アルゴリズム 50－7
（1996. 3. 15）

線形マトロイド・グラフアレンジメント・半順序での数え上げ問題に対する組合せ的・幾何的アプローチ

今井浩*，岩田覚†，関根京子*，吉田研秀*

*東京大学理学系研究科情報科学専攻
†京都大学数理解析研究所

アブストラクト：グラフに関する典型的な#P 困難な問題に対して，関根，今井，谷 [12, 13] は，中規模のサイズの問題を BDD を用いて解く新しいアプローチを与えている．本論文では，それを他の代表的離散構造である線形マトロイド，グラフに対するアレンジメント，半順序に関する数え上げ問題に拡張する．まず，2 値・3 値マトロイドについては，与えられた台集合の対応のもとで同型判定が多項式時間でできることを示して，その基底を表現する BDD が出力サイズに比例する時間で構成できることを示す．この BDD を用いて，線形マトロイドの Tutte 多項式，線形符号の重み生成多項式を計算することができる．次に，実数ベクトル空間の線形マトロイドについて，対応する幾何構造であるアレンジメントを構成するアルゴリズムを用いて，その Tutte 多項式が効率よく計算できることを示す．また，それをグラフアレンジメントの問題に特化させた場合に対応するグラフの無閉路向き付け問題や半順序のイデアル問題についてもアルゴリズムを与える．

Combinatorial and Geometric Approaches to Counting Problems on Linear Matroids, Graphic Arrangements and Partial Orders

Hiroshi Imai*, Satoru Iwata†, Kyoko Sekine* and Kensyu Yoshida*

* Department of Information Science, University of Tokyo, Tokyo 113, Japan
†Research Institute of Mathematical Sciences, Kyoto University, Kyoto, Japan

Abstract: For typical #P-hard problems on graphs, we have recently proposed an approach to solve those problems of moderate size rigorously by means of the binary decision diagram, BDD [12, 13]. This paper extends this approach to counting problems on linear matroids, graphic arrangements and partial orders, most of which are already known to be #P-hard, with using geometric properties. Specifically, we show that the BDD representing all bases of a binary or ternary matroid can be computed in an output-size sensitive manner. By using this BDD, the Tutte polynomial of the matroid and the weight enumeration of an (n, k) linear code over GF(2) and GF(3) can be computed in time proportional to the size of the BDD. Next, a method of computing the Tutte polynomial of a linear matroid over the reals via the arrangement construction algorithm in computational geometry is given. Computing the number of acyclic orientations of a graph and the number of ideals in a partially ordered set is also discussed.

1 Introduction

Recently, the binary decision diagram, BDD, has been used to solve combinatorial problems such as computing the chromatic and flow polynomials of a graph, the Jones polynomial of a link, the network reliability, etc., efficiently [12, 13]. Detailed descriptions about these problems can be found in [17]. This paper generalizes this approach for linear matroids, graphic arrangements and partial orders, and present efficient algorithms for solving counting problems related to them.

Specifically, we first show that the BDD representing all bases of a binary or ternary matroid can be constructed in an output-size sensitive manner by combining the top-down algorithm in [12, 15] with the isomorphism testing of their minors under the identity map. This can be used to compute the Tutte polynomial of these matroids and the weight enumeration of a linear code on GF(2) and GF(3) in time proportional to the size of this BDD.

Next, a linear matroid over the reals is investigated. This structure is combinatorially equivalent to hyperplane arrangements which have been intensively studied from the algorithmic viewpoint in computational geometry [6]. By using the algorithm for hyperplane arrangements, we show that the Tutte polynomial of m central hyperplanes in the n-dimensional space can be constructed in $O(m^n)$ time. Regarding n as a parameter, this complexity is exponential, but this would be inevitable since the problem of computing the number of cells in the arrangement is known to be #P-complete (see [17]).

When we restrict the arrangement to that associated with a graph, the number of cells can be computed efficiently by using the result in [12]. For example, for a planar graph with n vertices, it can be computed in $O(2^{O(\sqrt{n})})$ time. The number of cells in the graphic arrangement is the number of acyclic orientations of the graph [4, 7], which is also #P-hard [17], and this result sheds light on the relation between the discrete structure of graphs and the geometric structure of arrangements. We also consider computing the number of lower dimensional faces via BDD.

Each cell of the graphic arrangement corresponds to an acyclic orientation of the given undirected graph. Given a partially ordered set, this set is naturally associated with a cell in the arrangement. The intersection of this cell with the unit hypercube is known as an order polytope [3, 17]. Its vertices correspond to ideals of this partially ordered set, and the volume gives the number of extensions of this partial order. We here show that the BDD representing all the vertices of the order polytope can be constructed by the top-down algorithm, and, when a graph has a good vertex elimination ordering, the BDD can be made to be compact. For example, when the partial order corresponds to a planar acyclic graph with n vertices, the number of ideals of the partial order can be computed in $O(2^{O(\sqrt{n})})$ time. It is conjectured that this approach may be extended to counting the number of linear extensions of the partial order, which is also a very fundamental #P-hard problem [3].

2 Preliminaries

2.1 BDD of a Boolean function

An OBDD represents a Boolean function of m variables $x_1, \ldots, x_m$ by a labeled acyclic graph with a single source (root) and two sinks (0-node and 1-node). Each node besides the sinks has two edges emanating from it, one is labeled as 0-edge and the other 1-edge. The sink nodes are labeled as 0 and 1, and are called the 0-node and 1-node, respectively. All the directed paths from the source to the sinks have the same number of edges, and the level of a node is defined to be the number of edges of directed paths from the source to the node. The level of the source is 0, and that of the sink is m. Nodes in the $(i-1)$-th level $(i = 1, \ldots, m)$ correspond to a variable x_i. Each directed path from the root to the 1-node corresponds one-to-one to an assignment of x_i to the label of the edge, emanating from the node of x_i on this path $(i = 1, \ldots, m)$, with which the function value is 1. The size of OBDD is the total number of nodes, and varies by the ordering of variables. We denote the Boolean AND, OR, NOT by $\bigwedge$, $\bigvee$, $\overline{x_i}$, respectively.

We now give an example of BDD. Let $G =$

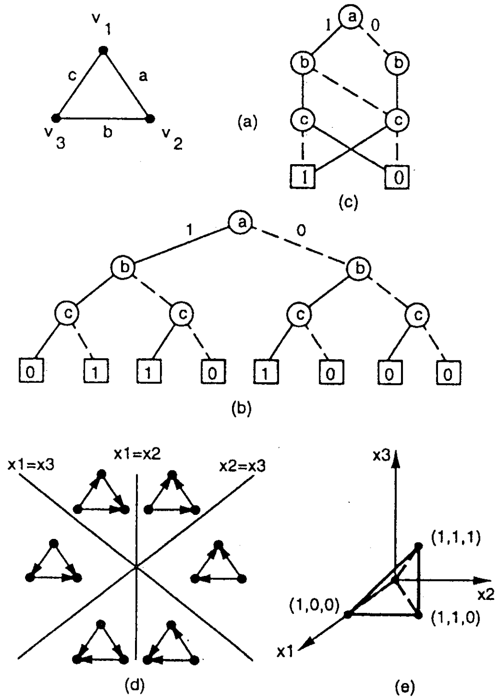

Figure 1: (a) K_3, (b) an OBDD-like complete binary tree representing all the spanning trees, (c) the QOBDD representing all the spanning trees, (d) the graphic arrangement for K_3, and (e) the order polytope for an acyclic orientation of K_3 in the bottom left in (d) where vertices correspond to ideals $\emptyset$, $\{v_3\}$, $\{v_2, v_3\}$, $\{v_1, v_2, v_3\}$

(V, E) be a simple connected undirected graph with a vertex set V and an edge set E. For each edge e_i in E, we associate a Boolean variable x_i ($i = 1, \ldots, |E|$). Then, consider a Boolean function representing all the spanning trees, i.e., it becomes 1 when edges e_i with $x_i = 1$ form a spanning tree, and 0 otherwise. In Fig.1, (a) is a complete graph K_3 of 3 vertices, and (b) is a binary tree representing all the spanning trees (this is an OBDD when the 0-node and 1-node are shared). In (c), all the isomorphic subtrees are shared, and is a canonical OBDD, called QOBDD. (d) explains a graphic arrangement for K_3, and (e) is an order polytope for an acyclic orientation of K_3, both mentioned in the introduction. For these two, we will explain them in more detail later.

Given the BDD of a Boolean function f, the number of truth assignments which make $f = 1$ can be computed in time proportional to the size of BDD (in the above example, the number of spanning trees).

As for the construction of BDD, the following is shown in [15].

Lemma 1 *[15] The BDD of a Boolean function f of m variables can be computed in a top-down fashion in time proportional to the size of BDD times some polynomial of m if the equivalence can be checked for two subfunctions obtained from f by setting the values of some variables to 0,1 in polynomial time in m.*

2.2 Tutte polynomial of a matroid

We will consider a linear matroid M on a finite set E. For matroids, see [4, 11, 16]. The most typical linear matroid is that over the reals. Given a set E of m vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$ in $\mathbf{R}^n$, linear independence among these vectors induces a linear matroid $\mathrm{M}(E)$ of vectors in E. The rank function $\rho: 2^E \to \mathbf{Z}$ of $\mathrm{M}(E)$ is defined by $\rho(A) = \dim(\{\boldsymbol{a}_i \mid \boldsymbol{a}_i \in A\})$ $(A \subseteq E)$. The linear matroid $\mathrm{M}(E)$ of vectors $\boldsymbol{a}_i \in E$ can be regarded as that of the arrangement of hyperplanes $h_i = \{\boldsymbol{x} \mid \boldsymbol{a}_i \cdot \boldsymbol{x} = 0\}$ $(i = 1, \ldots, m)$ in the dual $\mathbf{R}^n$.

The Tutte polynomial $T(\mathrm{M}; x, y)$ of matroid M on E is a two-variable polynomial. By the rank function ρ, it is defined by

$$T(\mathrm{M}; x, y) = \sum_{A \subseteq E} (x-1)^{\rho(E)-\rho(A)} (y-1)^{|A|-\rho(A)}.$$

There exists a recursive formula for the Tutte polynomial based on the deletion/contraction of an element from a matroid. From the recursive formula, the Tutte polynomial of a matroid can be computed by using the BDD representing all bases of the matroid. In [12], the case for graphic matroids is described, and also note that Fig.1 is such an example with spanning trees as bases.

The original definition of the Tutte polynomial by Tutte is expressed as the summation over all bases of a matroid. To describe this, we need more definitions. Let B be a base of matroid M. For $e \in E - B$, a minimal dependent set of $B \cup \{e\}$, including e, is uniquely determined, which is called the fundamental circuit of e with respect to B. For $e \in B$, $\{e' \in E \mid (B - \{e\}) \cup \{e'\}$ is a base$\}$ is called the fundamental cutset of e with respect to B. Given an ordering $e_1, e_2, \ldots e_m$ of elements of E, $e_i \in E - B$ is called externally active if its fundamental circuit with respect to B consists of e_j with $j \leq i$. $e_i \in B$ is called internally active if its fundamental cutset with respect to B consists of e_j with $j \leq i$. Then, for B, the external activity $r(B)$ is the number of external active elements, and the internal activity $s(B)$ is the number of internal active elements. Then, for this ordering, the Tutte polynomial is given by

$$T(\mathrm{M}; x, y) = \sum_{B:\text{ bases of } \mathrm{M}} x^{r(B)} y^{s(B)}.$$

The Tutte polynomial of a matroid has many meanings. For example, $T(\mathrm{M}; 1, 1)$ is the number of bases of M, $T(\mathrm{M}; 2, 1)$ the number of independent sets of M, and $T(\mathrm{M}; 1, 2)$ the number of spanning sets of M (see [4, 17]). With the arrangement of hyperplanes such that all the hyperplanes pass the origin, a linear matroid M over the reals is associated in a straightforward way. An arrangement is central if their hyperplanes have non-empty common intersection, and our arrangement is central. In this case, $T(\mathrm{M}; 2, 0)$ gives the number of regions of this central arrangement, and further interpretation in terms of arrangements for the coefficients of the characteristic polynomial is given [7] (see also [4, 17]).

3 Binary and Ternary Matroids

The computation of the Tutte polynomial is #P-hard in general, but it has so many applications not only in graph theory but also statistical physics, knot theory, code theory, etc. See [17]. The Tutte polynomial of a matroid can be computed via the BDD of its bases, and hence it is worth while devising an output-size sensitive algorithm to construct the BDD, which is practically available for matroids of moderate size. In order to adopt the top-down algorithm for the BDD mentioned in Lemma 1, we need an efficient procedure for testing isomorphism between minors under the identity map, which corresponds to the equivalence check of subfunctions of BDD representing all bases. This section is devoted to such procedures for binary and ternary matroids.

3.1 Binary matroid

A matroid is called binary if it can be represented as a linear matroid over GF(2). Given two matroids $\mathrm{M}(E)$ of m vectors $a_1, \ldots, a_m$ in $\mathrm{GF}(2)^n$ and $\mathrm{M}'(E)$ of m vectors $a'_1, \ldots, a'_m$ in $\mathrm{GF}(2)^n$, we will consider how to determine if $\mathrm{M}(E)$ and $\mathrm{M}'(E)$ are isomorphic under the identity mapping or not.

Let B be a base of $\mathrm{M}(E)$, and compute the coefficients β_{ij} that satisfy

$$a_j = \sum_{i \in B} \beta_{ij} a_i \qquad (j \in E - B).$$

We may suppose here that B is also a base of $\mathrm{M}'(E)$. Otherwise, $\mathrm{M}'(E)$ is not isomorphic to $\mathrm{M}(E)$. Then we obtain the coefficients β'_{ij} such that

$$a'_j = \sum_{i \in B} \beta'_{ij} a'_i \qquad (j \in E - B).$$

The following well-known theorem directly gives an efficient procedure for the isomorphism testing.

Theorem 1 *The binary matroids $\mathrm{M}(E)$ and $\mathrm{M}'(E)$ are isomorphic if and only if $\beta_{ij} = \beta'_{ij}$ holds for each $i \in B$ and $j \in E - B$.*

3.2 Ternary matroid

A matroid linearly representable over GF(3) is called a ternary matroid. Suppose we are given two matroids $\mathrm{M}(E)$ of m vectors $a_1, \ldots, a_m$ in $\mathrm{GF}(3)^n$ and $\mathrm{M}'(E)$ of m vectors $a'_1, \ldots, a'_m$ in $\mathrm{GF}(3)^n$. We will discuss how to detect the isomorphism between $\mathrm{M}(E)$ and $\mathrm{M}'(E)$.

Let B be a base of $\mathrm{M}(E)$, and define the coefficients β_{ij} and β'_{ij} similarly to the case of binary matroids. The following theorem is helpful for the isomorphism testing. See [11, §10.1] for the proof.

Theorem 2 *The ternary matroids $\mathrm{M}(E)$ and $\mathrm{M}'(E)$ are isomorphic if and only if there exists an appropriate mapping $\alpha : E \to \{1, -1\}$ such that $\alpha(i)\beta_{ij} = \alpha(j)\beta'_{ij}$ holds for each $i \in B$ and $j \in E - B$.*

We now consider how to perform the isomorphism testing based on Theorem 2. Suppose that $\{(i,j) \mid i \in B, j \in E-B, \beta_{ij} \neq 0\} = \{(i,j) \mid i \in B, j \in E - B, \beta'_{ij} \neq 0\}$. Because otherwise, $\mathrm{M}(E)$ and $\mathrm{M}(E)$ are not isomorphic. Construct a graph $H = (E, F)$ with vertex set E and edge set $F = F_+ \cup F_-$ defined by

$$F_+ = \{(i,j) \mid i \in B, j \in E - B, \beta_{ij} = \beta'_{ij} \neq 0\},$$
$$F_- = \{(i,j) \mid i \in B, j \in E - B, \beta_{ij} = -\beta'_{ij} \neq 0\}$$

Let $H^\circ = (E^\circ, F_-)$ be a graph obtained from H by contracting F_+. Then we have the following theorem, which gives an efficient procedure to detect the isomorphism. Recall that the bipartiteness of a graph can be checked in linear time.

Theorem 3 *The ternary matroids $\mathrm{M}(E)$ and $\mathrm{M}'(E)$ are isomorphic under the identity mapping if and only if the graph H° thus constructed is bipartite.*

Proof: Suppose $\mathrm{M}(E)$ and $\mathrm{M}'(E)$ are isomorphic. According to $\alpha : E \to \{1, -1\}$ of Theorem 2, we partition E into $E_+ = \{e \in E \mid \alpha(e) = 1\}$ and $E_- = \{e \in E \mid \alpha(e) = -1\}$. Then each edge of F_+ connects two vertices in the same side of E_+ or E_-, whereas the edges of F_- connect E_+ and E_-. Hence the contraction of F_+ from H yields a bipartite graph.

Conversely, suppose $H^\circ = (E^\circ, F_-)$ is a bipartite graph with a bipartition of E° into E°_+ and E°_-. We put $\alpha(e) = 1$ if $e \in E$ corresponds to a vertex in E°_+, and $\alpha(e) = -1$ otherwise. Then it is clear that $\alpha(i) = \alpha(j)$ if $(i,j) \in F_+$, whereas $\alpha(i) = -\alpha(j)$ for $(i,j) \in F_-$. Therefore $\alpha(i)\beta_{ij} = \alpha(j)\beta'_{ij}$ holds for every $(i,j) \in F$, which together with Theorem 2 implies that $\mathrm{M}(E)$ and $\mathrm{M}'(E)$ are isomorphic. □

4 Linear Matroid over the Reals and Arrangements

In the previous section, we have shown that for binary and ternary matroids, the BDD of all bases can be constructed in an output-size sensitive manner by the isomorphism test described there. But, this generally seems hard for linear matroids over fields except GF(2) and GF(3). For linear matroids over the reals, we can compute the Tutte polynomial directly by a different method based on its geometric structure.

Let $\mathrm{M} = \mathrm{M}(E)$ be a linear matroid of set E of vectors a_i $(i = 1, \ldots, m)$ in $\mathbf{R}^n$. Throughout this section, we regard n as a constant. Using the definition by the rank function directly, the Tutte polynomial can be computed by treating all the subsets, but this takes at least $\Omega(2^m)$ time. By using the original definition of the Tutte polynomial, we can compute it by enumerating all the bases, and computing the external and internal activities of each base. All the bases can be enumerated efficiently by the reverse search [2], and then the activities can be found in $O(m)$ time by regarding n as a constant. Summarizing this, we obtain the following.

Theorem 4 *The Tutte polynomial of the matroid* M *can be computed in* $O(mT(\mathrm{M};1,1))$ *time where* $T(\mathrm{M};1,1)$ *gives the number of bases of* M.

In this paper, we omit the proof of this theorem due to the space limitations, and instead we will concentrate on the use of the arrangement [6] to compute the Tutte polynomial.

Consider the arrangement of m hyperplanes $h_i = \{x \mid a_i \cdot x = 0\}$ $(i = 1, \ldots, m)$ in the dual $\mathbf{R}^n$. Each hyperplane h_i passes the origin, and this arrangement is central. We construct its face lattice by the incremental algorithm [6]. Note that since this is a central arrangement in the n-dimensional space, its combinatorial complexity is $O(m^{n-1})$, and not $\Theta(m^n)$.

A subset S of E is called a flat (or closed or a subspace) of this linear matroid $\mathrm{M}(E)$ if the addition of $e \in E - S$ to S increases the rank by one. These flats form a lattice (e.g., see [11, 16]). From the face lattice of the arrangement, the lattice of flats of $\mathrm{M}(E)$ can be constructed directly in $O(m^{n-1})$ time and space. We will show that from this lattice of flats the Tutte polynomial can be computed efficiently.

Now, fix an ordering of elements of E like $e_1, e_2, \ldots, e_m$. First, we discuss the data structure representing flats for this ordering. We represent the lattice of flats in a standard way of representing lattices. Each flat is represented by a sorted list (array) of its elements with respect to this ordering. Furthermore, for each subset S of E consisting of at most $n-1$ elements, we associate a flat σS that is minimal with respect to set inclusion among flats containing S, and for each S a pointer to σS in the lattice is provided. σ is the closure operator.

In our algorithm, we check all subsets of E consisting of n elements for bases of the matroid. Suppose we have a subset $B = \{e_{i(1)}, e_{i(2)}, \ldots, e_{i(n)}\}$ with $1 \leq i(1) < i(2) < \ldots < i(n) \leq m$ which is a base of this matroid. Define B_j to be $\{e_{i(1)}, \ldots, e_{i(j)}\}$ $(j = 1, \ldots, n)$.

Lemma 2 *(a) An element* $e_{i'} \in E - B$ *with* $i(j) < i' < i(j+1)$ *for some* j *in* $\{1, \ldots, n-1\}$ *is externally active with respect to the base* B *if and only if* $e_{i'} \in \sigma B_{i(j)}$.

(b) An element $e_{i'} \in E - B$ *with* $i(n) < i' \leq m$ *is externally active.*

Lemma 3 *An element* $e_{i(j)} \in B$ *is internally active if and only if all the elements* $e_{i'}$ *in* $E - B$ *with* $i' > i(j)$ *are in* $\sigma(B - \{e_{i(j)}\})$.

From these lemmas, we can compute the external and internal activities as follows. As for the external activity, for each $j = 1, \ldots, n-1$, we count the number s_j of elements $e_{i'}$ with $i(j) < i' < i(j+1)$ which are contained in $\sigma B_{i(j)}$. Then, $(m - i(n)) + \sum_{j=1}^{n-1} s_j$ is the external activity. Note that by considering intervals $(i(j), i(j+1))$, each externally active element is counted exactly once. As for the internal activity, for each $j = 1, \ldots, n$, we count the number r_j of elements $e_{i'}$ in $\sigma(B - \{e_{i(j)}\})$ with $i' > i(j)$. Then, $e_{i(j)}$ is internally active if $r_j = m - i(j)$.

Thus, both activities can be computed by counting the number of elements of flats within some interval like $(i(j), i(j+1))$. By representing elements of flats in a sorted array A1 by the ordering, this counting can be done in $O(\log n)$ time by binary search. If for each flat σS we have an array A2 of length m such that the i'-th entry of this array stores the number of elements $e_{j'}$ in σS with $j' \leq i'$, this counting can be done in a constant time with $O(m)$ space.

Lemma 4 *(a) If the array A1 is used for each flat in representing the lattice of flats, the external and internal activities of* B *can be computed in* $O(\log m)$ *time, with* $O(m^{n-1})$ *space in total.*

(b) If the array A2 is used for each flat in representing the lattice of flats, the external and internal activities of B *can be computed in a constant time, with* $O(m^n)$ *space in total.*

We can generate all n-element subsets of E

in $O(m^n)$ time. Finally, to compute the Tutte polynomial by the original definition by Tutte, we have to count the number of terms with the same external and internal activities. Noting that the summation of these numbers is bounded by the number of bases, and hence is $O(m^n)$, this can be done in $O(m^n)$ time by counting them in a batched way at the end. We thus obtain the following theorem.

Theorem 5 *The Tutte polynomial of a linear matroid* M *of* m *vectors in* $\mathbf{R}^n$ *can be computed in* $O(m^n \log m)$ *time and* $O(m^{n-1})$ *space or in* $O(m^n)$ *time and* $O(m^n)$ *space, when* n *is regarded as a constant.*

5 Graphic Arrangement

In the previous section, by virtue of geometric structures of arrangements, we show that the Tutte polynomial can be computed in time linear to the number of bases in the worst case. However, this is not the best algorithm at all in some cases. For example, when all a_i $(i = 1, \ldots, m)$ are generic, the matroid $\mathrm{M}(E)$ is a uniform matroid $U_{m,n}$ of m elements and rank n, and hence the Tutte polynomial is very easily computed. Testing whether all vectors a_i are generic has connection with a well-known problem of testing whether a given arrangement is nondegenerate in computational geometry. (For instance, given n lines in the plane, testing whether there are three lines meeting at a common point is hard to solve $o(n^2)$ time, and it is mostly considered that $\Omega(n^2)$ time would be necessary to solve this decision problem.) $O(m^{n-1})$ is the size of the arrangement and the size of the lattice of flats, and one may be tempted to consider that $\Omega(m^{n-1})$ is a lower bound to this computation problem.

However, by restricting the arrangement, we can obtain a better bound than $\Omega(m^{n-1})$ via BDD. In fact, we have demonstrated that there exists an output-size sensitive algorithm for constructing the BDD of binary and ternary matroids in section 3, and from this BDD the Tutte polynomial can be computed in time proportional to the size of the BDD. For linear matroids over the reals, there is not known any efficient algorithm, like for binary and ternary cases, for testing the isomorphism of two linear matroids under a given map, and this also implies that, for some restricted arrangements related to binary and ternary matroids, counting problems on them such as counting the number of cells may be solved in $o(m^{n-1})$ time.

We here show that for the arrangement associated with an undirected graph, called the graphic arrangement, this is the case, and this consideration relates the discussion so far with problems of partially ordered sets.

For an undirected $G = (V, E)$ with vertex set $V = \{v_1, \ldots, v_n\}$, consider a set of $m = |E|$ hyperplanes in $\mathbf{R}^n$ defined by $x_i = x_j$ for each edge $(v_i, v_j) \in E$. The arrangement of these hyperplanes is called the graphic arrangement of G. In Fig.1(d), the graphic arrangement of K_3 is depicted. Since it is essentially a two-dimensional arrangement, we show it by viewing the whole arrangement from $x_1 = x_2 = x_3 = \infty$ to the origin, or by cutting it with $x_1 + x_2 + x_3 = 0$. As is readily seen, each cell of this graphic arrangement corresponds to an acyclic orientation of G one-to-one. Concerning the number of cells of this arrangement, the following is known.

Lemma 5 *[4, 7] In the graphic arrangement, the number of cells is given by* $T(\mathrm{M}(G); 2, 0)$ *where* $\mathrm{M}(G)$ *is the graphic matroid of* G.

In [12], it is shown that the BDD for this graphic arrangement can be computed by using the graph structure, especially efficiently for graphs having a good vertex elimination ordering, from which we obtain the following.

Theorem 6 *[12] In the graphic arrangement of a simple planar graph with* n *vertices, the number of cells can be computed in* $O(2^{O(\sqrt{n})})$ *time.*

By extending the results in [7] for formulae to count the number of lower-dimensional faces in the arrangement, we can further obtain the following whose proof is omitted here.

Theorem 7 *The number of* $(n-k)$*-dimensional faces of the graphic arrangement of a simple planar graph with* n *vertices can be computed in* $O(2^{O(\sqrt{n})})$ *time for fixed* k.

Thus, as far as computing the combinatorial complexities of the arrangement is concerned, it can be done with much less time than the total size of the arrangement, when for example it is a graphic arrangement of a planar graph.

6 Computing the Number of Ideals of a Partial Order

Consider a partial order $\preceq$ on a finite set V. We denote this partially ordered set by $(V, \preceq)$. An ideal of this partially ordered set is a subset U of V such that, for any $v \in U$ and $u \preceq v$, $u \in U$. An empty set and the whole set V are ideals. The ideals play an important role in decomposing the partially ordered set.

For $(V, \preceq)$, we can define a polytope by $\{x \mid x = (x_v) \in \mathbb{R}^V,\ x_u \leq x_v \text{ for } u \preceq v,\ 0 \leq x_v \leq 1\}$. This polytope is called an order polytope. The vertices of the order polytope is a 0-1 vector. Each vertex corresponds to an ideal one-to-one, i.e., the complement of the characteristic vector of an ideal is a vertex.

Let $G = (V, E)$ be an acyclic graph corresponding to the partially ordered set $(V, \preceq)$ (G should be made to the Hasse diagram.) Consider the graphic arrangement for the unoriented graph for G. Then, the order polytope is the intersection of a cell of the graphic arrangement corresponding to the orientation of G and the unit hypercube $[0,1]^V$. See [1, 3, 17].

The number of ideals of $(V, \preceq)$ can be computed via BDD. To do this, we have to first construct the BDD representing all ideals, or all vertices of the order polytope. The Boolean function f representing all vertices of the order polytope can be described as follows ($(u, v) \in E$ implies $v \preceq u$): $f = \bigwedge_{(u,v)\in E}(x_u \vee \overline{x_v})$ This is not monotone, and a technique to construct the BDD of monotone functions [8] cannot be used. However, each clause of this formula consists of two literals, and this enables us to test the equivalence of subfunctions of this function.

For $U \subseteq V$, consider two subsets U_1 and U_2 of U such that there is not a pair of $u \in U_l$ and $v \in U - U_l$ with $u \preceq v$ ($l = 1, 2$). Let f_l be a subfunction obtained by setting $x_u = 1$ for $u \in U_l$ and $x_v = 0$ for $v \in U - U_l$ ($l = 1, 2$). Let V_l^1 be a subset of vertices in $V - U$ from which to a vertex in U_l there is a directed path in G ($l = 1, 2$). Let V_l^0 be a subset of vertices in $V - U$ to which from a vertex in $U - U_l$ there is a directed path in G ($l = 1, 2$). Define $V_l = V - (U \cup V_l^1 \cup V_l^0)$ ($l = 1, 2$). Then, we have the following whose proof is omitted.

Lemma 6 *The subfunctions f_1 and f_2 are equivalent if and only if $V_1^1 = V_2^1$ and $V_1^0 = V_2^0$ (and hence $V_1 = V_2$).*

To use this lemma to check the equivalence between two subfunctions, we have to check the whole V_1^0, V_1^1, V_2^0, V_2^1. However, some of vertices in these sets are contained in them by transitivity. In this regard, only "boundary vertices" around U determine these sets. Let us define this concept rigorously.

Consider an ordering of vertices in V into $v_1, v_2, \ldots, v_n$. The i-th elimination front $\widetilde{V}_i$ is a vertex subset consisting of vertices v_l with $l > i$ such that v_l is adjacent to some vertex v_j with $j \leq i$. Then, the following holds.

Lemma 7 *Let W be the i-th elimination front. If $V_1^h \cap W = V_2^h \cap W$, then $V_1^h = V_2^h$ ($h = 0, 1$).*

Hence, the equivalence check can be done by checking the equivalent of partitions of the elimination front W into three sets $V_l^0 \cap W$, $V_l^1 \cap W$ and the remaining elements. The number of distinct partitions of an N-element set into at most three sets is at most 3^N. Combining these with Lemma 1, we have the following.

Theorem 8 *When the underlying acyclic graph (Hasse diagram) G has an ordering of vertices such that the size of any elimination front is bounded by N, the BDD representing all ideals of this partially ordered set can be constructed in $O((n^2 \log n)3^N)$ time.*

A planar graph has a good vertex ordering [12, 15] by the planar separator theorem, and we have the following. For graphs having a good vertex ordering, similar results hold.

Theorem 9 *When the underlying acyclic graph (Hasse diagram) G is a simple planar graph with n vertices, the number of ideals of this partially ordered set can be computed in $O(2^{O(\sqrt{n})})$ time.*

7 Concluding Remarks

The results of this paper bridge many combinatorial structures and geometric ones algorithmically via the Tutte polynomial and BDD. This gives rise to the following open problems.

(1) As in the case for graphic matroids [12], analyze the size of BDD of all bases of binary and ternary matroids. Since this paper shows that for these two classes of matroids the BDD of all bases can be constructed in an output-size sensitive manner, the size of this BDD becomes a more important parameter.

(2) Show if the isomorphism test among other matroids can be done efficiently or not, say for transversal matroids, lower-truncated transversal matroids such as the union of graphic matroids [9]. This may be applied to the system reliability analysis of a generic matrix and com-

puting the network reliability maintaining high connectivity [13].

(3) Is there any further meaning of the Tutte polynomial and BDD for the arrangement in low dimensions? We here show that the number of lower dimensional faces can be also computed from the BDD. As mentioned in this paper, the BDD and Tutte polynomial can judge the non-degeneracy of the arrangement, and there would be more meanings connecting these.

(4) Is the BDD approach applicable to computing the number of linear extensions of a partially ordered set? This problem has attracted many researchers and randomized fully polynomial approximation schemes have been developed. See [1, 3, 5, 10, 17]. Our approach counts the exact value for moderate-size problems, and should be compared with such approximation schemes for practicality.

Acknowledgment

This research was supported in part by the Grant-in-Aid of the Ministry of Education, Japan.

References

[1] M. D. Atkinson and H. W. Chang: Computing the Number of Merging with Constraints. *Information Processing Letters*, Vol.24 (1987), pp.289–292.

[2] D. Avis and K. Fukuda: A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra. *Discrete and Computational Geometry*, Vol.8 (1992), pp.295–313.

[3] G. Brightwell and P. Winkler: Counting Linear Extensions is #P-Complete. *Proc. of the 23rd Annual ACM Symp. on Theory of Computing*, 1991, pp.175–181.

[4] T. Brylawski and J. Oxley: The Tutte Polynomial and Its Applications. In "Matroid Applications" (N. White, ed.), Encyclopedia of Mathematics and Its Applications, Vol.40 (1992), pp.123–225.

[5] M. Dyer, A. Frieze and R. Kannan: A Random Polynomial Time Algorithm for Approximating the Volume of Convex Bodies. *Journal of the Association for Computing Machinery*, Vol.38 (1991), pp.1–17.

[6] H. Edelsbrunner: *Algorithms in Combinatorial Geometry.* Springer-Verlag, Heidelberg, 1987.

[7] C. Greene and T. Zaslavsky: On the Interpretation of Whitney Numbers Through Arrangements of Hyperplanes, Zonotopes, non-Radon partitions and Orientations of Graphs. *Transactions of the American Mathematical Society*, Vol.280 (1983), pp.97–126.

[8] K. Hayase, K. Sadakane and S. Tani: Output-size Sensitiveness of OBDD Construction Through maximal Independent Set Problem. *Proc. of the Conference on Computing and Combinatorics (COCOON'95)*, Lecture Notes in Computer Science, Vol.959 (1995), pp.229–234.

[9] H. Imai: Network-Flow Algorithms for Lower-Truncated Transversal Polymatroids. *Journal of the Operations Research Society of Japan*, Vol.26, No.3 (1983), pp.186–210.

[10] L. Khachiyan: Complexity of Polytope Volume Computation. In "*New Trends in Discrete and Computational Geometry*" (J. Pach, ed.), Algorithms and Combinatorics, Vol.10, Springer-Verlag, 1993, pp.91–101.

[11] J. Oxley: *Matroid Theory.* Oxford University Press, Oxford, 1992.

[12] K. Sekine, H. Imai and S. Tani: Computing the Tutte Polynomial of a Graph of Moderate Size. *Proc. of the 5th International Symp. on Algorithms and Computation (ISAAC'95)*, Lecture Notes in Computer Science, Vol.1004 (1995), pp.224–233.

[13] K. Sekine and H. Imai: A Unified Approach via BDD to the Network Reliability and Path Numbers. *Tech. Rep. 95-09*, Dept. of Information Science, University of Tokyo, 1995.

[14] P. D. Seymour: Matroid representation over GF(3), *Journal of Combinatorial Theory*, Ser. B, Vol.26 (1979), pp.159–173.

[15] S. Tani and H. Imai: A Reordering Operation for an Ordered Binary Decision Diagram and an Extended Framework for Combinatorics of Graphs. *Proc. of the 5th International Symp. on Algorithms and Computation (ISAAC'94)*, Lecture Notes in Computer Science, Vol.834 (1994), pp.575–583.

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

[17] D. J. A. Welsh: *Complexity: Knots, Colourings and Counting.* London Mathematical Society Lecture Note Series, Vol.186, Cambridge University Press, 1993.