ELSEVIER

Discrete Applied Mathematics 57 (1995) 93–103

DISCRETE
APPLIED
MATHEMATICS

The complexities of the coefficients of the Tutte polynomial

J.D. Annan*

Mathematical Institute, 24-29, St. Giles, Oxford, 0X1 3LB, UK

Received 24 August 1992; revised 13 September 1993

Abstract

The complexity of calculating the coefficients of the Tutte polynomial of a graph is considered. The calculation of some coefficients is shown to be #**P**-complete, whereas some other coefficients can be computed in polynomial time. However, even for a hard coefficient, it can be decided in polynomial time whether it is less than a fixed constant.

1. Introduction and definitions

The Tutte polynomial of a graph, introduced by Tutte in 1947 [7], is a generating function which contains a great deal of information about the graph. In particular, the chromatic polynomial, the flow polynomial and the reliability polynomial are all partial evaluations of the Tutte polynomial. It is defined as follows.

Let $G = (V, E)$ be a connected graph with vertex set V and edge set E, and let $n = |V|$ and $m = |E|$. For any subset A of E we define the *rank* $r(A)$ by

$$r(A) = |V| - k(A),$$

where $k(A)$ is the number of components of the graph with edge set A and vertex set V.

Definition 1. The Tutte polynomial of G, $T(G;x,y)$, is given by

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

We will often abbreviate $T(G;x,y)$ to $T(G)$.

Note that if I is an isthmus and L a loop, then

$$T(I) = x, \qquad T(L) = y$$

* Supported by a grant from SERC. Email: annan@vax.ox.ac.uk.

SSDI 0166-218X(94)00097-2

and if we write G'_e (G''_e) for the graphs obtained by deleting (contracting) an edge e then

$$T(G) = T(G'_e) + T(G''_e), \tag{1}$$

whenever e is neither an isthmus nor a loop. When e is an isthmus or loop we use

$$T(G) = \begin{cases} xT(G''_e) & \text{if } e \text{ is an isthmus,} \\ yT(G'_e) & \text{if } e \text{ is a loop,} \end{cases} \tag{2}$$

respectively. If G consists of a single vertex and no edges, then its Tutte polynomial is equal to 1.

The following partial evaluations of the Tutte polynomial are well known.

$$P(G;\lambda) = (-1)^{n-1}\lambda T(G; 1-\lambda, 0), \tag{3}$$

$$F(G;\lambda) = (-1)^{m-n+1} T(G; 0, 1-\lambda), \tag{4}$$

$$Rel(G,p) = p^{n-1}(1-p)^{m-n+1} T\left(G; 1, \frac{1}{1-p}\right), \tag{5}$$

where P, F and Rel are the chromatic, flow and reliability polynomials, respectively of a connected graph G.

The function class **#P**, introduced by Valiant [9, 10] is the class of functions that can be calculated in polynomial time by a *counting Turing machine.* This is a nondeterministic Turing machine that counts the number of distinct accepting computations that it can perform on a particular input, and outputs this number. Any **#P**-complete function is of course **NP**-hard to compute, and in fact has been shown to be **PH**-hard [6]. So **#P**-hardness can be taken as strong evidence of intractability.

It is well known that calculating the whole Tutte polynomial is **#P**-hard. For instance, evaluating it at $(x,y) = (-2,0)$ gives (a constant times) the number of 3-colourings, which is known to be **#P**-complete [1]. In fact, the following result of Jaeger et al. [3] shows that evaluation of the Tutte polynomial at any particular point is hard, except in a few special cases.

Theorem (Jaeger et al. [3]). *Evaluating the Tutte polynomial of a graph at any particular point of the complex plane is* **#P**-*hard except when either*

(1) *the point lies on the hyperbola* $(x-1)(y-1) = 1$ *or*

(2) *the point is one of the special points* $(1,1), (-1,0), (0,-1), (-1,-1), (i,-i), (-i,i), (j,j^2), (j^2,j)$ *where* $j = e^{2\pi i/3}$.

At these special points, the evaluation can be carried out in polynomial time.

We write

$$T(G;x,y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-n+1} b_{i,j}(G)x^i y^j.$$

We may omit the argument G when there is no risk of ambiguity.

Some facts are known about the values of the $b_{i,j}$. In particular, for a connected graph G with n vertices and m edges:

- $b_{i,j}(G) \geqslant 0$ for all i, j,
- $b_{0,0}(G) = 0$ if G contains at least one edge,
- $b_{1,0}(G) = b_{0,1}(G)$ if G contains at least 2 edges,
- $b_{i,j}(G) = 0$ for all $i > n - 1$ or $j > m - n + 1$,
- $b_{n-1,0}(G) = 1$ and $b_{n-2,0}(G) = m$ if G is loopless, and
- $b_{0,m-n+1}(G) = 1$ if G is isthmus-free.

As well as being # **P**-hard, the Tutte polynomial is # **P**-easy, as the following lemma shows.

Lemma 1. *Calculating the Tutte polynomial of a graph is # **P**-easy.*

Proof. We will show that the function $f(G,i,j) = b_{i,j}(G)$ is contained in # **P**, by describing a nondeterministic Turing machine that, on input (G,i,j), has exactly $b_{i,j}(G)$ accepting computations, and runs in polynomial time.

We order the edge-set arbitrarily. For any spanning tree T of a graph, we define the *internal activity* to be the number of edges $e \in T$ such that e is the least element in the cutset of the graph defined by the partition induced by $T \setminus e$. The *external activity* is given by the number of edges $e \in E \setminus T$ such that e is the least element in the unique circuit contained in $T \cup e$. It was shown by Tutte [8] that the value of $b_{i,j}(G)$ is equal to the number of spanning trees of G with internal activity i and external activity j, and therefore the ordering of the edges in the graph is immaterial. These spanning trees can clearly be recognised in polynomial time. A nondeterministic Turing machine that nondeterministically chooses $n - 1$ edges of the input graph G and halts in an accepting state if the edges form a tree which has internal activity i and external activity j will have exactly $b_{i,j}(G)$ accepting computations, and will run in polynomial time. This shows that $f(G,i,j) = b_{i,j}(G)$ is contained in # **P**. Hence the whole polynomial is # **P**-easy to calculate. □

2. Results

As we have seen above, some of the coefficients are easy to calculate. The following theorem extends this.

Theorem 1. *For any fixed integer constant k, on input G.*

(i) *If* $0 \leqslant j \leqslant m - n + 1$, $b_{n-1-k,j}(G)$ *can be calculated in polynomial time.*

(ii) *If* $0 \leqslant i \leqslant n - 1$, $b_{i,m-n+1-k}$ *can be calculated in polynomial time.*

Proof. By repeatedly applying the deletion/contraction formulae (1) and (2) to a graph G with ordered edge-set E we can construct a rooted binary tree of depth m, where the nodes of the tree correspond to graphs. The root corresponds to G, and the sons of a node correspond to the graphs obtained by contracting or deleting the lexicographically first edge remaining in the edge-set of the graph corresponding to that node, if it is not an isthmus or loop. If the first edge is an isthmus (loop), then the node only has one son, corresponding to the contraction (deletion) of that edge. The leaves of the tree correspond to single-vertex edgeless graphs.

Every leaf contributes 1 to one of the coefficients of the Tutte polynomial of the graph. If, on the path from the root to the leaf, i isthmuses and j loops are removed, then the leaf contributes 1 to the coefficient of $x^i y^j$. Summing the contributions from all the leaves gives the Tutte polynomial.

A leaf will make a contribution to one of the coefficients $b_{n-1-k,j}$ if exactly $n-1-k$ isthmuses are encountered on the path from the root. Now, every time an edge is contracted, the number of vertices of the graph is reduced by one. Exactly $n-1$ edges (some of which are isthmuses) are contracted, as we end up with a single vertex. Therefore, for the leaf to contribute to $b_{n-1-k,j}$ for any j, then, at the nodes with two sons, the path to the leaf must go to the son corresponding to the contraction of the edge exactly k times, and to the other son each other time. There are only a polynomial number of such paths (there are at most $\binom{m}{k}$) ways of choosing the k nodes with two sons at which the edge is contracted). The value of j in the index of the coefficient is equal to the number of loops removed on the path from the root to the leaf.

Of course, the tree is too big to construct in polynomial time, but the calculation can still be easily carried out in polynomial time by performing a depth-first search of the relevant nodes, by testing each subset of k edges (which we take as the edges that we wish to contract at the nodes with two sons) to see if they correspond to a possible path in the tree, and if so, how many loops are encountered on that path. The number of such subsets is bounded by m^k and so the algorithm clearly runs in polynomial time.

We can perform an analogous computation for the coefficients $b_{i,m-n+1-l}$, as in this case we must choose to delete exactly l edges at the nodes at which we have a choice, and must then sum the number of isthmuses removed along the path to a leaf to work out the value of i. So all these coefficients can also be calculated in polynomial time. □

We now show that the other end of the polynomial is hard.

Theorem 2. *$b_{1,0}(G)$ is a # **P**-complete function.*

Proof. We have already shown that $b_{1,0}(G)$ is a function in # **P**. We must also show that it is # **P**-hard.

To prove hardness, given a graph G, we construct a family of graphs $\{G^{(k)}\}$ from it. Using an oracle for the coefficients $b_{1,0}(G^{(k)})$, we calculate the number of 3-colourings of G, a known # **P**-complete function. First, a lemma concerning the density of primes.

Lemma 2. *For $n > 80$, the product of the primes less than n^2 is greater than 3^n.*

Proof. We define $\vartheta(n) = \ln \prod_{p \leqslant n} p$, where p runs over all primes less than n, and $\psi(n) = \sum_{i=1}^{\ln n/\ln 2} \vartheta(n^{1/i})$. From [2, Ch. 22], we find that $\vartheta(n) < 2n \ln 2$ and $\psi(n) \geqslant (n/4)\ln 2$ for $n \geqslant 2$. So we have

$$\begin{aligned}\vartheta(n^2) &= \psi(n^2) - \sum_{i=2}^{2\ln n/\ln 2} \vartheta(n^{2/i})\\ &\geqslant \frac{1}{4}n^2 \ln 2 - \frac{2\ln n}{\ln 2}\cdot 2n\ln 2\\ &\geqslant n(n/4 - 4\ln n)\\ &> n\ln 3\end{aligned}$$

for $n > 80$. This completes the proof of Lemma 2. □

Proof of Theorem 2 (*continued*). For the kth prime p_k, $3 \leqslant p_k \leqslant n^2$, we construct $G^{(k)}$ as follows. Form a clique on the vertices $\{v_1, \ldots, v_{p_k+1}\}$, and add edges (u_i, v_j) for all $u_i \in V(G)$ and $4 \leqslant j \leqslant p_k + 1$.

Consider a $(p_k + 1)$-colouring of $G^{(k)}$. Each vertex in the new clique must be coloured differently, and this can be done in $(p_k + 1)!$ ways. And for every colouring of the clique, the vertices of G can only be coloured with the three colours assigned to the vertices v_1, v_2 and v_3. Also, for any fixed colouring of the clique, there is a 1–1 correspondence between 3-colourings of G and extensions of the $(p_k + 1)$-colouring of the clique to the whole graph $G^{(k)}$. So,

$$P(G^{(k)}; p_k + 1) = (p_k + 1)! \cdot P(G; 3),$$

and, using Eq. (3) relating the Tutte and chromatic polynomials, we see that

$$(-1)^{n+p_k}(p_k + 1)T(G^{(k)}; -p_k, 0) = (p_k + 1)! \cdot P(G; 3),$$

$$(-1)^{n+p_k} \sum_{i=0}^{n+p_k} b_{i,0}(G^{(k)})(-p_k)^i = p_k! \cdot P(G; 3),$$

$$(-1)^{n+p_k} \sum_{i=1}^{n+p_k} b_{i,0}(G^{(k)})(-p_k)^{i-1} = (p_k - 1)! \cdot P(G; 3).$$

Taking both sides modulo p_k gives

$$(-1)^{n+p_k} b_{1,0}(G^{(k)}) \equiv (p_k - 1)! \cdot P(G; 3) (\mathrm{mod}\, p_k)$$

and using Wilson's lemma ($(p - 1)! \equiv -1 (\mathrm{mod}\, p)$) we reach

$$b_{1,0}(G^{(k)}) \equiv (-1)^{n+p_k+1} \cdot P(G; 3) (\mathrm{mod}\, p_k).$$

Thus, since p_k is certainly odd,

$$b_{1,0}(G^{(k)}) \equiv (-1)^n \cdot P(G; 3) (\mathrm{mod}\, p_k).$$

The construction of all the $G^{(k)}$ can certainly be performed in polynomial time, as the primes less than n^2 can be found by a process of trial division in time polynomial in n.

Therefore, using an oracle that returns the coefficient of x in the Tutte polynomial of a graph, we can find the number of 3-colourings of any graph modulo all primes between 3 and n^2. Clearly, the number of 3-colourings of a nonempty graph is even (and the empty graph on n vertices has 3^n 3-colourings), as permuting the colours generates 6 distinct colourings corresponding to any partition of the vertices into colour classes. Using the Chinese remainder theorem, we obtain the number of 3-colourings of G modulo the lowest common multiple of the primes less than n^2, which is of course their product. As this product is greater than the number of possible 3-colourings of G, we know that the unique solution in $\{0, \ldots, 3^n\}$ is the actual number of 3-colourings of G.

Hence, using our oracle, we can calculate a # **P**-hard function in polynomial time and so the # **P**-completeness of calculating $b_{1,0}(G)$ has been shown. □

Since, as noted earlier, $b_{0,1}$ is equal to $b_{1,0}$, it is also # **P**-complete to compute. This hardness result can be extended to include many more coefficients.

Corollary 1. *For all nonnegative integers i, j, the coefficients $b_{1+i,j}$ and $b_{i,1+j}$ are both # **P**-complete.*

Proof. Given a graph G, we construct a new graph H by adjoining a path of length i to G, consisting of new vertices $\{v_1, \ldots, v_i\}$ and edges $v_k v_{k+1}$, $1 \leqslant k < i$, and an edge uv_1 where u is an arbitrary vertex in G. We then add j loops at vertex v_i. The coefficient of $x^{i+1}y^j$ in the Tutte polynomial of H is equal to the coefficient of x in the Tutte polynomial of G, and so calculating it is # **P**-hard. Similarly, the coefficient of $x^i y^{j+1}$ in the Tutte polynomial of H is equal to the coefficient of y in the Tutte polynomial of G, and so is also # **P**-hard. □

In fact, we can extend this even further.

Corollary 2. *For all constants α, c, $0 \leqslant \alpha < 1$, $0 < c \leqslant 1$, $b_{\lfloor \alpha n + 1 \rfloor, 0}$ and $b_{\lfloor n - n^c + 1 \rfloor, 0}$ are both # **P**-complete.*

Proof. As above, given any graph G, we add a path to it to form H. If we add a path of length $\lfloor \alpha n/(1 - \alpha) \rfloor$, $\lfloor n^{1/c} - n \rfloor$ respectively, then it is simple to check that the given coefficient of the new graph H is equal to $b_{1,0}(G)$, and the transformation is a polynomial one. □

Similar results for large values of the second index of the coefficients can be obtained by adding loops to the graph, rather than isthmuses, and if we add both loops and isthmuses, we can easily reach results like the following.

Corollary 3. *Computing* $b_{\lfloor (n-1)/2 \rfloor, \lfloor (n-m+1)/2 \rfloor}(G)$ is *#* **P**-*complete.*

3. Other coefficients, and an open problem

Some coefficients are not amenable to either of the approaches given here. For any increasing function $f(n)$, calculating $b_{n-1-f(n),j}$ by brute force calculation as used in Theorem 1 above will take a superpolynomial $(O(\binom{n}{f(n)}))$ amount of time.

On the other hand, if the function increases too slowly (slower than n^c for all $c > 0$), the technique, described above, of adding a path and loops to the graph to change the indices of the coefficients of the polynomial will cause a superpolynomial expansion of the graph and hence not be a polynomial-time reduction.

An analogous situation arises in many other problems. For instance, counting the number of k-cliques of a graph is # **P**-complete for an arbitrary function $k(n)$ (for instance, $k = \lfloor n/2 \rfloor$), but can be done in polynomial time for any fixed constant k by brute force. However, for $k = \lfloor \log n \rfloor$, the complexity is not known. This suggests the following as an open problem.

Open problem. How hard is it to calculate $b_{\lfloor n-1-\log n \rfloor, j}$, for arbitrary integer j?

4. A polynomial predicate for the Tutte coefficients

Surprisingly, given the above theorem, it is still possible to find out some information about the lowest coefficient of the Tutte polynomial. For instance, it can easily be shown that, for a connected graph G, $b_{1,0}(G) = 0$ precisely when G contains an articulation vertex, and this is easy to check in polynomial time.

Furthermore, it can also be shown that $b_{1,0}(G) = 1$ if and only if G is a series–parallel graph (i.e. G is formed by a sequence of series and parallel extensions of the 2-cycle), and again, this can be checked easily. Here we extend these results to the following theorem.

Theorem 3. *The predicate "*$b_{1,0}(G)$ *is less than k" is in* **P** *for any fixed integer k.*

In order to prove Theorem 3, we need some preliminary lemmas. We begin with some results for connected matroids that can be found in [4]. We will restate them in the terminology of graph theory. The equivalent concept in graph theory to a connected matroid is a *block.*

Definition 2. The *blocks* of a graph are loops of the graph, plus the maximal 2-vertex-connected and loopless subgraphs of the graph.

Lemma 3. *If G is a block, then for any edge e in G, at least one of* G'_e, G''_e *is a block.*

Proof. Suppose G'_e is not a block, and let C_1 be one of its blocks. For any vertices $x \in C_1$ and $y \in G'_e \backslash C_1$, there is a circuit C_{xy} in G containing both x and y. This circuit must contain e, as there is no such circuit in G'_e. So $C_{xy} \backslash e$ is a circuit of G''_e containing both x and y. This shows that all $x \in C_1$ and $y \in G'_e \backslash C_1$ are contained within the same block in G''_e, and so it follows that G''_e is a block. □

A *minor* of a graph G is a graph M that can be obtained from a subgraph of G by contracting edges. We denote this by $M \leqslant G$.

Lemma 4. *For any block G, and any minor M of G that is also a block, there is a sequence of deletions and contractions that transforms G into M in which no loops or isthmuses are deleted or contracted.*

Proof. This is proved as Corollary 4.3.7 in [4], stated in terms of connected matroids. □

Now we can prove some lemmas necessary to complete the proof of Theorem 3.

Lemma 5. *G is a block if and only if* $b_{1,0}(G) > 0$.

Proof. First, we prove the right to left implication. It was shown by Tutte [8] that if a graph consists of some subgraphs that all intersect at a single vertex, then its Tutte polynomial is equal to the product of the Tutte polynomials of these subgraphs. Clearly, if this is the case, then $b_{1,0}(G) = 0$.

For the left to right implication, note that it is trivially true for all blocks of 2 edges (there is only one, with Tutte polynomial $x + y$) and assume for a contradiction that G is a counterexample with the fewest possible number of edges, that is, G is a block and $b_{1,0}(G) = 0$. Pick an arbitrary edge e in G. It is neither an isthmus nor a loop, for G has none. We know from Lemma 3 above that at least one of G'_e and G''_e is a block, and so by the inductive hypothesis at least one of $b_{1,0}(G'_e)$ and $b_{1,0}(G''_e)$ is nonzero (and of course, both are nonnegative). So, using the deletion/contraction formula (1) we see that $b_{1,0}(G) = b_{1,0}(G'_e) + b_{1,0}(G''_e) > 0$, and the contradiction has been reached. □

So, in order to determine if the lowest coefficient of the Tutte polynomial of a graph is less than k, we only need to consider blocks.

We say a set of graphs is *minor-closed* if, for any graph in the set, all minors of that graph are also contained in the set.

The following observation is trivial.

Lemma 6. *The set* $\mathscr{F}^k$ *of blocks G for which* $b_{1,0}(G) < k$, *together with all minors of these blocks, is minor-closed.*

Proof. Any minor of a minor of a graph is itself a minor of the original graph. □

Lemma 7. *For every graph $G \in \mathscr{F}^k$, $b_{1,0}(G) < k$.*

Proof. If there is a graph $H \in \mathscr{F}^k$ such that $b_{1,0}(G) \geqslant k$, then it must be a block, and also be a minor of some block G with $b_{1,0}(G) < k$.

Using the deletion/contraction formula (1) repeatedly, we can calculate the Tutte polynomial of a graph as the sum of the Tutte polynomials of smaller graphs.

By Lemma 4, we know that we can get from any block G to any minor H that is also a block with a sequence of deletions and contractions that do not create any loops or isthmuses. We order the edges of G according to this sequence, with the edges in the sequence preceding those not in it. When we apply the deletion/contraction formulae to the edges of G in order, one of the graphs created along the way will be H. Moreover, it will have been created by a sequence of deletions and contractions not involving loops or isthmuses, and so its contribution to the Tutte polynomial of G will not have to be multiplied by a power of x or y. This shows that it is possible to express the Tutte polynomial in the following way:

$$T(G) = \sum_{H \in \mathscr{H}} T(H),$$

where $\mathscr{H}$ is a set of minors of G including H. All the coefficients of the Tutte polynomial are nonnegative integers, so $b_{1,0}(G) \geqslant b_{1,0}(H)$. Hence if G is a block which has x-coefficient less than k, then for all minors $H \leqslant G$, $b_{1,0}(H) < k$. □

Finally, we are ready to prove Theorem 3.

Proof of Theorem 3. We use the positive resolution of Wagner's conjecture, as proved by Robertson and Seymour in [5]: *Any set of finite graphs contains only a finite number of minor-minimal elements.*

So there are only a finite number of minor-minimal graphs in the complement of $\mathscr{F}^k$. Let us denote by $\mathscr{O}^k$ the set of these graphs. We call these the *obstruction set for* $\mathscr{F}^k$.

Since the set $\mathscr{F}^k$ is minor-closed, in order to check for membership in $\mathscr{F}^k$, all we need to do is check whether or not a candidate graph has any element of the obstruction set as a minor.

For a fixed M, Robertson and Seymour have also proved that there is a polynomial-time algorithm that, on input G, will decide whether or not $M \leqslant G$ [5]. So we can check for all the possible "forbidden minors" in polynomial time, and hence determine whether or not $b_{1,0}(G) < k$. □

Corollary 4. *The predicate "$b_{1,0}(G)$ equals k" is solvable in polynomial time, for arbitrary fixed k.*

Proof. This is immediate, as $b_{1,0}(G) = k$ if and only if $b_{1,0}(G) < k + 1$ and $b_{1,0}(G) \nless k$. □

We can again extend these results to cover more coefficients, in the following way. We need to strengthen the result of Lemma 4 a little.

Lemma 8. *For any block G, and any connected minor M of G, there is a sequence of deletions and contractions that transforms G into M in which no loops or isthmuses are deleted or contracted.*

Proof. We use induction on the number of blocks of the minor M.

The base case, when the minor is itself a block, corresponds to Lemma 4.

Assume that it is true when the minor has at most i blocks. For a minor with $i + 1$ blocks, we can find a sequence of deletions and contractions that realises M. Let us assume that this sequence minimises the number of loops and isthmuses removed. There is a first time in the sequence that we remove an edge e such that the resulting graph H has more than one block. If, subsequently, all the edges in one of the blocks of H are deleted or contracted, then this necessitates the removal of a loop or isthmus. But this could have been avoided by deleting or contracting all of the edges of this block immediately before removing edge e. This can be done without removing any loops or isthmuses, by Lemma 3.

Therefore, there is a sequence of deletions and contractions of edges of H that reaches M, such that the edges of each block of H induce a nonempty subgraph of M consisting of fewer than i blocks. By the inductive hypothesis, taking each block of H separately, we can delete and contract some of its edges so as to reach this subgraph without having to delete or contract a loop or isthmus. □

Corollary 5. *The predicate "$b_{i,j}(G)$ is less than k" is solvable in polynomial time for arbitrary fixed i, j and k.*

Proof. Consider the set $\mathscr{F}^k_{i,j}$, consisting of all blocks G such that $b_{i,j}(G) < k$, plus all their minors. This set is of course minor-closed. As in the proof of Theorem 3, there can be no graph H contained in $\mathscr{F}^k_{i,j}$ such that $b_{i,j}(H) \geqslant k$, because if there was, it would have to be a minor of some block G with $b_{i,j}(G) < k$. This is impossible, as we can use Lemma 8 to show that we can express the Tutte polynomial of G as the sum of the Tutte polynomials of a set of graphs including H, as before.

So again we have an obstruction set $\mathscr{O}^k_{i,j}$, with a finite number of minor-minimal elements, and can check in polynomial time whether or not any of these forbidden minors is in fact a minor of the candidate graph. □

Acknowledgements

I am grateful to my supervisor, Dominic Welsh, for his help and suggestions in producing and improving this paper.

References

[1] M.R. Garey and D.S. Johnson, Computers and Intractability (Freeman, New York, 1979).

[2] G.H. Hardy and E.M. Wright, An Introduction to the Theory of Numbers (Oxford University Press, Oxford, 1960).

[3] F. Jaeger, D.L. Vertigan, and D.J.A. Welsh, On the computational complexity of the Jones and Tutte polynomials, Math. Proc. Cambridge Philos. Soc. 108 (1990) 35–53.

[4] J.G. Oxley, Matroid Theory (Oxford University Press, Oxford, 1992).

[5] N. Robertson and P.D. Seymour, Graph minors I–XVII, J. Combin. Theory Ser. B 1985–present.

[6] S. Toda, On the computational power of PP and ⊕P, in: Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science (1989) 514–519.

[7] W.T. Tutte, A ring in graph theory, Proc. Cambridge Philos. Soc. 43 (1947) 26–40.

[8] W.T. Tutte, A contribution to the theory of chromatic polynomials, Canad. J. Math. 6 (1954) 80–91.

[9] L.G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1979) 181–201.

[10] L.G. Valiant, The complexity of reliability and enumerability problems, SIAM J. Comput. (1979) 410–421.