Mathematical Programming 69 (1995) 449–470

New algorithms for linear k-matroid intersection and matroid k-parity problems ☆

Alexander I. Barvinok *,[1]

Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1003, USA

Received 24 October 1991; revised manuscript received 21 September 1994

Abstract

We present algorithms for the k-Matroid Intersection Problem and for the Matroid k-Parity Problem when the matroids are represented over the field of rational numbers and $k > 2$. The computational complexity of the algorithms is linear in the cardinality and singly exponential in the rank of the matroids. As an application, we describe new polynomially solvable cases of the k-Dimensional Assignment Problem and of the k-Dimensional Matching Problem. The algorithms use some new identities in multilinear algebra including the generalized Binet–Cauchy formula and its analogue for the Pfaffian. These techniques extend known methods developed earlier for $k = 2$.

Keywords: k-Matroid Intersection Problem; Matroid k-Parity Problem; Hyperdeterminant

1. Introduction

In this paper, we present new algorithms for the k-Matroid Intersection Problem and for the Matroid k-Parity Problem when $k > 2$ and the given matroids are represented over the field of rational numbers. These problems are known to be NP-hard, and so far no algorithms with better worst-case complexity than that of exhaustive search are known for them. On the other hand, many problems of combinatorial optimization can be posed as special cases of these problems on matroids and therefore it would be useful to find somewhat faster algorithms (see, for example, [9]). Such a question was asked, for example, in [16]. In [9] it was conjectured that the methods of partial enumeration might be the best ones. The complexity of our algorithms is linear in

☆ A preliminary version of this paper appeared in the Proceedings of the Second IPCO Conference [2].

* e-mail: barvinok@math.lsa.umich.edu.

[1] Supported by the Mittag-Leffler Institute and KTH, Stockholm.

SSDI 0025-5610(94)00073-5

the cardinality of the matroids and singly exponential in their rank (for a fixed k). Thus if the cardinality grows faster than a linear function of the rank (this is the case for most combinatorial applications), then our algorithms are asymptotically faster than exhaustive search. Moreover, it follows that if the rank grows no faster than the logarithm of the cardinality, then our algorithms have polynomial-time complexity. This result is also new. Finally, if we fix both k, the number of matroids, and r, the rank of matroids, the algorithms summarized in Section 4 solve the problem in time that grows linearly with the cardinality of the ground set.

Although there are matroids that cannot be represented over the field of rationals, many combinatorially and algorithmically interesting matroids do have this property. Thus our algorithms lead to new results for some old algorithmic problems in combinatorics. In particular, we describe new polynomially solvable cases of the k-Dimensional Assignment Problem and of the k-Dimensional Matching Problem for $k > 2$ (see, for example, [7,9]). We prove that for any fixed k one can determine in polynomial time whether there exist $O(\log n)$ pairwise disjoint edges in a given uniform k-hypergraph on n vertices. We describe combinatorial applications in Section 5.

Our approach is based on multilinear algebra. This approach proved to be fruitful in the case of $k = 2$. The Binet–Cauchy formula for the determinant of the product of two matrices had been used for the Matroid Intersection Problem and a formula for the Pfaffian of a special matrix had been used for the Matroid Matching Problem (see, for example, [3,10,12,14]). In Section 2 we briefly sketch these connections. In this paper we develop this algebraic approach further for $k > 2$, finding underlying identities from multilinear algebra. To obtain these generalizations, we invoke some classical notions due to Cayley [4,5] including tensors and hyperdeterminants, and introduce the hyperpfaffian of a tensor. These new identities appear to be interesting in their own right. This technique is presented in Section 3. Finally, we reduce our problems to the computation of the hyperdeterminant or the hyperpfaffian of a tensor and then use dynamic programming (Section 4).

Let us formulate the problems that we will address. We consider linear matroids represented over the rationals (see [16] for the definition of a general matroid). Such a matroid is represented by an integral rectangular $r \times n$ matrix $A = \{A(i,j)\colon 1 \leqslant i \leqslant r, 1 \leqslant j \leqslant n\}$. (We write indices in parentheses rather than using subscripts.) We assume that $r \leqslant n$ and that rank $A = r$. The numbers n and r are referred to as the *cardinality* and the *rank* respectively of the matroid represented by A. For a subset $I \subset \{1,\dots,n\}$ of cardinality r we denote by A_I the $r \times r$ submatrix of A consisting of the columns indexed by the elements of I. A subset I for which $\det A_I \neq 0$ is called a *base* of the matroid represented by A. The *matroid* represented by A is the set $\{1,\dots,n\}$ together with the family of all bases. Note that different matrices can represent the same matroid. Let us state the first problem that we consider.

(1.1) k-Matroid Intersection Problem. Let us fix $k \in \mathbb{N}$. Given $r, n \in \mathbb{N}$, and k integral rectangular $r \times n$ matrices $A^1,\dots,A^k$, decide whether there exists a subset $I \subset \{1,\dots,n\}$ of cardinality r such that all the $r \times r$ submatrices $A^1_I,\dots,A^k_I$ are nonsingular, that is,

$\det A_I^1 \neq 0, \ldots, \det A_I^k \neq 0$.

In other words, we are interested in whether the matroids represented by $A^1, \ldots, A^k$ have a common base. In the Matroid k-Parity Problem we restrict ourselves to matroids whose rank and cardinality are divisible by k and look for a base of a special form.

(1.2) Matroid k-Parity Problem. Let us fix $k \in \mathbb{N}$. Suppose that natural numbers r, N and an $r \times N$ rectangular integral matrix A are given. We assume further that r and N are divisible by k, so $r = km$ and $N = kn$ for some $m, n \in \mathbb{N}$. Let $P_0 = \{1, \ldots, k\}$, $P_1 = \{k+1, \ldots, 2k\}, \ldots, P_{n-1} = \{N-k+1, \ldots, N\}$ be a partition of the set $\{1, \ldots, N\}$ into disjoint k-sets. Decide, whether there exists a subset $I \subset \{1, \ldots, N\}$ of the form $I = P_{i_1} \cup \cdots \cup P_{i_m}$ for some $0 \leqslant i_1 < \cdots < i_m \leqslant n-1$, such that the corresponding $r \times r$ submatrix A_I is nonsingular, i.e., $\det A_I \neq 0$.

For $k = 2$, Problems 1.1 and 1.2 admit polynomial time algorithms (see [11]). As we mentioned earlier, both of them are NP-hard for $k > 2$.

We present an algorithm for Problem 1.1 whose complexity is $O(r^{2k}(4^{rk} + n))$ and an algorithm for Problem 1.2 whose complexity is $O(r^{2k+1}(4^r + n))$. The computational model is the RAM with the uniform cost criterion (see [1]). We take care that the bit size of numbers encountered in the course of our algorithms is bounded by a polynomial in the total bit size of the input data.

We pose (1.1) and (1.2) as decision problems. One can reduce the problem of finding a base I to a sequence of decision problems using the standard divide-and-conquer approach. Let us consider the k-Matroid Intersection Problem (the Matroid k-Parity Problem can be treated in a similar way). Suppose we know that a base I in Problem 1.1 indeed exists. Let us check if $n \in I$. Let $\tilde{A}^i$, $i = 1, \ldots, k$, be the $r \times n-1$ submatrix of A^i consisting of the first $n-1$ columns. We apply an algorithm for testing the decision problem (1.1) with these submatrices. If the answer is "no", then necessarily $n \in I$ and we try the next element, say, $n-1$. If the answer is "yes", then there exists a base I such that $n \notin I$ and we try the next element $n-1$ with the submatrices $\tilde{A}^i$. This construction adds an extra factor n to the complexity bound for the corresponding decision problem.

Notation. We denote by $[1 : r]$ the set of natural numbers $\{1, 2, \ldots, r\}$. We denote by $|I|$ the cardinality of a finite set I. We denote by S_r the symmetric group, i.e., the group of all permutations of the set $[1 : r]$. For a number $i \in [1 : r]$ and a permutation $\sigma \in S_r$ we denote by $\sigma(i)$ the image of i under permutation σ. Thus $\sigma(i) \in [1 : r]$. Let $I = (i_1, \ldots, i_r)$ be a string of distinct natural numbers. A pair i_s, i_t such that $s < t$ and $i_s > i_t$ is called an *inversion* in I. We denote by $\operatorname{inv}(I)$ the number of inversions in I. If $\sigma \in S_r$ is a permutation, then by $\operatorname{inv}(\sigma)$ we denote $\operatorname{inv}(\sigma(1), \sigma(2), \ldots, \sigma(r))$. Finally, let $\operatorname{sgn} \sigma = (-1)^{\operatorname{inv}(\sigma)}$. We write the indices of matrices and tensors in parentheses. Thus the determinant of an $r \times r$ square matrix A can be written as follows:

$$\det A = \sum_{\sigma \in S_r} \operatorname{sgn} \sigma \prod_{i=1}^{r} A\big(i, \sigma(i)\big).$$

2. Preliminaries. The case $k = 2$

In this section we recall some known connections between our problems for $k = 2$ and identities involving determinants and Pfaffians (see [3,10,12,14]). Our main goal is to provide a certain intuition on interactions between multilinear algebra and problems on matroids which will be applicable for $k > 2$ as well. However, this is neither a survey of [3,10,12,14] and related papers, nor is it intended to be.

(2.1) The Matroid Intersection Problem and the Binet–Cauchy formula. Let A^1 and A^2 be $r \times n$ integral matrices. Let us define an $r \times r$ square matrix C by the formula

$$C(i,j) = \sum_{s=1}^{n} A^1(i,s) \cdot A^2(j,s),$$

for all $1 \leqslant i, j \leqslant r$. In other words, C is the product of A^1 and the transpose of A^2. Then the Binet–Cauchy formula (see, for example, [13, Theorem 9, p. 78]) asserts that

$$\sum_{I \subset [1:n]:\, |I| = r} \det A^1_I \cdot \det A^2_I = \det C. \tag{2.1.1}$$

As is known, the determinant of the $r \times r$ matrix C can be computed using $O(r^3)$ arithmetic operations so that the bit size of all the numbers involved in the computation is bounded by a polynomial in the input size. If $\det C \neq 0$, then there exists a common base of the matroids represented by A^1 and A^2. However, if $\det C = 0$, then we cannot immediately tell whether there is a common base since nonzero summands on the left-hand side of (2.1.1) might annihilate one another. To overcome this difficulty, several approaches can be used. First, in some lucky instances it might happen that all the summands in the left-hand side of (2.1.1) are nonnegative. Then the equality $\det C = 0$ implies that no common base of the matroids represented by A^1 and A^2 exists. This is the case, for example, if $A^1 = A^2$. Second, we can "perturb" the matrix A^1 multiplying its columns by randomly chosen nonzero integers $t_1, \ldots, t_n$ (this perturbed matrix represents the same matroid). Then for "almost any" choice of the parameters $t_1, \ldots, t_n$ the equality $\det C = 0$ implies that no common base exists. This approach is used, for example, in [3,14], where some efficient probabilistic algorithms for solving Problem 1.1 and its weighted versions are described.

(2.2) The Matroid Parity Problem and the Pfaffian. We can treat Problem 1.2 in a similar way. Instead of the determinant, we use another object, namely the Pfaffian. Let r be an even number, $r = 2m$. The Pfaffian of an $r \times r$ square matrix C is defined by the formula

$$\operatorname{Pf} C = \frac{1}{m! \cdot 2^m} \sum_{\sigma \in S_r} \operatorname{sgn} \sigma \prod_{i=0}^{m-1} C\big(\sigma(2i+1), \sigma(2i+2)\big)$$

(see, for example, [13]). The Pfaffian of an $r \times r$ integral matrix can be computed using $O(r^3)$ arithmetic operations [6] (again, the bit size of all the involved numbers is bounded by a polynomial in the input size). Let us consider a possible application of the Pfaffian to Problem 1.2. For a given $r \times N$ rectangular matrix A, where $r = 2m$ and $N = 2n$, let us compute an $r \times r$ matrix C as follows:

$$C(i,j) = \sum_{s=0}^{n-1} A(i, 2s+1) \cdot A(j, 2s+2),$$

for all $1 \leqslant i, j \leqslant r$. Then,

$$\sum_I \det A_I = 2^m \cdot \operatorname{Pf} C, \tag{2.2.1}$$

where the sum is taken over all subsets $I \subset [1 : N]$ of cardinality r that can be represented as a disjoint union of m pairs $P_{i_1} = \{2i_1 + 1, 2i_1 + 2\}, \dots, P_{i_m} = \{2i_m + 1, 2i_m + 2\}$ for some $0 \leqslant i_1 < \cdots < i_m \leqslant n-1$ (see [3]). Again, if the right-hand side of (2.2.1) is nonzero, then the answer in Problem 1.2 is "yes". It might happen that all the summands in (2.2.1) have the same sign and thus the converse is also true. This is the case, for example, when $A(i, 2j-1) = A(i+m, 2j)$ and $A(i+m, 2j-1) = A(i, 2j) = 0$ for $i = 1, \dots, m$ and $j = 1, \dots, n$ (such a matrix appears when we pose the problem of finding a base in a single linear matroid as an instance of the Matroid Parity Problem). Otherwise, we can perturb A, multiplying its columns by randomly chosen nonzero integers $t_1, \dots, t_N$ so that the converse is true with high probability. Such an approach is used in [10] where an efficient probabilistic algorithm for Problem 1.2 is described. In [3] a version of identity (2.2.1) is used to design a pseudopolynomial random algorithm for a weighted version of the problem.

For applications of determinants and Pfaffians to problems on graphs, see also [12].

In order to tackle Problems 1.1 and 1.2 when $k > 2$, we generalize (2.2.1) and (2.2.2). Namely, we want expressions for

$$\sum_{I \subset [1:n] : |I| = r} \det A_I^1 \cdot A_I^2 \cdots \det A_I^k \quad \text{and} \quad \sum_I \det A_I,$$

where the last sum is taken over subsets I of the type required by Problem 1.2. The expressions that we obtain will require the evaluation of "hyperdeterminants" and "hyperpfaffians". These evaluations are much more time-consuming than those of determinants and Pfaffians, nonetheless we achieve computational savings over enumeration.

3. Tensors, their hyperdeterminants and hyperpfaffians

In this section we present some technique of multilinear algebra that we make use of in our algorithms. It turns out that, passing from $k = 2$ to $k > 2$, we should replace matrices by tensors, determinants by hyperdeterminants, and Pfaffians by hyperpfaffians. The notion of hyperdeterminant was introduced by Cayley [4,5], whereas the definition of hyperpfaffian is new. All the corresponding identities are quite simple and straightforward although formulas sometimes might seem cumbersome. We consider a tensor as a k-dimensional array of real numbers.

(3.1) Definition. Let us choose natural numbers k and r. We denote by $[1:r]^k$ the product $[1:r]\times\cdots\times[1:r]$ (k times), i.e., the set of all ordered k-tuples $(i_1,\dots,i_k)$ where $1 \leqslant i_1,\dots,i_k \leqslant r$. By a (real) k-dimensional tensor of order r we understand a map

$$C : [1:r]^k \longrightarrow \mathbb{R}.$$

We also write

$$C = \{C(i_1,\dots,i_k):\ 1 \leqslant i_1,\dots,i_k \leqslant r\},$$

thus considering the tensor C as a k-dimensional $r \times \cdots \times r$ array of the numbers $C(i_1,\dots,i_k)$. We say that $C(i_1,\dots,i_k)$ are *entries* of C.

To generalize the determinant of a matrix, we introduce the hyperdeterminant of a tensor.

(3.2) Definition (Cayley [4,5]). Suppose that k is even. For a k-dimensional tensor

$$C = \{C(i_1,\dots,i_k):\ 1 \leqslant i_1,\dots,i_k \leqslant r\}$$

of order r, the expression

$$\operatorname{DET} C = \frac{1}{r!} \sum_{\sigma_1,\sigma_2,\dots,\sigma_k \in S_r} \operatorname{sgn}\sigma_1 \cdot \operatorname{sgn}\sigma_2 \cdots \operatorname{sgn}\sigma_k \times \prod_{i=1}^{r} C\big(\sigma_1(i),\sigma_2(i),\dots,\sigma_k(i)\big) \tag{3.2.1}$$

is called the *hyperdeterminant* of C. Since for any given set of $k-1$ permutations $\tau_2,\dots,\tau_k \in S_r$ all the $r!$ summands of (3.2.1) corresponding to the permutations $\{\sigma_1 = \sigma, \sigma_2 = \tau_2 \cdot \sigma, \dots, \sigma_k = \tau_k \cdot \sigma : \sigma \in S_r\}$ are equal, we get yet another expression for the hyperdeterminant:

$$\operatorname{DET} C = \sum_{\sigma_2,\dots,\sigma_k \in S_r} \operatorname{sgn}\sigma_2 \cdots \operatorname{sgn}\sigma_k \prod_{i=1}^{r} C\big(i,\sigma_2(i),\dots,\sigma_k(i)\big). \tag{3.2.2}$$

If $k = 2$, we get the usual determinant of a matrix. If k is odd, then the expression (3.2.1) is identically zero.

The following result provides the key tool for our consideration of the k-Matroid Intersection Problem. It can be considered as a natural generalization of the Binet–Cauchy formula. Although very simple, this result is new.

(3.3) Lemma. *Let k be even and let $A^1, \dots, A^k$ be rectangular real $r \times n$ matrices; $r \leqslant n$. Thus $A^s = \{A^s(i,j)\colon\ 1 \leqslant i \leqslant r, 1 \leqslant j \leqslant n\}$, $s = 1, \dots, k$. For a subset $I \subset [1:n]$ of cardinality r we denote by A^s_I the $r \times r$ submatrix of the matrix A^s consisting of the columns of A^s indexed by the elements of the set I. Let us define a k-dimensional tensor C of order r by the formula*

$$C(i_1, i_2, \dots, i_k) = \sum_{j=1}^{n} A^1(i_1, j) \cdot A^2(i_2, j) \cdots A^k(i_k, j), \tag{3.3.1}$$

for all $1 \leqslant i_1, \dots, i_k \leqslant r$. Then,

$$\sum_I \det A^1_I \cdot \det A^2_I \cdots \det A^k_I = \operatorname{DET} C,$$

where the sum is taken over all subsets $I \subset [1:n]$ of cardinality r.

Proof. We substitute (3.3.1) into (3.2.1). Thus we have

$$\begin{aligned}
\operatorname{DET} C &= \frac{1}{r!} \sum_{\sigma_1, \sigma_2, \dots, \sigma_k \in S_r} \operatorname{sgn}\sigma_1 \cdot \operatorname{sgn}\sigma_2 \cdots \operatorname{sgn}\sigma_k \\
&\quad \times \prod_{i=1}^{r} \sum_{j=1}^{n} A^1\big(\sigma_1(i), j\big) \cdot A^2\big(\sigma_2(i), j\big) \cdots A^k\big(\sigma_k(i), j\big) \\
&= \frac{1}{r!} \sum_{\sigma_1, \sigma_2, \dots, \sigma_k \in S_r} \operatorname{sgn}\sigma_1 \cdot \operatorname{sgn}\sigma_2 \cdots \operatorname{sgn}\sigma_k \\
&\quad \times \sum_{1 \leqslant j_1, \dots, j_r \leqslant n} \prod_{i=1}^{r} A^1\big(\sigma_1(i), j_i\big) \cdot A^2\big(\sigma_2(i), j_i\big) \cdots A^k\big(\sigma_k(i), j_i\big) \\
&= \frac{1}{r!} \sum_{1 \leqslant j_1, \dots, j_r \leqslant n} \sum_{\sigma_1, \dots, \sigma_k \in S_r} \operatorname{sgn}\sigma_1 \cdots \operatorname{sgn}\sigma_k \\
&\quad \times \prod_{i=1}^{r} A^1\big(\sigma_1(i), j_i\big) \cdots A^k\big(\sigma_k(i), j_i\big).
\end{aligned}$$

For a given sequence $J = (j_1, \dots, j_r)$, $1 \leqslant j_1, \dots, j_r \leqslant n$, and $s = 1, \dots, k$ let us denote by $\tilde{A}^s_J$ the $r \times r$ real matrix whose ith column is the j_ith column of the matrix A^s for $i = 1, \dots, r$. Then for all $J = (j_1, \dots, j_r)$ we have

$$\sum_{\sigma_1,\sigma_2,\ldots,\sigma_k\in S_r} \operatorname{sgn}\sigma_1\cdot\operatorname{sgn}\sigma_2\cdots\operatorname{sgn}\sigma_k$$
$$\times\prod_{i=1}^{r} A^1(\sigma_1(i),j_i)\cdot A^2(\sigma_2(i),j_i)\cdots A^k(\sigma_k(i),j_i)$$
$$=\Big(\sum_{\sigma\in S_r}\operatorname{sgn}\sigma\prod_{i=1}^{r}A^1(\sigma(i),j_i)\Big)\cdots\Big(\sum_{\sigma\in S_r}\operatorname{sgn}\sigma\prod_{i=1}^{r}A^k(\sigma(i),j_i)\Big)$$
$$=\det\tilde{A}^1_J\cdot\det\tilde{A}^2_J\cdots\det\tilde{A}^k_J.$$

Therefore we get

$$\operatorname{DET} C=\frac{1}{r!}\sum_{J=(j_1,\ldots,j_r)}\det\tilde{A}^1_J\cdot\det\tilde{A}^2_J\cdots\det\tilde{A}^k_J,$$

where the sum is taken over all sequences $1\leqslant j_1,\ldots,j_r\leqslant n$. If a sequence $J=(j_1,\ldots,j_r)$ contains a pair of equal numbers, then the corresponding summand is equal to zero, since the matrix $\tilde{A}^1_J$, say, contains a pair of identical columns. If we transpose two elements of a given sequence J, then all the numbers $\det\tilde{A}^1_J,\ldots,\det\tilde{A}^k_J$ reverse their signs. Since k is even, then all the summands corresponding to the $r!$ different orderings of a given set $\{j_1,\ldots,j_r\}\subset[1:n]$ are equal. Therefore we get

$$\operatorname{DET} C=\sum_{J=(j_1<j_2<\cdots<j_r)}\det\tilde{A}^1_J\cdot\det\tilde{A}^2_J\cdots\det\tilde{A}^k_J$$
$$=\sum_{I\subset[1:n],\,|I|=r}\det A^1_I\cdot\det A^2_I\cdots\det A^k_I,$$

and the proof follows. □

Next, we generalize the Pfaffian to tensors.

(3.4) Definition. Let k be an even number and let

$$C=\{C(i_1,\ldots,i_k):\ 1\leqslant i_1,\ldots,i_k\leqslant r\}$$

be a k-dimensional tensor of order r. Assume that $r=km$ for some $m\in\mathbb{N}$. The expression

$$\operatorname{PF} C=\frac{1}{m!}\sum_{\sigma\in S_r}\operatorname{sgn}\sigma\prod_{i=0}^{m-1}C\big(\sigma(ki+1),\sigma(ki+2),\ldots,\sigma(ki+k)\big)\tag{3.4.1}$$

will be called the *hyperpfaffian* of C.

Note that for any given permutation $\tau\in S_r$ all the $m!$ summands of (3.4.1) corresponding to the permutations $\{\sigma=\tau\cdot\mu$: μ permutes the ordered k-tuples $(1,\ldots,k)$, $(k+1,\ldots,2k),\ldots,(r-k+1,\ldots,r)\}$ are equal. In particular, if all entries of C are integers, then PF C is also an integer. If $k=2$, then C is an $r \times r$ square matrix and PF $C = 2^m \cdot \mathrm{Pf}\, C$, where Pf is the usual Pfaffian of a matrix. One can observe that if k is odd, then the expression (3.4.1) is identically zero. Applications of the hyperpfaffian to the Matroid k-Parity Problem are based on the following result.

(3.5) Lemma. *Let k, N and r be natural numbers. Assume that k is even and that $r = km$ and $N = kn$ where $m, n \in \mathbb{N}$. Let $A = \{A(i,j) \colon 1 \leqslant i \leqslant r, 1 \leqslant j \leqslant N\}$ be a rectangular $r \times N$ real matrix. For a subset $I \subset [1:N]$ of cardinality r we denote by A_I the submatrix of A consisting of the columns indexed by the elements of I. Suppose that the set $\{1,\dots,N\}$ is represented as a disjoint union of the sets $P_0 = \{1,\dots,k\}$, $P_1 = \{k+1,\dots,2k\},\dots,P_{n-1} = \{N-k+1,\dots,N\}$, each of cardinality k. Let us define a k-dimensional tensor C of order r by the formula:*

$$C(i_1, i_2, \dots, i_k) = \sum_{j=0}^{n-1} A(i_1, kj+1) \cdot A(i_2, kj+2) \cdots A(i_k, kj+k), \tag{3.5.1}$$

for all $1 \leqslant i_1, \dots, i_k \leqslant r$. Then,

$$\sum_I \det A_I = \mathrm{PF}\, C,$$

where the sum is taken over all subsets $I \subset [1:N]$ of cardinality r that can be represented as a union $P_{i_1} \cup \cdots \cup P_{i_m}$ for some $0 \leqslant i_1 < \cdots < i_m \leqslant n-1$.

Proof. We substitute (3.5.1) into (3.4.1). We have

$$\begin{aligned}
\mathrm{PF}\, C &= \frac{1}{m!} \sum_{\sigma \in S_r} \operatorname{sgn} \sigma \prod_{i=0}^{m-1} \sum_{j=0}^{n-1} A\big(\sigma(ki+1), kj+1\big) \\
&\qquad \times A\big(\sigma(ki+2), kj+2\big) \cdots A\big(\sigma(ki+k), kj+k\big) \\
&= \frac{1}{m!} \sum_{\sigma \in S_r} \operatorname{sgn} \sigma \sum_{0 \leqslant j_0, \dots, j_{m-1} \leqslant n-1} \prod_{i=0}^{m-1} A\big(\sigma(k \cdot i+1), k \cdot j_i + 1\big) \\
&\qquad \cdots A\big(\sigma(k \cdot i+k), k \cdot j_i + k\big) \\
&= \frac{1}{m!} \sum_{0 \leqslant j_0, \dots, j_{m-1} \leqslant n-1} \sum_{\sigma \in S_r} \operatorname{sgn} \sigma \\
&\qquad \times \prod_{i=0}^{m-1} A\big(\sigma(k \cdot i+1), k \cdot j_i + 1\big) \cdots A\big(\sigma(k \cdot i+k), k \cdot j_i + k\big).
\end{aligned}$$

For a given sequence $J = (0 \leqslant j_0, \dots, j_{m-1} \leqslant n-1)$ let us construct a real $r \times r$ matrix $\tilde{A}_J$ in the following way: we consecutively place first the columns of A indexed by the members of P_{j_0}, then the columns of A indexed by the members of P_{j_1}, and so on, finally we place the columns of A indexed by the members of $P_{j_{m-1}}$. Then for any sequence $J = (j_0, \dots, j_{m-1})$ we have

$$\sum_{\sigma \in S_r} \operatorname{sgn} \sigma \prod_{i=0}^{m-1} A\big(\sigma(k \cdot i + 1), k \cdot j_i + 1\big) \cdots A\big(\sigma(k \cdot i + k), k \cdot j_i + k\big)$$

$$= \sum_{\sigma \in S_r} \operatorname{sgn} \sigma \prod_{i=1}^{r} \tilde{A}_J\big(\sigma(i), i\big) = \det \tilde{A}_J.$$

Therefore we have

$$\operatorname{PF} C = \frac{1}{m!} \sum_{J=(j_0,\dots,j_{m-1})} \det \tilde{A}_J,$$

where the sum is taken over all sequences $0 \leqslant j_0, \dots, j_{m-1} \leqslant n-1$. If a sequence $J = (j_0, \dots, j_{m-1})$ contains a pair of equal numbers, then the corresponding summand is equal to zero, since the matrix $\tilde{A}_J$ contains a pair of identical columns. A transposition of any two terms of a given sequence J results in k transpositions of the columns of the matrix $\tilde{A}_J$. Since k is even, all the summands corresponding to the $m!$ different orderings of a given set $\{j_0, \dots, j_{m-1}\} \subset [0 : n-1]$ are equal. Therefore we get

$$\operatorname{PF} C = \sum_{J=(j_0 < j_2 < \cdots < j_{m-1})} \det \tilde{A}_J = \sum_I \det A_I,$$

and the proof follows. □

We describe some recurrences for hyperdeterminants and hyperpfaffians which we will use later.

(3.6) Definition. Let

$$C = \{C(i_1, \dots, i_k)\colon\ 1 \leqslant i_1, \dots, i_k \leqslant r\}$$

be a k-dimensional tensor of order r. Let $I_1, I_2, \dots, I_k \subset [1 : r]$ be subsets of the set $[1 : r]$ of cardinality $t \leqslant r$. Let us define a *subtensor* A of C in the following natural manner.

Let ϕ_j, $j = 1, \dots, k$, be the unique order-preserving bijection $\phi_j : [1 : t] \longrightarrow I_j$. Thus $\phi_j(s)$ is the sth element of the set I_j in increasing order. Let

$$A(i_1, \dots, i_k) = C\big(\phi_1(i_1), \dots, \phi_k(i_k)\big),$$

for all $1 \leqslant i_1, \dots, i_k \leqslant t$. Thus A is a k-dimensional tensor of order t. We write

$$A = C(I_1, \dots, I_k),$$

referring to the chosen subsets $I_1, \dots, I_k$. If $t = 1$ and $I_1 = \{i_1\}, \dots, I_k = \{i_k\}$, then we identify the subtensor A with the number $C(i_1, \dots, i_k)$.

(3.7) Lemma. *Let C be a k-dimensional tensor of order r.*

(3.7.1) *Suppose that k is even. Then,*

$$\operatorname{DET} C = \sum_{1 \leqslant i_2, \ldots, i_k \leqslant r} (-1)^{1+i_2+\cdots+i_k} C(1, i_2, \ldots, i_k) \times \operatorname{DET} C\big([1:r] \setminus \{1\}, [1:r] \setminus \{i_2\}, \ldots, [1:r] \setminus \{i_k\}\big).$$

(3.7.2) *Suppose that k is even and that $r = km$ for some $m \in \mathbb{N}$. Then,*

$$\operatorname{PF} C = \frac{1}{m} \cdot \sum_{I=\{i_1, i_2, \ldots, i_k\}} (-1)^{(i_1-1)+(i_2-2)+\cdots+(i_k-k)} \operatorname{PF} C(I, I, \ldots, I) \times \operatorname{PF} C\big([1:r] \setminus \{I\}, \ldots, [1:r] \setminus \{I\}\big),$$

where the sum is taken over all k-subsets $I = \{i_1, \ldots, i_k\}$ of the set $[1:r]$.

Proof. Formula (3.7.1) is not new (see, for example, [15]), but for the sake of completeness we give its proof here. Using (3.2.2), we get

$$\begin{aligned}
\operatorname{DET} C &= \sum_{\sigma_2, \ldots, \sigma_k \in S_r} \operatorname{sgn} \sigma_2 \cdots \operatorname{sgn} \sigma_k \prod_{i=1}^{r} C\big(i, \sigma_2(i), \ldots, \sigma_k(i)\big) \\
&= \sum_{\sigma_2, \ldots, \sigma_k \in S_r} \operatorname{sgn} \sigma_2 \cdots \operatorname{sgn} \sigma_k \cdot C\big(1, \sigma_2(1), \ldots, \sigma_k(1)\big) \\
&\quad \times \prod_{i=2}^{r} C\big(i, \sigma_2(i), \ldots, \sigma_k(i)\big) \\
&= \sum_{i_2, \ldots, i_k} C\big(1, i_2, \ldots, i_k\big) \sum_{\sigma_2 : \sigma_2(1)=i_2, \ldots, \sigma_k : \sigma_k(1)=i_k} \operatorname{sgn} \sigma_2 \cdots \operatorname{sgn} \sigma_k \\
&\quad \times \prod_{i=2}^{r} C\big(i, \sigma_2(i), \ldots, \sigma_k(i)\big),
\end{aligned}$$

where the outer sum is taken over all sequences $1 \leqslant i_2, \ldots, i_k \leqslant r$, whereas the inner sum is taken over the set of all permutations $\sigma_2, \ldots, \sigma_k$ such that σ_j maps 1 to i_j for $j = 2, \ldots, k$.

Let us choose $t = i_j \in [1:r]$ and let $\phi : [1:r-1] \longrightarrow [1:r] \setminus \{t\}$ be the order-preserving bijection. Then each permutation $\sigma \in S_r$ such that $\sigma(1) = t$ corresponds to the permutation $\tau \in S_{r-1}$ defined by the formula $\tau(s) = \phi^{-1}\big(\sigma(s+1)\big)$ for all $s \in [1:r-1]$. If σ ranges over the set of permutations which map 1 to t, then τ ranges over the group S_{r-1}. Moreover, since ϕ^{-1} is order-preserving, the number of inversions in the string $\big(\tau(1), \ldots, \tau(r-1)\big)$ is equal to the number of inversions in the string $\big(\sigma(2), \ldots, \sigma(r)\big)$ which is equal to $\operatorname{inv}(\sigma) - t - 1$ (since $\sigma(1) = t$). Therefore for any sequence $i_2, \ldots, i_k$ and $A = C\big([1:r] \setminus \{1\}, [1:r] \setminus \{i_2\}, \ldots, [1:r] \setminus \{i_k\}\big)$ we get (we recall that k is even)

$$\sum_{\sigma_2:\,\sigma_2(1)=i_2,\dots,\sigma_k:\,\sigma_k(1)=i_k} \operatorname{sgn}\sigma_2\cdots\operatorname{sgn}\sigma_k \prod_{i=2}^{r} C\big(i,\sigma_2(i),\dots,\sigma_k(i)\big)$$
$$= (-1)^{1+i_2+\cdots+i_k} \sum_{\tau_2,\dots,\tau_k\in S_{r-1}} \operatorname{sgn}\tau_2\cdots\operatorname{sgn}\tau_k \prod_{i=1}^{r-1} A\big(i,\tau_2(i),\dots,\tau_k(i)\big)$$
$$= (-1)^{1+i_2+\cdots+i_k}\,\mathrm{DET}\,A,$$

and the proof of (3.7.1) follows.

Let us prove (3.7.2). We observe that by (3.4.1),

$$\mathrm{PF}\,C = \frac{1}{m!}\sum_{\sigma\in S_r}\operatorname{sgn}\sigma\prod_{i=0}^{m-1} C\big(\sigma(ki+1),\sigma(ki+2),\dots,\sigma(ki+k)\big)$$
$$= \frac{1}{m!}\sum_{\sigma\in S_r}\operatorname{sgn}\sigma\cdot C\big(\sigma(1),\dots,\sigma(k)\big)$$
$$\times\prod_{i=1}^{m-1} C\big(\sigma(ki+1),\sigma(ki+2),\dots,\sigma(ki+k)\big)$$
$$= \frac{1}{m!}\sum_{I=(i_1,\dots,i_k)} C(i_1,\dots,i_k) \sum_{\sigma:\,\sigma(1)=i_1,\dots,\sigma(k)=i_k}\operatorname{sgn}\sigma$$
$$\times\prod_{i=1}^{m-1} C\big(\sigma(ki+1),\sigma(ki+2),\dots,\sigma(ki+k)\big), \qquad (*)$$

where the outer sum is taken over all sequences of pairwise different numbers $I=(1\leqslant i_1,\dots,i_k\leqslant r)$ whereas the inner sum is taken over the set of all permutations $\sigma\in S_r$ such that $\sigma(1)=i_1,\dots,\sigma(k)=i_k$. Let us choose a sequence $I=(i_1,\dots,i_k)$. Let $\phi:[1:r-k]\longrightarrow[1:r]\setminus\{I\}$ be the order-preserving bijection. To each permutation $\sigma\in S_r$ which maps 1 to i_1, 2 to $i_2,\dots,k$ to i_k we let correspond a permutation $\tau\in S_{r-k}$ defined by the formula $\tau(s)=\phi^{-1}\big(\sigma(s+k)\big)$ for all $s\in[1:r-k]$. Since ϕ^{-1} is order-preserving, the number of inversions in the string $\big(\tau(1),\dots,\tau(r-k)\big)$ is equal to the number of inversions in the string $\big(\sigma(k+1),\sigma(k+2),\dots,\sigma(r)\big)$. Let us compute the last number. For $s=1,\dots,k$ let us denote $l_\sigma(s)=\big|\{i_j\colon j<s \text{ and } i_j<i_s\}\big|$. Then,

$$\operatorname{inv}\big(\sigma(k+1),\dots,\sigma(r)\big)=\operatorname{inv}(\sigma)-\sum_{s=1}^{k}\big(i_s-1-l_\sigma(s)\big).$$

On the other hand, we have that

$$\operatorname{inv}(i_1,\dots,i_k)=\sum_{s=1}^{k}\big(s-1-l_\sigma(s)\big),$$

and therefore

$$\operatorname{inv}(\tau) = \operatorname{inv}(\sigma) - \operatorname{inv}(i_1, \dots, i_k) - \sum_{s=1}^{k} (i_s - s).$$

Therefore, letting $A = C\big([1:r] \setminus \{I\}, \dots, [1:r] \setminus \{I\}\big)$, we get

$$\begin{aligned}
&\sum_{\sigma:\, \sigma(1)=i_1, \dots, \sigma(k)=i_k} \operatorname{sgn} \sigma \prod_{i=1}^{m-1} C\big(\sigma(ki+1), \dots, \sigma(ki+k)\big) \\
&\quad = (-1)^{(i_1-1)+\dots+(i_k-k)} (-1)^{\operatorname{inv}(i_1,\dots,i_k)} \sum_{\tau \in S_{r-k}} \operatorname{sgn} \tau \\
&\qquad \times \prod_{i=0}^{m-2} A\big(\tau(ki+1), \dots, \tau(ki+k)\big) \\
&\quad = (m-1)! \cdot (-1)^{(i_1-1)+\dots+(i_k-k)+\operatorname{inv}(i_1,\dots,i_k)} \operatorname{PF} A.
\end{aligned}$$

Now we observe that for any given subset $I \subset [1:r]$ of cardinality k the sum

$$\sum (-1)^{\operatorname{inv}(i_1,\dots,i_k)} C(i_1, \dots, i_k)$$

taken over all $k!$ permutations of the set $I = \{i_1, \dots, i_k\}$ is equal to $\operatorname{PF} C(I, \dots, I)$. From $(*)$ we deduce the desired formula. $\square$

4. The algorithms

In this section we describe our algorithms for the k-Matroid Intersection Problem and for the Matroid k-Parity Problem. We begin with the algorithms that compute hyperdeterminants and hyperpfaffians.

(4.1.1) Computing the hyperdeterminant.

Let us fix an even $k \in \mathbb{N}$.

Input: A natural number r and a k-dimensional tensor C of order r:

$$C = \{C(i_1, \dots, i_k):\ 1 \leqslant i_1, \dots, i_k \leqslant r\},$$

where all the numbers $C(i_1, \dots, i_k)$ are integral.

Output: The integral number $\operatorname{DET} C$.

Algorithm: We use dynamic programming based on the recurrence (3.7.1). For any k-tuple of nonempty subsets $I_1, \dots, I_k$ of the set $[1:r]$ such that $|I_1| = \dots = |I_k|$ let us define a variable $x(I_1, \dots, I_k)$. The common cardinality of the sets $I_1, \dots, I_k$ we call the *level* of x. If the level of x is 1, and therefore $I_1 = \{i_1\}, \dots, I_k = \{i_k\}$, for some $i_1, \dots, i_k \in [1:r]$, we let

$$x(I_1, \dots, I_k) = C(i_1, \dots, i_k).$$

For $s = 2, \dots, r$ we consecutively compute the values of variables of level s using previously computed values of variables whose level is $s-1$. Let $\phi_t(i)$, $t = 1, \dots, k$, $i = 1, \dots, s$, denote the ith element of the set I_t in increasing order. Let

$$x(I_1, \dots, I_k) = \sum_{1 \leqslant i_2, \dots, i_k \leqslant s} (-1)^{1+i_2+\dots+i_k} C\big(\phi_1(1), \phi_2(i_2), \dots, \phi_k(i_k)\big) \times x\big(I_1 \setminus \{\phi_1(1)\}, I_2 \setminus \{\phi_2(i_2)\}, \dots, I_k \setminus \{\phi_k(i_k)\}\big).$$

Finally let

$$\operatorname{DET} C = x\big([1:r], \dots, [1:r]\big).$$

(4.1.2) Proposition. *The algorithm of* (4.1.1) *computes the hyperdeterminant of a given k-dimensional tensor C of order r using (for a fixed k) $\mathrm{O}(2^{rk} \cdot r^k)$ arithmetic operations. The sizes of the numbers involved in the algorithm are bounded by a polynomial in the input size.*

Proof. By recurrence (3.7.1) it follows that $x(I_1, \dots, I_k) = \operatorname{DET} C(I_1, \dots, I_k)$ for all subsets $I_1, \dots, I_k \subset [1:r]$ such that $|I_1| = \dots = |I_k|$. Therefore the algorithm indeed computes the desired value of $\operatorname{DET} C$. The number of various variables $x(I_1, \dots, I_k)$ does not exceed 2^{rk}. In order to compute the value of a variable we have to sum up at most r^{k-1} summands. To compute the index of a variable, that is, to delete an element with a given number from a subset $I \subset [1:r]$, it suffices to perform $\mathrm{O}(r)$ operations. Thus the algorithm has the desired complexity.

Let us denote by L the maximal absolute value of $C(i_1, \dots, i_k)$ for $1 \leqslant i_1, \dots, i_k \leqslant r$. By (3.2.2) it follows that all the absolute values of $\operatorname{DET} C(I_1, \dots, I_k)$ are bounded by $(r!)^{k-1} \cdot L^r$. Therefore the sizes of all the numbers involved in the algorithm are bounded by a polynomial in the input size (we note that the input size is at least $r^k + \text{size } L$). □

(4.2.1) Computing the hyperpfaffian.

Let us fix an even $k \in \mathbb{N}$.

Input: A natural number r, $r = km$ for some $m \in \mathbb{N}$ and a k-dimensional tensor C of order r:

$$C = \{C(i_1, \dots, i_k) \colon 1 \leqslant i_1, \dots, i_k \leqslant r\},$$

where all the numbers $C(i_1, \dots, i_k)$ are integral.

Output: The integral number $\operatorname{PF} C$.

Algorithm: We use dynamic programming based on the recurrence (3.7.2). For any nonempty subset I of the set $[1:r]$ such that $|I|$ is divisible by k, let us define a variable $x(I)$. The number $|I|/k$ we call the *level* of x. Let $\phi(i)$, $i \in [1:|I|]$, denote the ith element of the set I in increasing order. If the level of x is 1, we let

$$x(I) = \sum_{\sigma \in S_k} \operatorname{sgn} \sigma \, C\big(\sigma(\phi(1)), \dots, \sigma(\phi(k))\big).$$

For $s = 2, \dots, m$ we consecutively compute the values of variables of level s in terms of variables whose level is $s-1$:

$$x(I) = \frac{1}{s} \cdot \sum_{J \subset [1:ks],\, J=\{i_1,\dots,i_k\}} (-1)^{(i_1-1)+(i_2-2)+\dots+(i_k-k)} x\big(\{\phi(i_1),\dots,\phi(i_k)\}\big) \times x\big(I \setminus \{\phi(i_1),\dots,\phi(i_k)\}\big),$$

where the sum is taken over all k-subsets J of the set $[1:ks]$.

Finally let

$$\mathrm{PF}\, C = x\big([1:r]\big).$$

(4.2.2) Proposition. *The algorithm of* (4.2.1) *computes the hyperpfaffian of a given k-dimensional tensor C of order r using (for a fixed k) $O(2^r \cdot r^{k+1})$ arithmetic operations. The sizes of the numbers involved in the algorithm are bounded by a polynomial in the input size.*

Proof. By the recurrence (3.7.2) it follows that $x(I) = \mathrm{PF}\, C(I,\dots,I)$ for all nonempty subsets $I \subset [1:r]$ such that $|I|$ is divisible by k. The number of various variables $x(I)$ does not exceed 2^r. To compute the value of a variable of level 1 we have to perform a constant number of arithmetic operations (since k is fixed). If the level of a variable is bigger than 1, we have to sum up not more than r^k summands. To compute the index of a variable it suffices to perform $O(r)$ operations. Thus the algorithm has the desired complexity.

Let us denote by L the maximal absolute value of $C(i_1,\dots,i_k)$ for $1 \leqslant i_1,\dots,i_k \leqslant r$. By (3.4.1) it follows that all the absolute values of $\mathrm{PF}\, C(I,\dots,I)$ are bounded by $r! \cdot L^m$. Therefore the sizes of all the numbers involved in the algorithm are bounded by a polynomial in the input size. $\square$

Now we can complete our algorithms for the k-Matroid Intersection Problem and for the Matroid k-Parity Problem.

(4.3.1) Algorithm for Problem 1.1.

Let us compute a $2k$-dimensional tensor C of order r by the formula

$$C(i_1, i_2, \dots, i_{2k-1}, i_{2k}) = \sum_{j=1}^{n} \prod_{s=1}^{k} \big(A^s(i_{2s-1}, j) \cdot A^s(i_{2s}, j)\big),$$

for all $1 \leqslant i_1,\dots,i_{2k} \leqslant r$. Using (4.1.1), let us compute an integer

$$D = \mathrm{DET}\, C.$$

If $D \neq 0$, then there exists a common base of the matroids represented by $A^1,\dots,A^k$, and if $D = 0$, then no such base exists.

(4.3.2) Theorem. *Algorithm* 4.3.1 *solves Problem* 1.1 *using (for a fixed* k*)* $O(r^{2k} \cdot (4^{rk}+n))$ *arithmetic operations. The sizes of the numbers involved in the algorithm are bounded by a polynomial in the input size.*

Proof. By Lemma 3.3 we conclude that

$$D = \sum_I (\det A_I^1)^2 \cdots (\det A_I^k)^2,$$

where the sum is taken over all subsets $I \subset [1:n]$ of cardinality r. Hence we conclude that the algorithm is correct. To compute the values of $C(i_1, i_2, \ldots, i_{2k-1}, i_{2k})$ we need $2nk \cdot r^{2k}$ operations. By Proposition 4.1.2 we conclude that the algorithm has the desired complexity. □

(4.4.1) Algorithm for Problem 1.2.

Let us define a $2r \times 2N$ matrix B in the following way. For $j = 1, \ldots, N$ put

$$B(i, 2j-1) = \begin{cases} A(i,j), & \text{if } i \leqslant r, \\ 0, & \text{if } i > r, \end{cases} \quad \text{and} \quad B(i,2j) = \begin{cases} 0, & \text{if } i \leqslant r, \\ A(i-r, j), & \text{if } i > r. \end{cases}$$

Let us compute a $2k$-dimensional tensor C of order $2r$ by the formula

$$C(i_1, i_2, \ldots, i_{2k-1}, i_{2k}) = \sum_{j=0}^{n-1} \prod_{s=1}^{2k} B(i_s, 2kj + s),$$

for all $1 \leqslant i_1, \ldots, i_{2k} \leqslant 2r$. Using (4.2.1) let us compute an integer

$$D = \operatorname{PF} C.$$

If $D \neq 0$, then there exists a base $I \subset [1:N]$ represented in the desired form $I = P_{i_1} \cup \cdots \cup P_{i_m}$, and if $D = 0$, then no such base exists.

(4.4.2) Theorem. *Algorithm* 4.4.1 *solves Problem* 1.2 *using (for a fixed* k*)* $O(r^{2k+1} \cdot (4^r + n))$ *arithmetic operations. The sizes of the numbers involved in the algorithm are bounded by a polynomial in the input size.*

Proof. Let us consider the partition of the set $[1:2N]$ into disjoint $2k$ subsets $Q_0 = \{1, \ldots, 2k\}, Q_1 = \{2k+1, \ldots, 4k\}, \ldots, Q_{m-1} = \{2N-2k+1, \ldots, 2N\}$ together with the initial partition of the set $[1:N]$ into disjoint k-subsets $P_0 = \{1, \ldots, k\}, P_1 = \{k+1, \ldots, 2k\}, \ldots, P_{m-1} = \{N-k+1, \ldots, N\}$. Applying Lemma 3.5 (with the Q_i here playing the role of the P_i in the lemma) we conclude that

$$D = \sum_J \det B_J,$$

where the sum is taken over subsets $J \subset [1:2N]$ of cardinality $2r$ that can be represented in the form $J = Q_{i_1} \cup \cdots \cup Q_{i_m}$, where $0 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n-1$.

For such $J = Q_{i_1} \cup \cdots \cup Q_{i_m}$ let us consider the corresponding r-subset $I \subset [1 : N]$, $I = P_{i_1} \cup \cdots \cup P_{i_m}$. We claim that $\det B_J = \epsilon \cdot (\det A_I)^2$ with $\epsilon \in \{-1, 1\}$ depending on r alone. Indeed, let $\pi \in S_{2r}$ be a permutation such that $\pi(2i-1) = i$ and $\pi(2i) = i+r$ for $i = 1, \ldots, r$. Applying the permutation π to the columns of B_J we get a $2r \times 2r$ matrix having A_I as diagonal blocks and zeros elsewhere. Thus we can choose $\epsilon = \operatorname{sgn} \pi$. Hence we conclude that

$$D = \epsilon \cdot \sum_I (\det A_I)^2,$$

where the sum is taken over all r-subsets $I \subset [1 : N]$ represented in the form $I = P_{i_1} \cup \cdots \cup P_{i_m}$ for some $0 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n-1$. Therefore the algorithm is correct.

To compute the values of $C(i_1, \ldots, i_{2k})$ we need $2nk(2r)^{2k}$ arithmetic operations. By Proposition 4.2.2 we conclude that the algorithm has the desired complexity. □

The main feature of our algorithms is that their complexity is linear in the cardinality of the given matroids. If both r and k are fixed, then our algorithms solve Problems 1.1 and 1.2 in $O(n)$ time. An exhaustive search for Problems 1.1 and 1.2 requires $O(r^3 \cdot \binom{n}{r})$ arithmetic operations. If n grows faster than any linear function of r, then Algorithms 4.3.1 and 4.4.1 are more efficient. This is the case for many combinatorial applications (see also Section 5). Moreover, if we restrict ourselves to a class of problems with $r = O(\log n)$, then both Algorithms 4.3.1 and 4.4.1 have polynomial complexity. More precisely, the following result holds.

(4.5) Corollary. *Let us fix $c > 0$. Let us consider a class of Problems* 1.1 *and* 1.2 *where $r \leqslant c \cdot \log n$. Then this class of problems admits a polynomial time algorithm.*

One can find that the condition $r = O(\log n)$ for polynomial solvability of Problems 1.1 and 1.2 is too strong. However, if we choose instead, say, $r = O(n^\epsilon)$ for some fixed $\epsilon > 0$, then the problems remain NP-hard since we can reduce the general problem to a problem with $r = O(n^\epsilon)$ by appending columns of zeros. Therefore we have little hope to solve Problem 1.1 or 1.2 in polynomial time unless n is exponentially bigger than r. A natural question in this context is to explore the case $r = O(\log^\epsilon n)$ for some $\epsilon > 1$. In general, using the construction of *truncation*, we can test in polynomial time the existence of a common *independent set* (that is, a subset of a base) of a reasonably small size in matroids. In Section 5 we give some examples where the truncation can be computed efficiently.

5. Combinatorial applications and examples

In this section we apply our algorithms to some special problems, namely, to the k-Dimensional Assignment Problem and to the k-Dimensional Matching Problem. Both of them are polynomially solvable if $k = 2$ (see [9,12]) and NP-hard if $k > 2$ (see

[7]). We also express the number of Hamiltonian paths in a directed graph as a certain hyperdeterminant. First we discuss some particular matroids.

(5.1) Transversal Matroid and its Truncation. Suppose that the set $\{1,\dots,n\}$ is represented as a disjoint union of nonempty subsets $U_1,\dots,U_t \subset [1:n]$. Let us choose $r \leqslant t, r \in \mathbb{N}$. Let us define an $r \times n$ integral matrix A by

$$A(i,j) = q^i, \quad \text{if } j \in U_q.$$

A subset $I \subset [1:n]$ of cardinality r is a base of the matroid represented by A if and only if the intersection of I with each set $U_1,\dots,U_t$ consists of at most a single element. Indeed, if I contains a pair of elements from the same set U_q, then the submatrix A_I contains a pair of identical columns and therefore is singular. If I contains not more than one element from each set U_q, then $\det A_I \neq 0$ as Vandermonde's determinant (see, for example, [13]). Thus the matroid represented by A is the truncation at r of the transversal matroid associated with the partition $U_1 \cup \cdots \cup U_t = [1:n]$ (see, for example, [16]).

If $r = t$, then A represents the *transversal matroid* associated with the partition $U_1 \cup \cdots \cup U_t = [1:n]$ (see [16]).

(5.2) Cycle Matroid. Let $V = [1 : r+1]$ be the set of vertices and E be the set of edges of a directed connected graph $G = (V,E)$ without loops. Let $n = |E|$ and label the edges by the numbers $1,\dots,n$. Let us define an $r \times n$ matrix $A = \{A(i,j)\colon 1 \leqslant i \leqslant r, 1 \leqslant j \leqslant n\}$:

$$A(i,j) = \begin{cases} 1, & \text{if } i \text{ is the tail of the edge } j,\\ -1, & \text{if } i \text{ is the head of the edge } j,\\ 0, & \text{otherwise.} \end{cases}$$

The matrix A represents a matroid, called the *cycle matroid* of the graph G. As is known (see, for example, [16]), $I \subset E$ is a base if and only if I is the set of edges of a spanning tree in G.

Let us consider a particular case of the k-Matroid Intersection Problem.

(5.3) k-Dimensional Assignment Problem. Let us fix $k \in \mathbb{N}$. Consider the k-dimensional integral cube $[1:t]^k$. A set of the form $M_j(q) = \{(i_1,\dots,i_k) \in [1:t]^k\colon i_j = q\}$ is called a *section*. We are interested in the following k-Dimensional Assignment Problem:

For a given $t \in \mathbb{N}$, $U \subset [1:t]^k$, and $r \in \mathbb{N}$, decide whether there exist r distinct points from U such that no two of them belong to the same section.

As is known, this problem can be solved as an instance of Problem 1.1. We present a particular construction here.

(5.3.1) Algorithm. Let $n = |U|$ and identify the set U with the interval $[1:n]$. For $j = 1,\ldots,k$, let Q_j be the set of $q \in [1:t]$ such that the intersection $U \cap M_j(q)$ is nonempty and let

$$U = \bigcup_{q \in Q_j} (U \cap M_j(q))$$

be the corresponding partition of the set U. If $|Q_j| < r$ for some j, then the answer is "no". Otherwise, let us construct the $r \times n$ matrix A^j that represents the transversal matroid truncated at r associated with the partition (see (5.1)). Then apply Algorithm 4.3.1 with the constructed matrices $A^1,\ldots,A^k$.

Theorem 4.3.2 implies the following result.

(5.3.1) Proposition. *Let us fix $c > 0$. If $r \leqslant c \cdot \log|U|$, then Algorithm* 5.3.1 *solves the k-Dimensional Assignment Problem in polynomial time.*

We also note that if we fix both r and k, then the k-Dimensional Assignment Problem can be solved in $O(|U| + t)$ time.

(5.4) *k*-Dimensional Matching Problem. Let us fix $k \in \mathbb{N}$. Let $E = \{e_1,\ldots,e_n\}$ be a family of k-subsets of the set $V = [1:t]$. Such an object $H = (V, E)$ is called a *uniform k-hypergraph*. The elements of the set V are called *vertices* and the elements of the set E are called *edges*. We assume that $e_1 \cup \cdots \cup e_n = V$, that is, every vertex is covered by an edge. We are interested in the following k-Dimensional Matching Problem.

For a given $t \in \mathbb{N}$, uniform k-hypergraph $H = (V, E)$ on vertex set $V = [1:t]$, and $r \in \mathbb{N}$, decide whether there exist r pairwise vertex-disjoint edges from E.

If $k = 2$, then we have the ordinary matching problem in a graph which admits a polynomial time algorithm (see [12]). As we mentioned, the corresponding problem is NP-complete for $k > 2$ [7]. As is known, this problem can be solved as an instance of Problem 1.2. We present a particular construction here.

(5.4.1) Algorithm. Let $N = kn$. Let us construct a string $f_1,\ldots,f_N$ of numbers $f_j \in [1:t]$ as follows. Consecutively list first the k vertices of e_1 in increasing order, then the k vertices of e_2 in increasing order, and so on; finally list the k vertices of e_n in increasing order. For $i = 1,\ldots,t$, let us define a subset $U_i \subset [1:N]$ as follows. Let j be an element of U_i if and only if $f_j = i$. Let us construct the $r \times N$ matrix A representing the transversal matroid truncated at r associated with the partition $[1:N] = \bigcup_{i \in [1:t]} U_i$. Then apply Algorithm 4.4.1 with the matrix A.

Theorem 4.4.2 implies the following result.

(5.4.1) Proposition. *Let us fix $c > 0$. If $r \leqslant c \cdot \log|E|$, then Algorithm* 5.4.1 *solves the k-Dimensional Matching Problem in polynomial time.*

Again we note that if we fix both k and r, then the k-Dimensional Matching Problem can be solved in $O(|E|)$ time.

Since the k-Matroid Intersection Problem can be reduced to the computation of the hyperdeterminant of a $2k$-dimensional tensor (see Section 4), one can easily derive that to decide if the hyperdeterminant of a k-dimensional tensor is zero is an NP-hard problem for $k \geqslant 6$. We will show that this problem is NP-hard already for $k = 4$ in contrast to the case $k = 2$.

(5.5) Hamiltonian paths in graphs and hyperdeterminants. Let $G = (V, E)$ be a directed graph without loops with the set of vertices V and the set of edges E. Let $r = |V| - 1$ and identify the set V with the interval $[1 : r + 1]$. Furthermore, let $n = |E|$ and label the edges from E by the numbers $1, \dots, n$. Let us assume that the following conditions hold:

(i) each vertex except $r + 1$ is the tail of an edge;

(ii) each vertex except 1 is the head of an edge.

Let us introduce $r \times n$ matrices A^{b} and A^{e} as follows: for $i = 1, \dots, r$ and $j = 1, \dots, n$ let

$$A^{\mathrm{b}}(i,j) = \begin{cases} 1, & \text{if the vertex } i \in [1 : r] \text{ is the tail of edge } j, \\ 0, & \text{otherwise,} \end{cases}$$

$$A^{\mathrm{e}}(i,j) = \begin{cases} 1, & \text{if the vertex } i + 1 \text{ is the head of edge } j, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, let A^{c} be the $r \times n$ matrix representing the cycle matroid of the graph G (see (5.2)).

We observe that a set $I \subset [1 : n]$ is a common base of the matroids represented by A^{b}, A^{e} and A^{c} if and only if I is the set of edges of a directed path starting at 1, visiting each vertex exactly once, and arriving to $r + 1$. Such a path is called a *Hamiltonian path* from the vertex 1 to the vertex $r + 1$ (see also [16, Chapter 8, Section 5]). Let us define a four-dimensional tensor C of order r as follows:

$$C(i_1, i_2, i_3, i_4) = \sum_{j=1}^{n} A^{\mathrm{b}}(i_1, j) \cdot A^{\mathrm{e}}(i_2, j) \cdot A^{\mathrm{c}}(i_3, j) \cdot A^{\mathrm{c}}(i_4, j),$$

for all $1 \leqslant i_1, i_2, i_3, i_4 \leqslant r$.

(5.5.1) Proposition. *The value of* $\operatorname{DET} C$ *is equal to the number of directed Hamiltonian paths in* G *starting at the vertex* 1 *and arriving to the vertex* $r + 1$.

Proof. We use Lemma 3.3. We have

$$\operatorname{DET} C = \sum_{I} \det A^{\mathrm{b}}_I \cdot \det A^{\mathrm{e}}_I \cdot \left(\det A^{\mathrm{c}}_I\right)^2,$$

where the sum is taken over all subsets $I \subset [1:n]$ of cardinality r. As we mentioned, the corresponding summand is equal to 0 unless I is the set of edges of a directed Hamiltonian path starting at 1 and arriving to $r+1$. Moreover, since the matrix A^{c} is totally unimodular, we conclude that $(\det A_I^{\mathrm{c}})^2 = 1$ for such I (see, for example, [8] for the incidence matrix of a graph). Furthermore,

$$\det A_I^{\mathrm{b}} \cdot \det A_I^{\mathrm{e}} = \det\left(A_I^{\mathrm{b}} \cdot (A_I^{\mathrm{e}})^{\mathrm{T}}\right),$$

where "T" denotes the transpose. For a given Hamiltonian path I we consider the Hamiltonian cycle $\sigma \in S_{r+1}$ which maps each $i \in [1:r]$ to the next vertex $i \in [1:r+1]$ along this path and, additionally, maps $r+1$ onto 1. Let Π_σ be the $(r+1)\times(r+1)$ matrix of this permutation, namely

$$\Pi_\sigma(i,j) = \begin{cases} 1, & \text{if } \sigma(i) = j, \\ 0, & \text{otherwise.} \end{cases}$$

If we delete the $(r+1)$th row and the first column of the matrix Π_σ, then we get the matrix $A_I^{\mathrm{b}} \cdot (A_I^{\mathrm{e}})^{\mathrm{T}}$. Therefore

$$\det\left(A_I^{\mathrm{b}} \cdot (A_I^{\mathrm{e}})^{\mathrm{T}}\right) = (-1)^r \cdot \det \Pi_\sigma = (-1)^r \cdot \operatorname{inv} \sigma = 1,$$

and the proof follows. □

(5.5.2) Corollary. *Let us fix an even number $k > 2$. The problem of deciding whether for a given k-dimensional tensor C with integral entries the hyperdeterminant* DET C *is equal to* 0 *is NP-hard.*

6. Remarks

The results of this paper can be generalized in at least two directions.

First, we can consider matroids represented over a different field. In case of the field of complex numbers one can design algorithms similar to 4.3.1 and 4.4.1. In Algorithm 4.3.1 we should adjoin the complex conjugate of each matrix A^i (but not just a copy as in the case of the reals). The matrix B in Algorithm 4.4.1 should be modified in a similar way using complex conjugation. In case of an arbitrary field one can use the "perturbation" described in Section 2 with nonzero elements $t_1, \ldots, t_n$ from the field or from its algebraic extension. This leads to probabilistic algorithms in the k-Matroid Intersection Problem and the Matroid k-Parity Problem; the author does not know, however, whether it is possible to design deterministic algorithms with similar bounds of complexity in case of an arbitrary field. Our methods are not applicable to nonrealizable matroids, given by their oracles. We also note that for general matroids already the usual ($k = 2$) Matroid Parity Problem has exponential complexity [11].

Second, one can consider weighted versions of Problems 1.1 and 1.2. Namely, we assign integral weights to the elements of $[1:n]$ and look for a base of a maximal or given weight. Here one can use either of the (essentially equivalent) approaches developed in [3,14], or sketched in the preliminary version of this paper [2].

We do not develop these topics here since one can immediately transfer the methods used in [3,10,14] in the case $k = 2$ to the case $k > 2$ replacing identities from (2.1) and (2.2) by the identities derived in Lemmas 3.3 and 3.5 respectively.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, MA, 1974).

[2] A.I. Barvinok, "Optimization problems on matroids and exponential sums," in: E. Balas, G. Cornuéjols and R. Kannan, eds., *Integer Programming and Combinatorial Optimization*, Proceedings of the Second IPCO Conference, Carnegie Mellon University (1992) pp. 316–333.

[3] P.M. Camerini, G. Galbiati and F. Maffioli, "Random pseudo-polynomial algorithms for exact matroid problems," *Journal of Algorithms* 13 (1992) 258–273.

[4] A. Cayley, "On the theory of determinants," in: *Collected Papers, Vol. 1* (Cambridge University Press, Cambridge, 1889) pp. 63–80.

[5] A. Cayley, "On the theory of linear transformations," in: *Collected Papers, Vol. 1* (Cambridge University Press, Cambridge, 1889) pp. 80–94.

[6] G. Galbiati and F. Maffioli, "On the computation of pfaffians," *Discrete Applied Mathematics* 51 (1994) 269–275.

[7] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness* (Freeman, San Francisco, CA, 1979).

[8] F. Harary, *Graph Theory* (Addison-Wesley, Reading, MA, 1969).

[9] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart and Winston, New York, 1976).

[10] L. Lovász, "On determinants, matchings and random algorithms," in: L. Budach, ed., *Fundamentals of Computation Theory*, Proceedings of the Conference on Algebraic, Arithmetic and Categorial Methods in Computation Theory (Akademie-Verlag, Berlin, 1979) pp. 565–574.

[11] L. Lovász, "The matroid matching problem," in: L. Lovász and V.T. Sós, eds., *Algebraic Methods in Graph Theory* (North-Holland, Amsterdam, 1981) pp. 495–517.

[12] L. Lovász and M.D. Plummer, *Matching Theory* (Akadémia Kiadò, Budapest/North-Holland, Amsterdam, 1986).

[13] I. Satake, *Linear Algebra* (Marcel Dekker, New York, 1975).

[14] Yu.G. Smetanin and L.G. Khachiyan, "Use of pseudopolynomial algorithms for some problems of combinatorial optimization with constraints," *Izvestiya Akademii Nauk SSSR Tekhnicheskaya Kibernetika* (6) (1986) 139–144 (in Russian); translated in: *Soviet Journal of Computer and Systems Science* 25 (2) (1987) 161–165.

[15] N.P. Sokolov, *Spatial Matrices and their Applications* (Fizmatgiz, Moscow, 1960, in Russian).

[16] D.J.A. Welsh, *Matroid Theory* (Academic Press, London, 1976).