15}$$

3: Let $\bar{R}$ be the set of nodes reachable from $\bar{S}_X$ in $\bar{G}_X$.

4-1: If $\bar{R} \cap \bar{T}_X \neq \emptyset$, for a shortest path P from $\bar{S}_X$ to $\bar{T}_X$, replace X by $X\Delta V(P)$; go to step 1.
4-2: If $\bar{R} \cap \bar{T}_X = \emptyset$, then let $c_i^1 := c_i^1 - \epsilon$, $c_i^2 := c_i^2 + \epsilon$ for $i \in \bar{R}$, and increase ϵ from 0 until $\bar{R}$ increases. If $\epsilon = \infty$, then output $-\infty$ and stop. Go to step 2.

We consider a modified update of the weight splitting. Let $\bar{R}'$ be the subset of nodes $i \in E \backslash (X \cup \bar{R})$ such that all arcs leaving i enters $X \cap \bar{R}$. Then the step 4-2 can be replaced by the following:

4-2′: If $\bar{R} \cap \bar{T}_X = \emptyset$, then let $c_i^1 := c_i^1 - \epsilon$, $c_i^2 := c_i^2 + \epsilon$ for $i \in \bar{R} \cup \bar{R}'$, and increase ϵ from 0 until $\bar{R}$ increases or $\bar{R}'$ changes. If $\epsilon = \infty$, then output $-\infty$ and stop. Repeat until $\bar{R}$ increases and go to step 2.

One can easily check that X keeps the optimality (3.12), (3.13) in the modified update. Hence, the modified algorithm using **4-2′** is also correct.

We prove that G_X^0, S_X^0, T_X^0 in our algorithm and $\bar{G}_X$, $\bar{S}_X$, $\bar{T}_X$ in modified Frank's algorithm are the same up to an obvious redundancy. Here an arc in $\bar{G}_X$ is said to be *redundant* if it leaves a node i that has no arc entering i.

Proposition 3.6 *Suppose that* X, α *and* β *are obtained in an iteration of* **Deg-Det-WMI**. *Define weight splitting* $c_i = c_i^1 + c_i^2$ *by* (3.6), (3.7) *and* $\bar{G}_X$, $\bar{S}_X$ *and* $\bar{T}_X$ *by* (3.14), (3.15). *Then we have the following*:

(1) G_X^0 *is equal to the subgraph of* $\bar{G}_X$ *obtained by removing redundant arcs.*
(2) S_X^0 *is equal to* $\bar{S}_X$.
(3) T_X^0 *is equal to the subset of* $\bar{T}_X$ *obtained by removing isolated nodes.*
(4) R^0 *is equal to* $\bar{R}$.
(5) *The total sum of increases* κ *until* R^0 *changes is equal to that of increases* ϵ *until* $\bar{R}$ *changes in the modified Frank's algorithm.*

Proof Recall (the proof of) Lemma 3.4 that X is a common independent set of $\mathbf{M}(A)$ and $\mathbf{M}(B)$. Suppose that $X = \{1, 2, \ldots, h\}$ and $A^0[[h], X] = B^0[[h], X] = I$ with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_h$ and $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_h$. Observe first that $S_X^0 \subseteq S_X$. Indeed, from Lemma 3.3 and (3.11), $A[I^*, X]$ is a zero matrix. Therefore, if a_i^0 has a nonzero vector in a row in I^*, i.e., $i \in S_X^0$, then a_i is independent from $a_{i'}$ ($i' \in X$), i.e., $i \in S_X$. Consider the weight splitting of nodes in S_X^0. For $i \in S_X^0$, $c_i^1 = -\alpha_k$ ($k \in I^*$), and $-\alpha_k \geq c_j^1$ for $j \in S_X$ by (3.11). Thus $S_X^0 \subseteq \bar{S}_X$. Also, for any $i' \in S_X \backslash S_X^0$, $a_{i'}^0$ is a zero vector. Indeed, it holds $(a_{i'})_{k^*} \neq 0 = (a_{i'}^0)_{k^*}$ for some $k^* \in I^*$. This means $\alpha_{k^*} + \beta_\ell + c_{i'} < 0$ for all $\ell \in [n]$ with $(a_{i'})_{k^*}(b_{i'})_\ell \neq 0$. By (3.11), it holds $\alpha_k \leq \alpha_{k^*}$ for all $k \in [n]$, and $\alpha_k + \beta_\ell + c_{i'} < 0$ for all k, ℓ with $(a_{i'})_k(b_{i'})_\ell \neq 0$, which implies $a_{i'}^0 = \mathbf{0}$. Thus, it holds $c_{i'}^1 < -\alpha_k$ for $k \in I^*$. Then $c_{i'}^1 < -\alpha_k = c_i^1$ for $i \in S_X^0$. Thus we have (2).

Showing (3) is similar. As above, we see that $T_X^0 \subseteq T_X$ and for $i \in T_X^0$, $c_i^2 = -\beta_\ell$ ($\ell \in J^*$). Then $T_X^0 \subseteq \bar{T}_X$. Let $i \in T_X \backslash T_X^0$. Then b_i^0 is a zero vector, and so is a_i^0.

Suppose that arc ji for $j \in X$ exists in G_X. Recall that $A[[h], X]$ and $B[[h], X]$ are lower triangular. Then j is at least the minimum index k with $(a_i)_k \neq 0$. Then for $\ell \in J^*$, $c_j^1 = -\alpha_j \geq -\alpha_k > c_i + \beta_\ell = c_i - c_i^2 = c_i^1$, where the strictly inequality follows from the fact that a_i^0 and b_i^0 are zero vectors. Then ji does not exist in $\bar{G}_X$. Similarly, arc ij does not exist in $\bar{G}_X$ and this means i is an isolated node. Thus we have (3).

Next we compare G_X^0 and $\bar{G}_X$ to prove (1) and (4). Consider a node $i \in E \backslash X$ such that a_i^0 and b_i^0 are nonzero. Suppose that arc ki exists in G_X^0, i.e., $(a_i^0)_k \neq 0$ for $k \in [h] = X$. Then $c_i^1 = -\alpha_k = c_k^1$. We show that ki exists also in $\bar{G}_X$. Since $\alpha_j \geq \alpha_k$ $(j \neq k)$ implies $(a_k)_j = 0$ by Lemma 3.3, Gaussian elimination making A X-diagonal does not affect $(a_i)_k$. Thus the arcs ki exists in G_X and in $\bar{G}_X$. Similarly, if $i\ell$ exists in G_X^0, then $i\ell$ exists in $\bar{G}_X$. Therefore, for any node $i \in E \backslash X$ with nonzero a_i^0, b_i^0, the arcs incident to i are the same in G_X^0 and $\bar{G}_X$.

Consider a node $i \in E \backslash X$ such that a_i^0 and b_i^0 are zero vectors. In G_X^0, there are no arcs incident to i. For $k \in X, \ell \in [n]$ with $(a_i)_k (b_i)_\ell \neq 0$, it holds $c_k^1 = -\alpha_k > \beta_\ell + c_i \geq -c_i^2 + c_i = c_i^1$. This means that arcs ki entering i do not exist in $\bar{G}_X$, and thus arcs $i\ell$ leaving i are redundant. Thus we have (1). From (1), (2), and (3), we have (4).

Finally we prove (5). The step 2-2 in **Deg-Det-WMI** changes α, β as $\alpha \leftarrow \alpha + \kappa \mathbf{1}_I$, $\beta \leftarrow \beta - \kappa \mathbf{1}_{[n] \backslash J}$. We analyze the corresponding change of the weight splitting $c = c^1 + c^2$ defined by (3.6), (3.7). Consider $i \in [m]$ such that a_i^0 and b_i^0 are nonzero vectors. Suppose that $i \in R^0 = \bar{R}$. Then a_i^0 and b_i^0 have nonzero entries in a row in I and in $[n] \backslash J$, respectively; see Fig. 4. Therefore $c_i^1 = -\alpha_k$ for some $k \in I$ and $c_i^2 = -\beta_\ell$ for some $\ell \in [n] \backslash J$, and c_i^1, c_i^2 are changed as $c_i^1 \leftarrow c_i^1 - \kappa$, $c_i^2 \leftarrow c_i^2 + \kappa$. Suppose that $i \notin R^0$.

Then a_i^0 and b_i^0 have nonzero entries in a row in $[n] \backslash I$ and in J, respectively. In particular, $c_i^1 = -\alpha_k$ for some $k \in [n] \backslash I$ and $c_i^2 = -\beta_\ell$ for some $\ell \in J$.

Then the weight splitting does not change.

Thus, for any node i with nonzero a_i^0, b_i^0, the update corresponds to the step 4-2 or 4-2′.

Consider a node i with $a_i^0 = b_i^0 = \mathbf{0}$. Let Λ be the set of indices k that attain $\max_{k \in [n]:(a_i)_k \neq 0} \alpha_k$, and let Π be the set of indices ℓ that attain $\max_{\ell \in [n]:(b_i)_\ell \neq 0} \beta_\ell = -c_i^2$.

Case 1: $\Pi \cap J \neq \emptyset$. Then c_i^2 does not change and so does c_i^1. If $\Lambda \cap I \neq \emptyset$, then κ can increase until c_i^1 becomes $-\alpha_k$ for some $k \in \Lambda$.

Case 2: $\Pi \cap J = \emptyset$ ($\Leftrightarrow \Pi \subseteq [n] \backslash J$) Then c_i^2 changes as $c_i^2 \leftarrow c_i^2 + \kappa$, and hence c_i^1 changes as $c_i^1 \leftarrow c_i^1 - \kappa$. Here κ can increase until $\Pi \cap J \neq \emptyset$; then the situation goes to (Case 1).

Notice that arc $i\ell$ exists in $\bar{G}_X$ precisely when $-c_i^2 = \beta_\ell = -c_\ell^2$ for $\ell \in X \cap J$, and hence a node i in the case 2 is precisely a node in $\bar{R}'$. Therefore the changes of the weight splitting are the same in **Deg-Det-WMI** and in the modified Frank's algorithm (using step 4-2′). The steps are iterated for the same zero submatrix until

R^0 changes. Therefore, the total sum of κ is the same as that of ϵ in the modified Frank's algorithm. □

By this property, the obtained sequences of common independent sets X can be the same in **Deg-Det-WMI** and the modified Frank's algorithm. Therefore **Deg-Det-WMI** can also be viewed as yet another implementation of Frank's algorithm for linear matroids. A notable feature of **Deg-Det-WMI** is to skip unnecessary eliminations in constructing the residual graphs. To see this fact, consider the partition $\{\sigma_1, \sigma_2, \dots, \sigma_{n'}\}$ of $[n]$ such that $k, k' \in [n]$ belong to the same part if and only if $\alpha_k = \alpha'_k$. Then the elimination matrix K is a block diagonal matrix with block diagonals of size $|\sigma_i| \times |\sigma_i|$; recall (3.5). This means that the Gaussian elimination for A in step 1 is done in $O(m \sum_i |\sigma_i|^2)$ time. Therefore, if values α_k, β_ℓ are scattered, then K, L are very sparse, and the update of G_X^0 after X changes is very fast. On the other hand, necessary eliminations skipped at this moment will be done in the occurrences of (a-1) and (b-1). Hence, **Deg-Det-WMI** reduces eliminations compared with the usual implementation of Frank's algorithm to linear matroids. More thorough analysis (e.g., incorporating Cunningham's estimate [1] for the length of augmenting paths) is left to a future work.

We close this paper by giving an example in which the elimination results are actually different in the two algorithms.

Example 3.7 Consider matrices

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

and weight $c = (3\ 2\ 3\ 1\ 1)$. The both algorithms for this input can reach at $\alpha = (-2\ \ -2\ \ -2\ \ -2)$, $\beta = (-1\ 0\ 0\ 0)$ and $X = \{1, 2\}$ without elimination. Consider **Deg-Det-WMI** from this moment. The matrices A^0 and B^0 are given by

$$A^0 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, B^0 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

The Gaussian elimination makes $(a_2^0)_2$ zero. Then G_X^0 consists of one arc 31, and $S_X^0 = \{3\}$ and $T_X^0 = \emptyset$. The reachable set R^0 is determined as $R^0 = \{1, 3\}$, and I, J are given by $I = \{1, 2, 3\}$, $J = \{2, 3, 4\}$, $I^* = \{1, 3\}$, and $J^* = \{2, 3\}$. Then α, β are changed as $\alpha = (-1\ \ -1\ \ -1\ \ -2)$, $\beta = (-2\ 0\ 0\ 0)$ without occurrences of (a-1) and (b-1). Nonzero elements appear in $A^0[I^*, \{4, 5\}]$ and $B^0[J^*, \{4, 5\}]$, which implies $S_X^0 \cap T_X^0 = \{4, 5\}$. So X is increased.

Therefore **Deg-Det-WMI** succeeds the augmentation without eliminating $(b_2)_1$, whereas Frank's algorithm eliminates this element in constructing G_X.

Acknowledgements The authors thank Kazuo Murota for comments and the anonymous referee for careful reading and helpful comments. The second author was supported by JSPS KAKENHI Grant Numbers JP17K00029 and JST PRESTO Grant Number JPMJPR192A, Japan.

References

1. Cunningham, W.H.: Improved bounds for matroid partition and intersection algorithm. SIAM J. Comput. **15**(4), 948–957 (1986)
2. Edmonds, J.: Systems of distinct representatives and linear algebra. J. Res. Natl. Bureau Stand. **71B**, 241–245 (1967)
3. Edmonds, J.: Matroid intersection. Ann. Discret. Math. **4**, 39–49 (1979)
4. Frank, A.: A weighted matroid intersection algorithm. J. Alg. **2**, 328–336 (1981)
5. Gabow, H.N., Xu, Y.: Efficient theoretic and practical algorithms for linear matroid intersection problems. J. Comput. Syst. Sci. **53**, 129–147 (1996)
6. Garg, A., Gurvits, L., Oliveira, R., Wigderson, A.: Operator scaling: theory and applications. Found. Comput. Math. (2019)
7. Harvey, N.J.A.: Algebraic algorithms for matching and matroid problems. SIAM J. Comput. **39**(2), 679–702 (2009)
8. Hirai, H.: Computing the degree of determinants via discrete convex optimization on Euclidean buildings. SIAM J. Appl. Algebra Geom. **3**(3), 523–557 (2019)
9. Huang, C.-C., Kakimura, N., Kamiyama, N.: Exact and approximation algorithms for weighted matroid intersection. Math. Program. Ser. A **177**, 85–112 (2019)
10. Ivanyos, G., Qiao, Y., Subrahmanyam, K.V.: Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics. Comput. Complex. **27**, 561–593 (2018)
11. Murota, K.: Computing the degree of determinants via combinatorial relaxation. SIAM J. Comput. **24**, 765–796 (1995)
12. Murota, K.: Matrices and Matroids for Systems Analysis. Springer-Verlag, Berlin (2000)
13. Lovász, L.: Singular spaces of matrices and their application in combinatorics. Boletim da Sociedade Brasileira de Matemática **20**, 87–99 (1989)
14. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms, 6th edn. Springer, Berlin (2018)
15. Lawler, E.L.: Matroid intersection algorithms. Math. Program. **9**, 31–56 (1975)
16. Schrijver, A.: Combinatorial optimization-polyhedra and efficiency. Springer, Berlin (2003)
17. Tomizawa, N., Iri, M.: An algorithm for determining the rank of a triple matrix product AXB with application to the problem of discerning the existence of the unique solution in a network. Trans. Inst. Electron. Inf. Commun. Eng. Japan **57A**, 834–841 (1974). (English translation in *Electronics and Communications in Japan***57A** (1974), 50–57)