Combinatorics, Probability and Computing (19XX) 00, 000–000. © 19XX Cambridge University Press
DOI: 10.1017/S0000000000000000 Printed in the United Kingdom

The Tutte Polynomial for Matroids of Bounded Branch-Width†

Petr Hliněný ‡§

(not mailing address!)

School of Mathematical and Computing Sciences,

Victoria University of Wellington

P.O. Box 600, Wellington, New Zealand

It is a classical result of Jaeger, Vertigan and Welsh that evaluating the Tutte polynomial of a graph is $\#P$-hard in all but few special points. On the other hand, several papers in past years have shown that the Tutte polynomial of a graph can be efficiently computed for graphs of bounded tree-width. In this paper we present a recursive formula computing the Tutte polynomial of a matroid M represented over a finite field (which includes all graphic matroids), using a so called parse tree of a branch-decomposition of M. This formula provides an algorithm computing the Tutte polynomial for a representable matroid of bounded branch-width in polynomial time with a fixed exponent.
Keywords: matroid, Tutte polynomial, branch-width, parse tree.
2000 Math subject classification: primary 05B35, secondary 68R05.

1. Introduction

We refer to the next section for necessary formal definitions. The Tutte polynomial $T(M;\,x,y)$ of a matroid naturally extends the original Tutte's definition of this polynomial for graphs [11]. If M is the cycle matroid of a graph G, then $T(M;\,x,y) = T(G;\,x,y)$. Most of the complexity related research has been so far focused on the graphic version of the polynomial.

Specific evaluations of the Tutte polynomial include such interesting and difficult quantities as the chromatic and flow polynomials of a graph, the critical number of a matroid, the weight enumerator of a linear code, the all-terminal reliability of a network, the partition function of the Ising model in physics, or the Jones and Kaufman bracket polynomials of an alternating knot. See, for example, [13, 14]. It is not surprising that computing or evaluating the Tutte polynomial is $\#P$-hard except for a few special cases, which was proved by Jaeger, Vertigan and Welsh [7]. It is even $\#P$-hard to evaluate the Tutte polynomial at all but few special points for planar graphs [12].

† The research has been supported by a New Zealand Marsden Fund research grant to Geoff Whittle.
‡ Current affiliation and *mailing address*: Department of Computer Science, FEI VŠB – Technical University of Ostrava, 17. listopadu 15, 70833 Ostrava, Czech Republic.
§ E-mail: `petr.hlineny@vsb.cz`

In view of that hardness result, researchers have focused on computing the Tutte polynomial for special graph classes. Most noticeable to us are algorithms computing the Tutte polynomial for graphs of bounded tree-width: Andrzejak [1] has given a polynomial-time algorithm for computation of the Tutte polynomial for such graphs, but unfortunately, with an exponent heavily depending on tree-width. Noble [9] has independently developed a low-exponent polynomial-time algorithm for evaluation of the Tutte polynomial at a rational point. (We have noticed that some papers refer to Noble's result as to an evaluation of the Tutte polynomial in "linear time", but this is not quite true – the polynomial is evaluated using a linear number of arithmetic operations, and the cost of one operation depends significantly on the size of the graph, as it is precisely stated in [9, Theorem 1.1].) Another, "logic-type" approach to the same problem, using generating functions definable in the monadic second order logic, has been presented by Makowsky in [8]. However, that approach provides no reasonably explicit algorithm.

Lastly, we want to mention the concept of *parametrized complexity* [3] that has received attention over the past years. In view of this concept, evaluating the Tutte polynomial at any rational point is a uniformly fixed parameter tractable problem for graphs of bounded tree-width. (In general, many hard graph problems become easy on bounded tree-width graphs, and an analogous phenomenon occurs in represented matroids of bounded branch-width [4].)

In Section 4 of this paper we present a recursive formula that straightforwardly computes the Tutte polynomial of a matroid of bounded branch-width represented over a finite field using its "parse tree". Parse trees for those matroids have been introduced by the author in [4]. The formula, together with an algorithm for finding a bounded branch-decomposition [5], provide an algorithm computing the Tutte polynomial of such a matroid in time $O(n^6 \log n \log\log n)$.

The concept of branch-width is very similar to that of tree-width, but branch-width has an immediate generalization to matroids. Moreover, it seems that branch-width is a slightly better concept than tree-width for applications. (For example, the above mentioned Noble's paper [9] uses a tree-decomposition of a graph to derive an information which follows more directly from a branch-decomposition.) If a graph G has bounded tree-width, then its cycle matroid $M(G)$ has bounded branch-width; and a bounded-width branch-decomposition of $M(G)$ can be quickly computed from a bounded-width tree-decomposition of G. So our result generalizes the results of [1, 9].

2. Basics of Matroids

We refer to Oxley [10] for matroid terminology. A *matroid* is a pair $M = (E, \mathcal{B})$ where $E = E(M)$ is the ground set of M (elements of M), and $\mathcal{B} \subseteq 2^E$ is a nonempty collection of *bases* of M. Moreover, matroid bases satisfy the "exchange axiom"; if $B_1, B_2 \in \mathcal{B}$ and $x \in B_1 - B_2$, then there is $y \in B_2 - B_1$ such that $(B_1 - \{x\}) \cup \{y\} \in \mathcal{B}$. Subsets of bases are called *independent sets*, and the remaining sets are *dependent*. Minimal dependent sets are called *circuits*. An element e of M is called a *loop* if $\{e\}$ is dependent in M. All bases have the same cardinality called the *rank* $\mathrm{r}(M)$ of the matroid. The *rank* (function) $\mathrm{r}_M(X)$ in M is the maximal cardinality of an independent subset of a set $X \subseteq E(M)$.

If G is a graph, then its *cycle matroid* on the ground set $E(G)$ is denoted by $M(G)$. The independent sets of $M(G)$ are acyclic subsets (forests) in G, and the circuits of $M(G)$ are the cycles in G. Another example of a matroid is a finite set of vectors with usual linear dependency. If $\boldsymbol{A}$ is a matrix, then the matroid formed by the column vectors of $\boldsymbol{A}$ is called the *vector matroid* of $\boldsymbol{A}$, and denoted by $M(\boldsymbol{A})$. The matrix $\boldsymbol{A}$ is a *representation* of a matroid $M \simeq M(\boldsymbol{A})$. In particular, cycle matroids of graphs have representations over any field. (Fig. 1.)

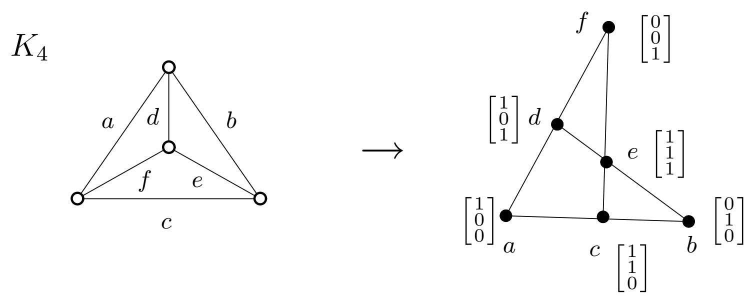

Figure 1 An example of a vector representation of the cycle matroid $M(K_4)$. The matroid elements are depicted by dots, and their (linear) dependency is shown using lines.

The *Tutte polynomial* of a matroid M on the ground set E is given by

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

Notice that $T(M;\, x,y) = T(M^*;\, y,x)$ for the dual matroid M^*. The Tutte polynomial of a graph G is the polynomial $T(G;\, x,y) = T(M(G);\, x,y)$. The following statement is proved in [7]:

Theorem 2.1. (Jaeger, Vertigan, and Welsh, 1990) *Let G be a graph. Evaluating the Tutte polynomial $T(G;\, x,y)$ at $(x,y) = (a,b)$ is #P-hard unless $(a-1)(b-1) = 1$ or $(a,b) \in \big\{(1,1),(-1,-1),(0,-1),\ (-1,0),(i,-i),(-i,i),\ (j,j^2),(j^2,j)\big\}$, where $i^2 = -1$ and $j = e^{2\pi i/3}$.*

Another important concept is matroid connectivity, which is close, but somehow different, to traditional graph connectivity. The *connectivity function* λ_M of a matroid M is defined for all subsets $A \subseteq E = E(M)$ by

$$\lambda_M(A) = \mathrm{r}_M(A) + \mathrm{r}_M(E-A) - \mathrm{r}(M) + 1\,.$$

Notice that the function is symmetric $\lambda_M(A) = \lambda_M(E-A)$. A partition $(A, E-A)$ is called a *k-separation* if $\lambda_M(A) \le k$ and both $|A|, |E-A| \ge k$. Geometrically, the spans of the two sides of a k-separation intersect in a subspace of rank less than k. In a corresponding graph view, the connectivity function $\lambda_G(F)$ of an edge subset $F \subseteq E(G)$ equals the number of vertices of G incident both with F and with $E(G)-F$. (Then $\lambda_G(F) = \lambda_{M(G)}(F)$ provided both sides of the separation are connected in G.)

A sub-cubic tree is a tree in which all vertices have degree at most three. Let $\ell(T)$

denote the set of leaves of a tree T. Let M be a matroid on the ground set $E = E(M)$. A *branch-decomposition* of M is a pair (T, τ) where T is a sub-cubic tree, and τ is an injection of E into $\ell(T)$. Let e be an edge of T, and T_1 be one of the connected components of $T - e$. The *width* of e in T equals $\lambda_M(A) = \lambda_M(E - A)$, where $A = \tau^{-1}(\ell(T_1))$ are the elements mapped to the component T_1. Width of the branch-decomposition (T, τ) is maximum of the widths of all edges of T, and *branch-width* of M is the minimal width over all branch-decompositions of M. See examples in Fig. 2.

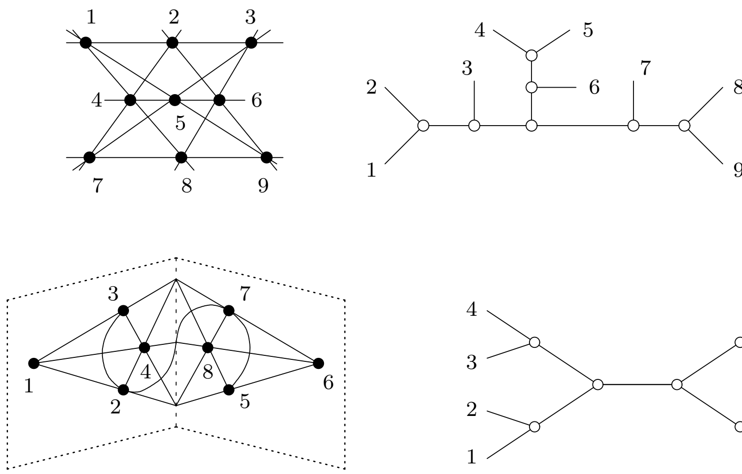

Figure 2 Two examples of width-3 branch decompositions of the Pappus matroid (top left, in rank 3) and of the binary affine cube (bottom left, in rank 4). The lines in matroid pictures show dependencies among elements.

In the context of our research, we have to be particularly precise about the way we handle matroid representations over a field $I\!F$:

We denote by $PG(r, I\!F)$ the *projective geometry (space)* obtained from a vector space $I\!F^{r+1}$. See [10, Section 6.1,6.3] for an overview of projective spaces and of equivalence of matroid representations. As one can easily see, a loopless matroid representation could be viewed as a multiset of points in the projective space $PG(r, I\!F)$ where r is the rank of M. For a set $A \subseteq PG(r, I\!F)$, we denote by $\langle A \rangle$ the span of A in the space (the affine closure in terms of the underlying vector space). The projective rank $\mathrm{r}(A)$ of A is the maximal cardinality of an affinely independent subset of A. A projective transformation is a mapping between two projective spaces over $I\!F$ that is induced by a linear transformation between the underlying vector spaces.

We call a finite multiset of points in a projective space over $I\!F$ a *point configuration.* We say that two point configurations P_1, P_2 spanning projective spaces over $I\!F$ are *equivalent* if there is a non-singular projective transformation between the projective spaces that maps P_1 onto P_2 bijectively. We represent a loop in a point configuration by the empty subspace $\emptyset$. In the above equivalence, loops are mapped only to loops. An *$I\!F$-represented matroid* is defined as an equivalence class of finite point configurations over $I\!F$.

Obviously, all point configurations in one equivalence class represent the same isomorphism class of matroids. (The converse is not always true.) When we want to deal with an $\mathbb{F}$-represented matroid, we actually pick an arbitrary point configuration from the equivalence class. Standard matroidal terms are inherited from matroids to represented matroids. Since we do not label points in our configurations, we are dealing with unlabeled matroid elements, which is in correspondence with the definition of the Tutte polynomial.

3. Parse Trees for Bounded Branch-Width

In this section we introduce our basic formal tool — parse trees for represented matroids (of bounded branch-width). We repeat the definitions and basic results from [4, Section 3]. All matroids throughout the next sections are $\mathbb{F}$-represented for some fixed finite field $\mathbb{F}$. Hence, for simplicity, if we say a "(represented) matroid", then we mean an $\mathbb{F}$-represented matroid. If we speak about a projective space, we mean the projective geometry over the field $\mathbb{F}$. Let $[s,t]$ denote the set $\{s, s+1, \ldots, t\}$.

The following definition presents a possible way of formalizing the notion of a "matroid with a boundary of size (rank) t". (Since matroids have no vertices unlike graphs, we have to introduce special elements that define the span of a matroid boundary.)

Definition. A pair $\bar{N} = (N, \delta)$ is called a *t-boundaried (represented) matroid* if the following holds: $t \geq 0$ is an integer, N is a represented matroid, and $\delta : [1,t] \to E(N)$ is an injective mapping such that the image $\delta([1,t])$ is independent in N.

We call $J(\bar{N}) = E(N) - \delta([1,t])$ the *internal elements* of $\bar{N}$, elements of $\delta([1,t])$ the *boundary points* of $\bar{N}$, and t the *boundary rank* of $\bar{N}$. We denote by $\partial(\bar{N})$ the boundary subspace spanned by $\delta([1,t])$. To understand the definition properly, one should imagine a represented matroid on the internal elements $J(\bar{N})$, equipped with the boundary space $\partial(\bar{N})$ of rank t which is represented by t additional independent boundary points of $\bar{N}$. The basic operation we shall use is the *boundary sum* described by the next definition.

Definition. Let $\bar{N}_1 = (N_1, \delta_1)$, $\bar{N}_2 = (N_2, \delta_2)$ be two t-boundaried represented matroids. We denote by $\bar{N}_1 \bar{\oplus} \bar{N}_2 = N$ the represented matroid defined as follows: Let Ψ_1, Ψ_2 be projective spaces such that the intersection $\Psi_1 \cap \Psi_2$ has rank exactly t. Suppose that, for $i = 1, 2$, $P_i \subset \Psi_i$ is a point configuration representing N_i such that $P_1 \cap P_2 = \delta_1([1,t]) = \delta_2([1,t]) \subset \Psi_1 \cap \Psi_2$, and $\delta_2(j) = \delta_1(j)$ for $j \in [1,t]$. Then N is the matroid represented by $(P_1 \cup P_2) - \delta_1([1,t])$.

Informally, the boundary sum $\bar{N}_1 \bar{\oplus} \bar{N}_2 = N$ on the ground set $E(N) = J(\bar{N}_1) \dot{\cup} J(\bar{N}_2)$ is obtained by gluing the representations of N_1 and N_2 on a common subspace of rank t, so that the boundary points are identified in order and then deleted. See in Fig. 3. It is a matter of elementary linear algebra to verify that a boundary sum is well defined with respect to equivalence of point configurations.

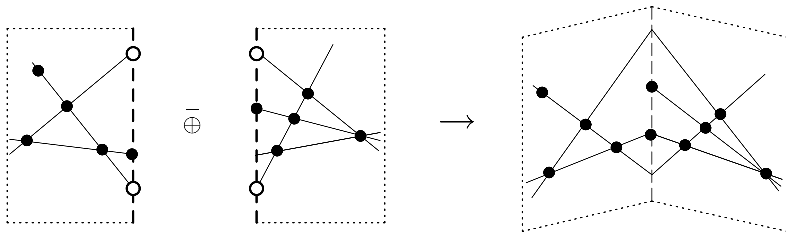

Figure 3 An example of a boundary sum of two 2-boundaried matroids. The internal elements are drawn as solid dots, the boundary points as hollow dots, and the boundary subspaces of rank 2 are drawn with thick dashed lines. The resulting sum is a matroid represented on two intersecting planes in rank 4 (like the "3-dimensional" picture on the right).

Definition. A *$\leq t$-boundaried composition* operator is defined as a quadruple $\odot = (R, \gamma_1, \gamma_2, \gamma_3)$, where R is a represented matroid, $\gamma_i : [1, t_i] \to E(R)$ is an injective mapping for $i = 1, 2, 3$ and some fixed $0 \leq t_i \leq t$, each $\gamma_i([1, t_i])$ is an independent set in R, and $\big(\gamma_i([1, t_i]) : i = 1, 2, 3\big)$ is a partition of $E(R)$.
The $\leq t$-boundaried composition operator $\odot$ is a binary operator applied to a t_1-boundaried represented matroid $\bar{N}_1 = (N_1, \delta_1)$ and to a t_2-boundaried represented matroid $\bar{N}_2 = (N_2, \delta_2)$. The result of the composition is a t_3-boundaried represented matroid $\bar{N} = (N, \gamma_3)$, written as $\bar{N} = \bar{N}_1 \odot \bar{N}_2$, where a matroid N is defined using boundaried sums: $N' = \bar{N}_1 \bar{\oplus} (R, \gamma_1)$, $N = (N', \gamma_2) \bar{\oplus} \bar{N}_2$.

Speaking informally, a boundaried composition operator is a bounded-rank configuration with three boundaries distinguished by $\gamma_1, \gamma_2, \gamma_3$, and with no other internal points. The meaning of a composition $\bar{N} = \bar{N}_1 \odot \bar{N}_2$ is that, for $i = 1, 2$, we glue the represented matroid N_i with R, matching $\delta_i([1, t_i])$ with $\gamma_i([1, t_i])$ in order. The result is a t_3-boundaried matroid $\bar{N}$ with boundary $\gamma_3([1, t_3])$. Notice that, in general, there are more than one boundaried composition operators with the same ranks. For reference we denote by $t_i(\odot) = t_i$, by $R(\odot) = R$, and by $\gamma_i(\odot) = \gamma_i$.

Let $\bar{\Omega}_t$ denote the *empty* t-boundaried matroid (Ω, δ_0) where $t \geq 0$ and $\delta_0([1, t]) = E(\Omega)$ (t will often be implicit in the context). If $\bar{N} = (N, \delta)$ is an arbitrary t-boundaried matroid, then $\bar{N} \bar{\oplus} \bar{\Omega}_t$ is actually the restriction of $\bar{N}$ to $E(N) - \delta([1, t])$. Let $\bar{\Upsilon}$ denote the *single-element* 1-boundaried matroid (Υ, δ_1) where $E(\Upsilon) = \{x, x'\}$ are two parallel elements, and $\delta_1(1) = x'$. Let $\bar{\Upsilon}_0$ denote the *loop* 0-boundaried matroid (Υ_0, δ_0) where $E(\Upsilon_0) = \{z\}$ is a loop, and $\delta_0 : \emptyset \to \emptyset$. Let $\mathcal{R}_t^{I\!F}$ denote the set of all $\leq t$-boundaried composition operators over the field $I\!F$.

We set $\Pi_t = \mathcal{R}_t^{I\!F} \cup \{\bar{\Upsilon}, \bar{\Upsilon}_0\}$. Let Π_t^{**} denote the set of all rooted sub-binary trees with vertices labeled by elements of Π_t. Considering a vertex v of a tree $T \in \Pi_t^{**}$; we set $\varrho(v) = 1$ if v is labeled by $\bar{\Upsilon}$, $\varrho(v) = 0$ if v is labeled by $\bar{\Upsilon}_0$, and $\varrho(v) = t_3(\odot)$ if v is labeled by $\odot$. We call T a *$\leq t$-boundaried parse tree* if the following are true:

- Only leaves of T are labeled by $\bar{\Upsilon}$ or $\bar{\Upsilon}_0$.
- If a vertex v of T labeled by a composition operator $\odot$ has no left (no right) son, then $t_1(\odot) = 0$ ($t_2(\odot) = 0$).

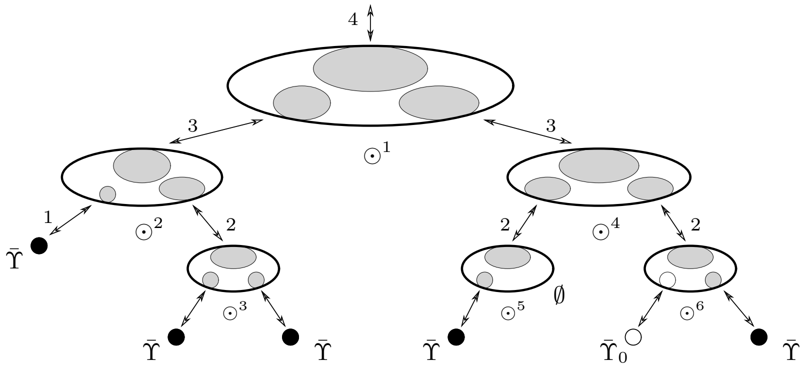

Figure 4 An example of a boundaried parse tree. The ovals represent composition operators, with shaded parts for the boundaries and edge-numbers for the boundary ranks. (E.g. $\odot^4 = (R^4, \gamma_1^4, \gamma_2^4, \gamma_3^4)$ where $\gamma_1^4, \gamma_2^4 : [1,2] \to E(R^4)$, $\gamma_3^4 : [1,3] \to E(R^4)$.)

- If a vertex v of T labeled by $\odot$ has left son u_1 (right son u_2), then $t_1(\odot) = \varrho(u_1)$ ($t_2(\odot) = \varrho(u_2)$).

Informally, the boundary ranks of composition operators and/or single-element terminals must agree across each edge. Notice that $\bar\Upsilon$ or $\bar\Upsilon_0$ are the only labels from Π_t that "create" elements of the resulting represented matroid $P(T)$ in the next definition. See an illustration example in Fig. 4.

Definition. Let T be a $\leq t$-boundaried parse tree. The $\leq t$-boundaried represented *matroid $\bar P(T)$ parsed by T* is recursively defined as follows:

- If T is an empty tree, then $\bar P(T) = \bar\Omega_0$.
- If T has one vertex labeled by $\bar\Upsilon$ (by $\bar\Upsilon_0$), then $\bar P(T) = \bar\Upsilon$ ($= \bar\Upsilon_0$).
- If the root r of T is labeled $\odot_r$, and r has the left subtree T_1 and the right subtree T_2 (possibly empty trees), then $\bar P(T) = \bar P(T_1) \odot_r \bar P(T_2)$.

The composition is well defined according to the parse-tree description in the previous paragraph. The represented matroid parsed by T is $P(T) = \bar P(T) \,\bar\oplus\, \bar\Omega$, i.e. $\bar P(T)$ restricted to its internal elements.

Proposition 3.1. (PH) *The set $\mathcal{R}_t^{\mathbb{F}}$ is finite if $\mathbb{F}$ is a finite field.*

We need one additional technical condition on a parse tree for computing of the Tutte polynomial in Theorem 4.1. We say that a t-boundaried represented matroid $\bar M = (M, \delta)$ is *spanning* if the boundary subspace $\partial(\bar M)$ is contained in the span $\langle J(\bar M)\rangle$ of the internal points of $\bar M$. We say that a $\leq t$-boundaried parse tree T is *spanning* if, for each nonempty subtree T_1 of T, the boundaried matroid $\bar P(T_1)$ is spanning and nonempty. The following two theorems are proved in [4] and in [5], respectively. (Their validity is not dependent on whether we require the additional condition of being spanning or not.)

Theorem 3.1. (PH, [4]) *A represented matroid M has branch-width at most $t+1$ if and only if it is parsed by some $\leq t$-boundaried spanning parse tree.*

Theorem 3.2. (PH, [5]) *Let $\mathbb{F}$ be a finite field, and let t be an integer constant. Suppose that N is an n-element $\mathbb{F}$-representable matroid given by a matrix over $\mathbb{F}$. If N has branch-width at most t, then one can find a $\leq 3t$-boundaried spanning parse tree T such that $P(T) \simeq N$ in time $O(n^3)$.*

4. Boundaried Tutte Polynomial

After having defined the necessary formal tools, we are going to apply matroid parse trees to a recursive computation of the Tutte polynomial. So, first of all, we need a formal way to handle the definition of the Tutte polynomial over boundaried represented matroids; hence to formally distinguish, or to "mark", a subspace of the matroid boundary. Let $\bar{\Omega}_t$ be the empty t-boundaried matroid as above, and let $\Phi \subseteq \partial(\bar{\Omega}_t)$ be a subspace of the boundary space of $\bar{\Omega}_t$. Then the pair $\mathrm{K} = [\bar{\Omega}_t, \Phi]$ is called a *t-boundary mark*. (We extend projective transformations of point configurations to subspaces in the obvious sense.) The set of all t-boundary marks is denoted by $\mathcal{K}_t$, and the cardinality $|\mathcal{K}_t|$ clearly equals the number of subspaces of the rank-t projective geometry over $\mathbb{F}$.

Let $\bar{M} = (M, \delta)$ be a t-boundaried represented matroid, and let $J = J(\bar{M})$ be the internal elements of $\bar{M}$. We denote by $Z_t = (z_{\mathrm{K}} : \mathrm{K} \in \mathcal{K}_t)$ a vector of $|\mathcal{K}_t|$ free variables. Moreover, for $A \subseteq J$, we denote by $\mathrm{K}\left(\bar{M}\,|A\right) \in \mathcal{K}_t$ the t-boundary mark formed by the boundary points of $\bar{M}$, and by the subspace $\partial(\bar{M}) \cap \langle A \rangle$ which is the intersection of the boundary of $\bar{M}$ with the span of A. The *boundaried Tutte polynomial* of a t-boundaried matroid $\bar{M}$ is given by

$$T_B\left(\bar{M};\, x, y, Z_t\right) = \sum_{A \subseteq J} (x-1)^{\mathrm{r}_M(J) - \mathrm{r}_M(A)} \cdot (y-1)^{|A| - \mathrm{r}_M(A)} \cdot z_{\mathrm{K}\left(\bar{M}\,|A\right)}\,.$$

Proposition 4.1. $T\big(\bar{M} \,\bar{\oplus}\, \bar{\Omega};\, x, y\big) = T_B\left(\bar{M};\, x, y, (1, \ldots, 1)\right)$.

Proof. Since $\bar{M} \,\bar{\oplus}\, \bar{\Omega} = M_1$ is a restriction of M to J, we have $J = E(M_1)$. Moreover, the rank function $\mathrm{r}_{M_1} : 2^J \to \mathbb{N}$ is a restriction of r_M to the subsets of J. Hence the claim follows directly from the above definition. $\square$

We shall combine the machinery of boundaried parse trees with the previous definition in order to produce a straightforward recursive formula for computing the boundaried Tutte polynomial over a given matroid parse tree. If M is a matroid (represented over $\mathbb{F}$) of branch-width at most $t+1$, then there is a $\leq t$-boundaried parse tree T_M by Theorem 3.1, such that the represented matroid $P(T_M)$ parsed by T_M is isomorphic to M. Thus we may use Proposition 4.1 to compute the Tutte polynomial for M from the boundaried one for $\bar{P}(T_M)$.

There is an obvious natural way to apply a composition operator to boundary marks. For that we need the following formalization: For a t-boundary mark $\mathrm{K} = [\bar{\Omega}_t, \Phi]$ where $\bar{\Omega}_t = (\Omega, \delta_0)$, we say that a t-boundaried matroid $\bar{L} = (L, \delta_0)$ is a *representative* for K if $E(L) \supseteq E(\Omega)$, and the span $\langle E(L) - E(\Omega) \rangle = \Phi$.

Let $\odot = (R, \gamma_1, \gamma_2, \gamma_3)$ be a $\leq t$-boundaried composition operator where $t_i = t_i(\odot)$, and let $\mathrm{K}_i \in \mathcal{K}_{t_i}$ for $i = 1, 2$ be a t_i-boundary mark with a representative $\bar{L}_i$. Then we denote by $[\mathrm{K}_1 \odot \mathrm{K}_2]$ the t_3-boundary mark $\mathrm{K}\left(\bar{L}\,|J\right)$ in the matroid $\bar{L} = \bar{L}_1 \odot \bar{L}_2$ shown by the set $J = J(\bar{L}) = J(\bar{L}_1) \cup J(\bar{L}_2)$. Moreover, we denote by $\sigma(\odot) = t_1 + t_2 - \mathrm{r}_R\left(\gamma_1([1, t_1]) \cup \gamma_2([1, t_2])\right)$ the projective rank of the intersection of boundaries $\partial(R, \gamma_1) \cap \partial(R, \gamma_2)$ in $\odot$, and by $\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2) = \mathrm{r}_L(J(\bar{L}_1)) + \mathrm{r}_L(J(\bar{L}_2)) - \mathrm{r}_L(J(\bar{L}))$ the rank of the intersection of the subspaces marked by K_1 and K_2 in $\odot$. It is easy to see that the outcomes of these definitions are independent of a particular choice of the representatives $\bar{L}_1, \bar{L}_2$.

Our recursive computation of the boundaried Tutte polynomial over a boundaried spanning parse tree is based on the following statement. Here $P(x, y, z, \ldots) \restriction z$ stands for the coefficient of the variable z in a polynomial P.

Theorem 4.1. *Let $\odot$ be a $\leq t$-boundaried composition operator, and let $t_i = t_i(\odot)$ for $i = 1, 2, 3$. Suppose that $\bar{N}_i$, $i = 1, 2$ are t_i-boundaried spanning represented matroids. Then*

$$T_B\left(\bar{N}_1 \odot \bar{N}_2;\, x, y, Z_{t_3}\right) =$$
$$= \sum_{\mathrm{K}_1 \in \mathcal{K}_{t_1},\ \mathrm{K}_2 \in \mathcal{K}_{t_2}} \left(T_B\left(\bar{N}_1;\, x, y, Z_{t_1}\right) \restriction z_{\mathrm{K}_1}\right) \cdot \left(T_B\left(\bar{N}_2;\, x, y, Z_{t_2}\right) \restriction z_{\mathrm{K}_2}\right) \cdot$$
$$\cdot (x-1)^{\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2) - \sigma(\odot)} \cdot (y-1)^{\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2)} \cdot z_{[\mathrm{K}_1 \odot \mathrm{K}_2]}\ .$$

Proof. Let $\odot = (R, \gamma_1, \gamma_2, \gamma_3)$, let $\bar{N}_i = (N_i, \delta_i)$ for $i = 1, 2$, and let $\bar{N} = (N, \delta) = \bar{N}_1 \odot \bar{N}_2$. We choose an arbitrary set $A \subseteq J = J(\bar{N})$, and we denote by $J_1 = J(\bar{N}_1)$, $J_2 = J(\bar{N}_2)$, and by $A_1 = A \cap J_1$, $A_2 = A \cap J_2$. (Notice that J_1, J_2 form a partition of J.) We denote by $\mathrm{K}_i = \mathrm{K}\left(\bar{N}_i\,|A_i\right)$ for $i = 1, 2$ the boundary mark shown by the set A_i in $\bar{N}_i$. Let $\mathrm{r}(\cdot)$ be the rank function of the projective space, which is modular.

For the point configuration representing N we may write $\mathrm{r}_N(J) = \mathrm{r}_N(J_1) + \mathrm{r}_N(J_2) - \mathrm{r}(\langle J_1\rangle \cap \langle J_2\rangle)$ using modularity. Since both N_1, N_2 are spanning, and $\langle J_1\rangle \cap \langle J_2\rangle \subseteq \langle E(R)\rangle$ in the composition $\odot$, we get $\mathrm{r}(\langle J_1\rangle \cap \langle J_2\rangle) = r\big(\partial(R, \gamma_1) \cap \partial(R, \gamma_2)\big) = \sigma(\odot)$. Hence $\mathrm{r}_N(J) = \mathrm{r}_N(J_1) + \mathrm{r}_N(J_2) - \sigma(\odot)$.

Considering similarly the points of A in N, we may write $\mathrm{r}_N(A) = \mathrm{r}_N(A_1) + \mathrm{r}_N(A_2) - \mathrm{r}(\langle A_1\rangle \cap \langle A_2\rangle)$. Let us choose a representative $\bar{L}_i$ of the boundary mark K_i, $i = 1, 2$, and denote by $\bar{L} = \bar{L}_1 \odot \bar{L}_2$. Then (considering first the point configuration of $\bar{N}$ and then the point configuration of $\bar{L}$) we have $\mathrm{r}(\langle A_1\rangle \cap \langle A_2\rangle) = r\big(\langle J(L_1)\rangle \cap \langle J(L_2)\rangle\big) = \mathrm{r}_L(J(L_1)) + \mathrm{r}_L(J(L_2)) - \mathrm{r}_L(J(L))$ by modularity of $\mathrm{r}(\cdot)$. Hence $\mathrm{r}_N(A) = \mathrm{r}_N(A_1) + \mathrm{r}_N(A_2) - \varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2)$.

Altogether, we conclude

$$(x-1)^{\mathrm{r}_N(J) - \mathrm{r}_N(A)} \cdot (y-1)^{|A| - \mathrm{r}_N(A)} =$$
$$= (x-1)^{\mathrm{r}_{N_1}(J_1) - \mathrm{r}_{N_1}(A_1) + \mathrm{r}_{N_2}(J_2) - \mathrm{r}_{N_2}(A_2)} \cdot (y-1)^{|A_1| - \mathrm{r}_{N_1}(A_1) + |A_2| - \mathrm{r}_{N_2}(A_2)} \cdot$$
$$\cdot (x-1)^{\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2) - \sigma(\odot)} \cdot (y-1)^{\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2)}\ .$$

Moreover, $[\mathrm{K}_1 \odot \mathrm{K}_2]$ is the boundary mark of the set A in $\bar{N}$. After multiplying the last expression by $z_{[\mathrm{K}_1\odot\mathrm{K}_2]}$, summing over all choices of $A \subseteq J$, and grouping the terms by $\mathrm{K}_1, \mathrm{K}_2$, we get the required formula. $\square$

In the next corollaries we provide a "computer science" view of the previous result. To distinguish the Z-variables of distinct Tutte polynomials, we introduce additional vectors of free variables Z'_{t_1} and Z''_{t_2}. Then, we may rewrite the formula of the previous theorem in the following way:

Corollary 4.1. *In the setting of Theorem 4.1,*

$$T_B\left(\bar{N}_1 \odot \bar{N}_2;\, x, y, Z_{t_3}\right) = T_B\left(\bar{N}_1;\, x, y, Z'_{t_1}\right) \cdot T_B\left(\bar{N}_2;\, x, y, Z''_{t_2}\right),$$

where

$$z'_{\mathrm{K}_1} \cdot z''_{\mathrm{K}_2} = (x-1)^{\varrho(\odot;\,\mathrm{K}_1,\mathrm{K}_2)-\sigma(\odot)} \cdot (y-1)^{\varrho(\odot;\,\mathrm{K}_1,\mathrm{K}_2)} \cdot z_{[\mathrm{K}_1\odot\mathrm{K}_2]}$$

for each pair $\mathrm{K}_i \in \mathcal{K}_{t_i}$, $i = 1, 2$.

Proof. The key fact to notice here is that each term of the polynomial $T_B\left(\bar{N}_1;\, x, y, Z'_{t_1}\right)$ contains exactly one of the variables z'_{K_1} for $\mathrm{K}_1 \in \mathcal{K}_{t_1}$, and z'_{K_1} is in the first power. An analogous statement applies to $T_B\left(\bar{N}_2;\, x, y, Z''_{t_2}\right)$. Hence the above substitutions of $z'_{\mathrm{K}_1} \cdot z''_{\mathrm{K}_2}$ provide the same result as the formula in Theorem 4.1. $\square$

Corollary 4.2. *Let $T \in \Pi^{**}_t$ be a $\leq t$-boundaried spanning parse tree. If T is an empty tree, then*

$$T_B\left(\bar{P}(T);\, x, y, Z_0\right) = T_B\left(\bar{\Omega}_0;\, x, y, Z_0\right) = z_{\mathrm{K}\left(\bar{\Omega}_0\,|\emptyset\right)}\,.$$

If T has exactly one vertex labeled by $\bar{\Upsilon}$ or $\bar{\Upsilon}_0$, then

$$T_B\left(\bar{P}(T);\, x, y, Z_1\right) = T_B\left(\bar{\Upsilon};\, x, y, Z_1\right) = (x-1) z_{\mathrm{K}\left(\bar{\Upsilon}\,|\emptyset\right)} + z_{\mathrm{K}\left(\bar{\Upsilon}\,|J(\bar{\Upsilon})\right)}\,, \quad or$$

$$T_B\left(\bar{P}(T);\, x, y, Z_0\right) = T_B\left(\bar{\Upsilon}_0;\, x, y, Z_0\right) = z_{\mathrm{K}\left(\bar{\Upsilon}_0\,|\emptyset\right)} + (y-1) z_{\mathrm{K}\left(\bar{\Upsilon}_0\,|J(\bar{\Upsilon}_0)\right)}\,.$$

If r is the root of T labeled by $\odot$, and T_1, T_2 are the sons of r in T, then

$$T_B\left(\bar{P}(T);\, x, y, Z_{t_3}\right) = T_B\left(\bar{P}(T_1);\, x, y, Z'_{t_1}\right) \cdot T_B\left(\bar{P}(T_2);\, x, y, Z''_{t_2}\right),$$

where

$$z'_{\mathrm{K}_1} \cdot z''_{\mathrm{K}_2} = (x-1)^{\varrho(\odot;\,\mathrm{K}_1,\mathrm{K}_2)-\sigma(\odot)} \cdot (y-1)^{\varrho(\odot;\,\mathrm{K}_1,\mathrm{K}_2)} \cdot z_{[\mathrm{K}_1\odot\mathrm{K}_2]}$$

for each pair $\mathrm{K}_i \in \mathcal{K}_{t_i(\odot)}$, $i = 1, 2$.

Proof. This corollary follows directly from the definitions of the boundaried parse tree and of the boundaried Tutte polynomial, and from Corollary 4.1. $\square$

The last corollary provides a straightforward recursive procedure for computing the boundaried Tutte polynomial on a boundaried parse tree T in $|V(T)|$ steps; each step involving one multiplication of polynomials, and a bounded number of variable substitutions.

Remark. One may consider "element-weighted" extensions of the Tutte polynomial, defined by various researchers. It is easy to see that our Theorem 4.1 can be simply extended to handle also those polynomials. However, since such an extension does not add anything substantially new to our result, we are making no formal statements about that issue here.

5. Complexity Review

We now review running time of an algorithm based on Corollary 4.2 and Theorem 3.2. Firstly, we want to make a remark about considering the length of arithmetic operations in combinatorial algorithms: Usually, combinatorial algorithms work with integer numbers of length proportional to $\log n$ where n is the length of the input. It is common (and well justified by practical applications) to consider that one arithmetic operation in such a case takes constant time. However, when computing or evaluating the Tutte polynomial, we deal with numbers of length proportional to n, which is too much to neglect.

Theorem 5.1. *Let $u(n)$ be the number of elementary integer arithmetic operations (sum or product) needed to compute the product of two two-variable polynomials of degrees at most n. Let $v(n)$ be time needed to multiply (or sum) two n-bit integers. Assume that $\mathbb{F}$ is a finite field, t is an integer constant, and that $u(n) \geq \Theta(n^2)$. If M is an n-element rank-r $\mathbb{F}$-represented matroid of branch-width at most t given by a matrix $\boldsymbol{A} \in \mathbb{F}^{r\times n}$, then the Tutte polynomial $T(M; x, y)$ can be computed in time $O(n \cdot u(n) \cdot v(n))$.*

Proof. Firstly, we compute a $\leq t$-boundaried parse tree T of M in time $O(n^3) \leq \Theta(n \cdot u(n))$, using the algorithm of Theorem 3.2.

Then we follow the formulas in Corollary 4.2. Computing the Tutte polynomial for a parse tree with at most one vertex takes constant time. We turn our attention to the main formula $T_B\left(\bar{P}(T); x, y, Z_{t_3}\right) = T_B\left(\bar{P}(T_1); x, y, Z'_{t_1}\right) \cdot T_B\left(\bar{P}(T_2); x, y, Z''_{t_2}\right)$. Notice that the set $\mathcal{K}_{t_i}$ has bounded size for a constant $t_i \leq t$, and that the variables of Z'_{t_1}, Z''_{t_2} appear only in the first power in the above polynomials. We may compute the product of these polynomials as a bounded number of partial products, thus using only $O(u(n))$ elementary arithmetic operations. Then we apply a bounded number of variable substitutions with bounded-degree polynomials in time $O(n^2)$.

Altogether, we have to compute $O(n)$ such products of polynomials. Looking at the definition of the boundaried Tutte polynomial, we see that the coefficients of the involved polynomials are not larger that $2^{O(n)}$ since $|E(P(T))| = n$. Hence the coefficients have lengths of at most $O(n)$ bits, and so one elementary arithmetic operation with them can be done in time $O(v(n))$. Finally, we substitute $Z_t = (1, \ldots, 1)$ in time $O(n^2)$. By combining all these estimates, we get the required upper time-bound of $O(n \cdot u(n) \cdot v(n))$. $\square$

Corollary 5.1. *Assume that $\mathbb{F}$ is a finite field, and that t is an integer constant. If M is an n-element $\mathbb{F}$-represented matroid of branch-width at most t, then the Tutte polynomial $T(M;\, x, y)$ can be computed in time $O(n^6 \log n \log\log n)$.*

Proof. A two-variable polynomial of degree at most n has at most $(n+1)^2$ terms. Hence the trivial algorithm can multiply two such polynomials using $u(n) = O(n^4)$ elementary arithmetic operations with the coefficients. A well known algorithm by Strassen multiplies two n-bit numbers in time $v(n) = O(n \log n \log\log n)$. The rest follows from Theorem 5.1. □

We remark that the input size of a represented matroid M is $O(nr)$, which typically is of order n^2. It is shown in [9] that to compute the Tutte polynomial of a graph of bounded tree-width and size n takes time at least $\Omega(n^3)$. The performance of our algorithm does not match this lower bound, but it is much better than the algorithm in [1] with the exponent depending on t. We also show that, with our more general matroidal algorithm, we can almost match the performance of [9] when evaluating the Tutte polynomial at a rational point.

Corollary 5.2. *Let $\mathbb{F}$ be a finite field, and let t be an integer constant. Suppose that a, b are rational numbers given as fractions of integers $a = \frac{p_a}{q_a}$, $b = \frac{p_b}{q_b}$ such that the combined length of p_a, q_a, p_b, q_b is l bits, and that M is an n-element $\mathbb{F}$-represented matroid of branch-width at most t. Then the Tutte polynomial of M can be evaluated as $T(M;\, a, b)$ at a, b in time $O(\, n^3 + n^2 l \cdot \log(nl) \cdot \log\log(nl)\,)$.*

Proof. We slightly modify the procedure of Theorem 5.1. This time $T_B\left(\bar{P}(T);\, a, b, Z_t\right)$ is a polynomial of degree 1 with variables Z_t. Since Z_t contains a bounded number of variables, each of the polynomials in the product $T_B\left(\bar{P}(T_1);\, a, b, Z'_{t_1}\right) \cdot T_B\left(\bar{P}(T_2);\, a, b, Z''_{t_2}\right)$ has bounded number of terms, and hence we may compute the product in total time $O(v(k)) = O(k \log k \log\log k)$ where $k = O(nl)$ is the number of bits sufficient to describe the coefficients. (We do not have to consider division since we evaluate the result as a fraction of two integers.) The rest follows easily.

The reader may ask whether our procedure is valid when $a = 1$; an evaluation of $(x-1)^{\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2) - \sigma(\odot)}$ at $x = a$ could mean division by zero when substituting the $z'_{\mathrm{K}_1} \cdot z''_{\mathrm{K}_2}$ variables. (See in Corollary 4.2.) Fortunately, this cannot happen, as we show now. If $\varrho(\odot;\, \mathrm{K}_1, \mathrm{K}_2) \le \sigma(\odot) - 1$ for some $\mathrm{K}_1, \mathrm{K}_2$, then at least one of the sides in the product, say $T_B\left(\bar{N}_1;\, x, y, Z'_{t_1}\right) \restriction z_{\mathrm{K}_1}$, contains the factor $(x-1)$. Otherwise, the term $T_B\left(\bar{N};\, x, y, Z_t\right) \restriction z_{[\mathrm{K}_1 \odot \mathrm{K}_2]}$ of the Tutte polynomial for $\bar{N}$ would have the factor $(x-1)$ in a negative power by Theorem 4.1, which contradicts the definition of the polynomial. Hence the term $z'_{\mathrm{K}_1} \cdot z''_{\mathrm{K}_2}$ does not appear at all in our evaluation at $x = a = 1$. □

Noble [9] constructs an algorithm evaluating the Tutte polynomial $T(G;\, a, b)$ at a, b for a graph G of bounded tree-width in time $O((v+p) \cdot el \cdot \log e \log\log e \cdot \log l \log\log l)$, where v is the number of vertices, e is the number of edges, and p the the size of the largest parallel class in G. Note that $n = e$ in our setting. The performance of our algorithm in Corollary 5.2 is quite close to Noble [9]. In fact, one can compute the parse tree for the cycle matroid of a graph of bounded branch-width in linear time using the graphic algorithm of Bodlaender and Thilikos [2], and so one can match the performance of [9] with our algorithm for cycle matroids of simple graphs.

6. Conclusions

Finally, we want to remark that our approach is based on similar ideas as those of Noble [9], but extended to represented matroids. By using a powerful, although not simple, machinery of boundaried parse trees, we are able to provide a more straightforward recursive procedure for computing the Tutte polynomial. That is why we think that our result is interesting and useful for computing the Tutte polynomial also over graphs of bounded branch-width.

It seems to be really necessary to consider $\mathbb{F}$-represented matroids in our approach since the notion of a "k-sum" is not well defined for abstract matroids if $k > 2$, and so we have no means to define a parse tree for an abstract matroid. Moreover, considering infinite fields $\mathbb{F}$, we have proved that computation of the Tutte polynomial (more precisely, evaluation of the number of bases) is $\#P$-hard on matroids of branch-width 3 represented by rational matrices [6].

References

[1] A. Andrzejak, *An Algorithm for the Tutte Polynomials of Graphs of Bounded Treewidth*, Discrete Math. 190 (1998), 39–54.

[2] H.L. Bodlaender, D.M. Thilikos, *Constructive Linear Time Algorithms for Branch-Width*, Proceedings 24th ICALP, Lecture Notes in Computer Science 1256, 1997, 627–637.

[3] R.G. Downey, M.R. Fellows, Parametrized Complexity, Springer-Verlag New York, 1999. ISBN 0 387 94833 X.

[4] P. Hliněný, *Branch-Width, Parse Trees, and Monadic Second-Order Logic for Matroids*, submitted, 2002. Extended abstract in: STACS 2003, Lecture Notes in Computer Science 2607, Springer Verlag (2003), 319–330.

[5] P. Hliněný, *A Parametrized Algorithm for Matroid Branch-Width*, submitted, 2003.

[6] P. Hliněný, *It is Hard to Recognize Free Spikes*, submitted, 2002.

[7] F. Jaeger, D.L. Vertigan, D.J.A. Welsh, *On the Computational Complexity of the Jones and Tutte Polynomials*, Math. Proc. Camb. Phil. Soc. 108 (1990), 35–53.

[8] J.A. Makowsky, *Colored Tutte Polynomials and Kauffman Brackets for Graphs of Bounded Tree Width*, Discrete Applied Mathematics, to appear.

[9] S.D. Noble, *Evaluating the Tutte Polynomial for Graphs of Bounded Tree-Width*, Combin. Probab. Computing 7 (1998), 307–321.

[10] J.G. Oxley, Matroid Theory, Oxford University Press, 1992,1997. ISBN 0-19-853563-5

[11] W.T. Tutte, *A Ring in Graph Theory*, Proc. Camb. Phil. Soc. 43 (1947), 26–40.

[12] D.L. Vertigan, D.J.A. Welsh, *The Computational Complexity of the Tutte Plane: the Bipartite Case*, Combin. Probab. Computing 1 (1992), 181–187.

[13] D.J.A. Welsh, *The Tutte Polynomial*, Random Structures Algorithms 15 (1999), 210–228.

[14] D.J.A. Welsh, *Matroid Theory*, Academic Press, London 1976. ISBN 0 12 744050 X.