Computing the Tutte Polynomial of a Graph and the All-Terminal Network Reliability

University of Tokyo °SEKINE Kyoko
01009200 University of Tokyo IMAI Hiroshi

1 Tutte polynomial of a graph

For a graph $G = (V, E)$ with vertex set V and edge set E, the Tutte polynomial of graph G is a two-variable polynomial $T(G; x, y)$ defined by

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

where $\rho: 2^E \to \mathbf{Z}$ is the rank function of graph G, i.e., $\rho(A)$ is the number of edges in a spanning forest of subgraph of edge set A. The problem of computing the Tutte polynomial of a graph has been a hot topic in recent years. The following invariants are some special cases of the Tutte polynomial, and the values of this polynomial at some specific points have important meanings (see [6]).

(Special cases of the Tutte polynomial)

- the chromatic polynomial, flow polynomial of a graph
- the all-terminal network reliability
- the Jones polynomial of an alternating link
- the partition function of a Q-state Potts model
- the weight enumerator of a linear code over $GF(q)$

(the Tutte polynomial at specific points)

- $T(G; 1, 1)$ counts the number of trees of G, which is polynomially computable.
- $T(G; 2, 1)$ counts the number of forests of G, which is #P-hard to compute.

The computation problem of the Tutte polynomial is #P-hard in general [6]. There have been known only algorithms which require at least time proportional to the number of trees of a given graph, which is intrinsically exponential. Recently, a polynomial-time randomized approximation scheme is proposed for dense graphs [1] and densely connected graphs [2]. Although it may compute an approximate value for large dense graphs, there is no algorithm which can exactly compute the Tutte polynomial of a graph of moderate size.

In this paper, we present a new algorithm by utilizing a fact that many 2-isomorphic minors appear in computing the Tutte polynomial of a graph by the so-called edge deletion/contraction formula. By this algorithm, we can compute the Tutte polynomial of any graph with at most 14 vertices and $\binom{14}{2} = 91$ edges and that of a planar graph such as 12×12 lattice graph with 144 vertices and $2 \cdot 12 \cdot 11 = 264$ edges by a standard workstation within about an hour.

Since the all-terminal network reliability is a special case of the Tutte polynomial of a graph, the above results carry over to the computation problem of all-terminal network reliability. We can generalize this approach to compute many types of network reliability.

2 Algorithm and Computational Results

For an edge e in E, we denote by $G\backslash e$ the graph obtained by deleting e from G, and by G/e the graph obtained by contracting e from G. For an edge e in E, the following recursive formula holds.

$$T(G;x,y) = \begin{cases} xT(G/e;x,y) & e\text{: coloop} \\ yT(G\backslash e;x,y) & e\text{: loop} \\ T(G\backslash e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

Here, a loop is an edge connecting the same vertex, and a coloop is an edge whose removal decreases the rank of the graph by 1.

Suppose we apply the recursive formula in the order of $e_1, e_2, \ldots, e_m$ ($m = |E|$) in a top-down fashion, which forms a binary expansion tree. Nodes in the i-th level in the expansion tree correspond to minors of G on $\{e_{i+1}, e_{i+2}, \ldots, e_m\}$ (the 0-th level is a root). These minors consist of the same edge set, and some of them may be *2-isomorphic* under the identity map, i.e., their families of trees are identical. Concerning 2-isomorphic graphs, the following holds.

Lemma 1 *Two 2-isomorphic graphs have the same Tutte polynomial.*

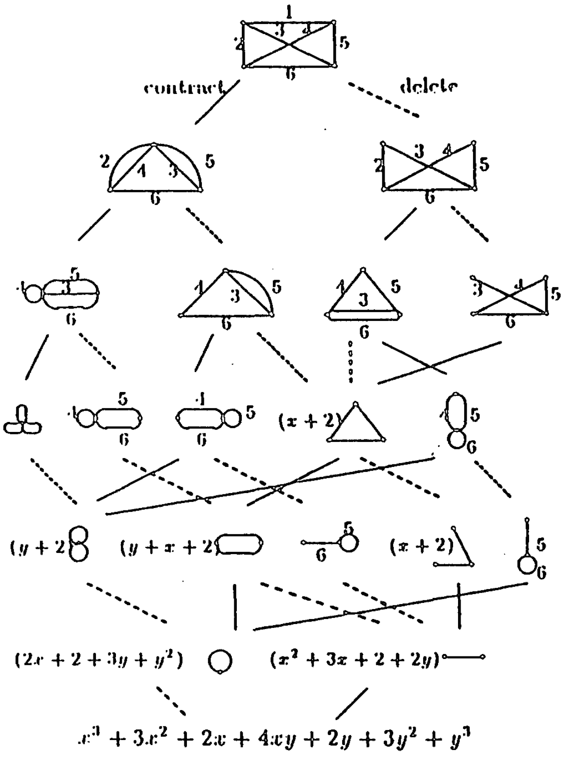

Figure 1: BDD of trees for K_4

Hence, in the i-th level in this expansion, we may represent 2-isomorphic minors among them by one of these members. By this modification, redundant computation of the same Tutte polynomial of 2-isomorphic graphs can be removed. Furthermore, testing the 2-isomorphism of two minors of the same edge set under the identity map can be done by testing the equivalence of partitions of a vertex subset induced by deletions and contractions. For details, see [3, 4, 5]. Then, we can share the same substructures in the expansion by using the 2-isomorphism, and obtain a directed acyclic graph instead of the expansion tree. This acyclic graph is called the *BDD* of trees. See an example of the BDD for K_4 in Fig.1 where edges should be regarded as directed downwards.

Based on the above observation, the BDD of all trees of G can be constructed in a top-down and breadth-first fashion. Then, the Tutte polynomial can also be computed as in Fig.1. The width of BDD is defined to be the maximum number of nodes in a level, which is 5 in this example. The width for K_n can be bounded by using the Bell number B_k which is the number of partitions of k-element set.

Theorem 1 *For $n \geq 12$, the width of the BDD of trees of K_n for some edge ordering is bounded by B_{n-2}*

Table 1: The size of BDD of trees of K_n

n	BDD width	B_{n-2}	BDD size	number of trees
2	1	-	2	1
3	2	(1)	6	3
4	5	(2)	20	16
5	14	(5)	67	125
6	42	(15)	225	1296
7	130	(52)	774	16807
8	406	(203)	2765	262144
9	1266	(877)	10292	4782969
10	3926	[4140]	39891	10^8
11	15106	[21147]	160837	$\approx 2.36 \times 10^9$
12	65232	115975	673988	$\approx 6.20 \times 10^{10}$
13	279982	678570	2932313	$\approx 1.79 \times 10^{12}$
14	1191236	4213597	13227701	$\approx 5.67 \times 10^{13}$

We have computed the size of BDD of trees of K_n up to $n = 14$ by the algorithm proposed here. The values are given in Table 1. The efficiency of our algorithm can be seen from these numbers well. We can generalize this for a general case.

Corollary 1 *For any simple connected graph G with n vertices ($n \geq 10$), there exists an edge ordering for which the width of the BDD of trees of G is bounded by B_{n-2}.*

Next, consider a $k \times k$ lattice $L_{k,k}$. For this very typical planar graph, by using a natural edge ordering, we can show the following, where C_k is the Catalan number defined to be $\frac{1}{k}\binom{2k-2}{k-1}$.

Theorem 2 *The width of BDD of $L_{k,k}$ is at most C_{k+1}.*

Again, to demonstrate that our algorithm can solve moderate-size cases here, we have computed the size of BDD of trees of $L_{k,k}$ up to $k = 12$, i.e., up to 144 vertices, by our algorithm. The values are given in Table 2. The efficiency of our algorithm can be seen from these numbers quite well. This result can be further generalized for planar graphs.

Table 2: BDD of trees of $k \times k$ lattice graph $L_{k,k}$ of $n = k^2$ vertices

k	n	BDD width ($= C_{k+1}$)	BDD size BDD	number of trees
2	4	2	8	4
3	9	5	47	192
4	16	14	252	100352
5	25	42	1260	557568000
6	36	132	6002	$\approx 3.26 \times 10^{13}$
7	49	429	27646	$\approx 1.99 \times 10^{19}$
8	64	1430	124330	$\approx 1.26 \times 10^{26}$
9	81	4862	549382	$\approx 8.32 \times 10^{33}$
10	100	16796	2395385	$\approx 5.69 \times 10^{42}$
11	121	58786	10336173	$\approx 4.03 \times 10^{52}$
12	144	208012	44232654	$\approx 2.95 \times 10^{63}$

Theorem 3 *The Tutte polynomial of a planar graph of n vertices can be computed in $O(2^{O(\sqrt{n})})$ time.*

3 All-terminal network reliability

For a connected graph $G = (V, E)$, let $R(G; p)$ be the probability that, when each edge is deleted with probability $1 - p$, the remaining graph is still connected. $R(G; p)$ is called the all-terminal network reliability. The computation of this reliability is #P-hard in general, and there are many papers on computing lower bounds, etc.

$R(G; p)$ can be expressed by the Tutte polynomial as

$$R(G; p) = (1 - p)^{|E|-|V|+1} p^{|V|-1} T(G; 1, 1/(1 - p))$$

(see [6]), and hence our algorithm can be used to compute the reliability of a graph of moderate size. Furthermore, the algorithm can be extended to the case that probability of deletion differs at each edge. This general case will be discussed elsewhere.

References

[1] N. Alon, A. Frieze and D. J. A. Welsh: Polynomial time randomised approximation schemes for the Tutte polynomial of dense graphs. *Proc. of the IEEE Annual Symp. on Foundations of Computer Science*, 1994, pp.24–35.

[2] D. R. Karger: A randomized fully polynomial time approximation scheme for the all terminal network reliability problem. *Proc. of the 27th Annual ACM Symp. on Theory of Computing*, 1995, pp.11–17.

[3] K. Sekine: *The flow polynomial and its computation.* Master's Thesis, University of Oxford, 1993.

[4] K. Sekine, H. Imai and S. Tani: Computing the Tutte polynomial of a graph and the Jones polynomial of a knot. Preprint, 1995.

[5] S. Tani and H. Imai: A reordering operation for an ordered binary decision diagram and an extended framework for combinatorics of graphs. Lecture Notes in Computer Science, Vol.834 (1994), pp.575–583.

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