# COMPLEXITY AND APPROXIMABILITY OF THE COVER POLYNOMIAL

Markus Bläser, Holger Dell, and Mahmoud Fouz

**Abstract.** The cover polynomial and its geometric version introduced by Chung & Graham and D'Antona & Munarini, respectively, are two-variate graph polynomials for directed graphs. They count the (weighted) number of ways to cover a graph with disjoint directed cycles and paths, they can be thought of as interpolations between determinant and permanent, and are proposed as directed analogues of the Tutte polynomial.
Jaeger, Vertigan, and Welsh showed that the Tutte polynomial is #**P**-hard to evaluate at all but a few special points and curves. It turns out that the same holds for the cover polynomials: We prove that, in almost the whole plane, the problem of evaluating the cover polynomial and its geometric version is #**P**-hard under polynomial-time Turing reductions, while only three points in the cover polynomial and two points in the geometric cover polynomial are easy. We also study the complexity of *approximately* evaluating the geometric cover polynomial. Under the reasonable complexity assumptions **RP** $\neq$ **NP** and **RFP** $\neq$ #**P**, we give a succinct characterization of a large class of points at which approximating the geometric cover polynomial within any polynomial factor is not possible.

**Keywords.** Graph Polynomial, Counting Complexity, Approximation, Permanent, Tutte Polynomial

**Subject classification.** 68Q17, 05C99

## 1. Introduction

Graph polynomials map directed or undirected graphs to polynomials in one or more variables, such that this mapping is invariant under graph isomorphisms. Probably the most famous graph polynomials are the chromatic polynomial or its generalization, the Tutte polynomial. The chromatic polynomial is the polynomial in the variable $\lambda$ that counts the number of valid $\lambda$-colourings of a given undirected graph. The Tutte polynomial $T$ in two variables $x$ and $y$ has interpretations from different fields of combinatorics. For example, $T(G; 1, 1)$ is the number of spanning trees, $T(G; 1, 2)$ is the number of spanning subgraphs of an undirected graph $G$, and also the number of nowhere-zero flows or the Jones polynomial of an alternating link are contained in the Tutte polynomial.

While the Tutte polynomial has been established for *undirected* graphs, the cover polynomial by Chung & Graham (1995) and its geometric version by D'Antona & Munarini (2000) are analogues for the directed case. Both the Tutte and the cover polynomials satisfy similar identities such as a contraction-deletion identity and product rule, but the exact relation between the Tutte and the cover polynomials is not yet known. The cover polynomials have connections to rook polynomials and drop polynomials, but from a complexity theoretic point of view, we tend to see them as generalizations of the permanent and the determinant of a graph. The cover polynomials of a graph are weighted sums of all of its spanning subgraphs that consist of disjoint, directed, and simple cycles and paths. As it is the case for most graph polynomials, the cover polynomials are of interest because they combine a variety of combinatorial problems into one generalized theoretical framework.

**Previous Results.** Jaeger *et al.* (1990) proved that, except along one hyperbola and at nine special points, computing the Tutte polynomial is #**P**-hard. For the chromatic polynomial, a substitution instance of the Tutte polynomial, this was shown first by Linial (1986). In recent years, the complexity and approximability of the Tutte polynomial has received increasing attention: Lotz & Makowsky (2004) proved that the coloured Tutte polynomial by Bollobás & Riordan (1999) is complete for Valiant's algebraic complexity class **VNP**, Giménez & Noy (2006) showed that evaluating the Tutte polynomial is #**P**-hard even for the rather restricted class of bicircular matroids, and Goldberg & Jerrum (2008) show that the Tutte polynomial is inapproximable in large regions of the Tutte plane.

A different graph invariant that seems related to the Tutte polynomial is the weighted sum of graph homomorphisms to a fixed graph $H$, so basically, it is the number of $H$-colourings. Bulatov & Grohe (2005) and Dyer *et al.* (2006) prove that the complexity of computing this sum is #**P**-hard for most graphs $H$.

Recently, results similar to ours have been shown for many other graph polynomials. For instance, Hoffmann (2010) proved that the edge elimination polynomial defined by Averbouch *et al.* (2008) is #**P**-hard to evaluate almost everywhere. Bläser & Hoffmann (2008) showed that the interlace polynomial is #**P**-hard to evaluate except for a finite number of lines. Bläser *et al.* (2008) extended the results by Jaeger, Vertigan, and Welsh to the colored Tutte polynomial. More importantly, they introduce algebraic reductions which give stronger hardness results: Evaluation at almost all points are not just shown to be hard but the evaluation at almost all points can be reduced to the evaluation at any other of these points. Makowsky (2008), in his so-called "difficult point conjecture", conjectures that this is a general phenomenon: Every polynomial that is definable in monadic second order logic is #**P**-hard to evaluate almost everwhere provided it has at least one hard point. Many of the graph polynomials studied in the literature are definable in monadic second order logic.

Courcelle *et al.* (2001) established the result that every graph polynomial that is definable in monadic second order logic *is* linear time computable on graphs with bounded treewidth. The dependence of the running time on the treewidth is usually very high. But it is often possible to obtain better algorithms for specific graph polynomial by exploiting their particular properties, for instance by Noble (1998) and Andrzejak (1998) for the Tutte polynomial or by Bläser & Hoffmann (2009) for the interlace polynomial. For general graphs, Björklund *et al.* (2008) proved that Tutte polynomials as well as the cover polynomials *can* be computed in vertex-exponential time, that is, in time $2^{\mathcal{O}(n)}\operatorname{poly}(n)$ where $n$ is the number of vertices.

**Our Contribution.** In this paper, we show that the problem of evaluating the cover polynomial and its geometric version is #**P**-hard at all evaluation points except for three and two points, respectively, where this is easy. Moreover, we prove that, under reasonable complexity assumptions, there is no *fully polynomial randomized approximation scheme* (FPRAS) for the geometric cover polynomial at all rational points $(x, y) \in \mathbf{Q}^2$ that have the following property: there exists a graph $D$ s.t. $C(D; x, y) = 0$. The only exceptions are three special points that are either polynomially computable or approximable and a half-line whose approximability is unknown. Interestingly, this class of inapproximable points turns out to exhibit different levels of intractability. At some points, like those on the positive $y$-axis, there exists an FPRAS for the geometric cover polynomial using an oracle for an **NP** predicate. In contrast, there are points at which approximating the geometric cover polynomial is as hard as its exact computation. In addition, we will extend some of these results to the cover polynomial by Chung and Graham.

Our main techniques for the #**P**-hardness proofs are gadgets and interpolation. Most notably, we present an elegant gadget that generalizes and simplifies the complicated XOR-gadget by Valiant (1979). Since interpolation is not approximation-preserving in general, the gadgets for the inapproximability proofs are more involved, and they are used in the reductions to amplify intractable information contained in the cover polynomial.

## 2. Preliminaries

Let $\mathbf{N} = \{0, 1, \ldots\}$. The graphs in this paper are directed *multigraphs* $D = (V, E)$ with parallel edges and loops allowed. We denote by $\mathcal{G}$ the set of all such graphs. We write $n$ for the number of vertices, and $m$ for the number of edges. Two graphs are called *isomorphic* if there is a bijective mapping on the vertices that transforms one graph into the other.

A *graph invariant* is a function $f : \mathcal{G} \to F$, mapping elements from $\mathcal{G}$ to some set $F$, such that all pairs of isomorphic graphs $G$ and $G'$ have the same image under $f$. In the case that $F$ is a polynomial ring, $f$ is called *graph polynomial*.

**Counting Complexity Basics.** Let $\Sigma = \{0, 1\}$. The class $\#\mathbf{P}$ consists of all functions $f : \Sigma^* \to \mathbf{N}$ for which there is a non-deterministic polynomial-time bounded Turing machine $M$ which has exactly $f(x)$ accepting paths on input $x$. We can extend $\#\mathbf{P}$ to include functions over the rationals that are not harder to compute than counting problems in $\#\mathbf{P}$. Specifically, we define $\#\mathbf{P}_{\mathbf{Q}}$ as the class of all mappings $f : \Sigma^* \to \mathbf{Q}$, such that $f = \frac{a}{b}$, where $a : \Sigma^* \to \mathbf{N}$ and $b : \Sigma^* \to \mathbf{Q}$ are counting problems with $a \in \#\mathbf{P}$ and $b \in \mathbf{FP}$. Here, $\mathbf{FP}$ is the class of polynomially computable functions.

For two counting problems $f, g : \Sigma^* \to \mathbf{Q}$, we say $f$ *Turing-reduces* to $g$ in polynomial time ($f \preccurlyeq^{\mathrm{p}}_{\mathrm{T}} g$), if there is a deterministic oracle Turing machine which computes $f$ in polynomial time with oracle access to $g$. If the oracle is used only once, we say $f$ *many-one reduces* to $g$ ($f \preccurlyeq^{\mathrm{p}}_{\mathrm{m}} g$), and if the oracle output is the output of the reduction, we speak of a *parsimonious* many-one reduction ($f \preccurlyeq^{\mathrm{p}} g$). The notions of $\#\mathbf{P}$-hardness and $\#\mathbf{P}$-completeness (under polynomial-time Turing reductions) are defined in the usual way. We will mainly use Turing reductions in our work. Of course, it would be desirable to get hardness results under many-one or even parsimonious reductions; however, we need interpolation to construct our reductions.

Although we formulate our reductions in terms of Turing-reductions over $\mathbf{Q}$ which is represented as strings over a binary alphabet, they are actually of a uniformly algebraic nature. In particular, they can be transferred effortlessly to, say, the BSS-model over $\mathbf{R}$ or $\mathbf{C}$ (cf. Blum *et al.* (1998)). “Algebraic nature” essentially means that our reductions map graphs to graphs (via polynomial time computable functions) and points to points (via rational functions).

**Approximability Basics.** For many optimization problems $f : \Sigma^* \to \mathbf{Q}$, no polynomial time algorithm is known to compute $f$ exactly. Still, in many cases, there exist algorithms that compute the value of $f$ approximately. In some sense, the best one can hope for in these cases is a *fully polynomial randomized approximation scheme* (FPRAS). A randomized approximation scheme for $f$ is a randomized algorithm that takes as input, besides the instance $x \in \Sigma^*$, an error parameter $\epsilon > 0$, and outputs a number $z \in \mathbf{Q}$ such that

$$\Pr\Big[\,|f(x) - z| \leq \epsilon\,|f(x)|\,\Big] \geq \tfrac{3}{4}. \tag{2.1}$$

It is said to be a *fully polynomial* randomized approximation scheme if its running time is bounded by a polynomial in $|x|$ and $\epsilon^{-1}$.

An interesting result (Valiant & Vazirani 1986, Corollary 3.6) states that #3SAT has an FPRAS that uses an oracle to an **NP** predicate. Since any problem in #**P** can be parsimoniously reduced to #3SAT (see Papadimitriou (1994) for details), we get the following corollary.

COROLLARY 2.2. *If $f \in \#\mathbf{P}_{\mathbf{Q}}$, then there exists an FPRAS for $f$ using an oracle to an* **NP** *predicate.*

PROOF. Let $f = \frac{a}{b}$ with $a \in \#\mathbf{P}$ and $b \in \mathbf{FP}$. Given input $x$, reduce $x$ parsimoniously to an instance $y$ of #3SAT. Apply the #3SAT-FPRAS with oracle access to the **NP**-predicate to $y$. Divide the result by $b(x)$. □

Parsimonious reductions from $f$ to $g$ are particularly useful as they allow the transformation of an FPRAS for $g$ into an FPRAS for $f$. They are *approximation preserving*. In general, a Turing reduction from $f$ to $g$ is said to be approximation preserving if it provides a $c$-approximation (i.e., an approximate result that is at most a factor of $c$ away from the exact result) for $f$ whenever the oracle is replaced by a $c$-approximation algorithm for $g$. If $f$ can be reduced to $g$ via an approximation preserving reduction, we write $f \leq_{\mathrm{AP}} g$. From the definition, we get the following easy, but useful fact.

FACT 2.3. *Let $f, g : \Sigma^* \to \mathbf{Q}$. If $f \leq_{\mathrm{AP}} g$ and there exists an FPRAS for $g$, then there also exists an FPRAS for $f$.*

What if there exists no FPRAS for a function $f$? Can we still hope for a randomized approximation algorithm that yields, say, constant factor approximations for $f(x)$? A surprising result states that this is not possible for a common class of problems, namely, *self-reducible* problems. We call a function $f$ self-reducible, if it can be evaluated by a deterministic polynomial time oracle Turing machine with oracle access to $f$ that is only allowed to query oracle strings of length less than the input length. Specifically, Jerrum & Sinclair (1989) prove the following 'all-or-nothing' theorem.

THEOREM 2.4 (Jerrum and Sinclair). *For a self-reducible function $f : \Sigma^* \to \mathbf{Q}$, if there is a polynomial time randomized algorithm for computing $f$ within a factor of poly($|x|$), then there is also an FPRAS for $f$.*

We define **RFP** to be the class of all functions computable by a **BPP**-machine, i.e., computable in polynomial time with error probability smaller than $\frac{1}{4}$.

When can we rule out the existence of an FPRAS for a counting problem $f$? The following lemma shows that, under the complexity assumption $\mathbf{RP} \neq \mathbf{NP}$, no counting version of an **NP**-complete problem can have an FPRAS.

LEMMA 2.5. *Let $f : \Sigma^* \to \mathbf{N}$ be a counting problem so that the associated decision problem $L = \{x\colon f(x) > 0\}$ is* **NP***-complete.*

*Then there exists no FPRAS for $f$ unless* $\mathbf{RP} = \mathbf{NP}$.

PROOF. An FPRAS for $f$ would directly entail $\mathbf{NP} \subseteq \mathbf{BPP}$. It is well known (see, e.g., (Papadimitriou 1994, Problem 11.5.18)) that the latter implies $\mathbf{RP} = \mathbf{NP}$. □

**Polynomials.** Polynomials $p(x_1, \ldots, x_m)$ are elements of the polynomial ring $\mathbf{Q}[x_1, \ldots, x_m]$, and, in this context, the variables are abstract objects. Any univariate polynomial can be interpolated if sufficiently many point-value pairs are known. For multivariate polynomials, this is not always true since the points must be positioned, say, in a grid. However, it is sufficient for us that the following univariate interpolation problem can be solved in polynomial time using Lagrange interpolation (where $m = 1$):
*Input:* Point-value pairs $(a_0, p_0), \ldots, (a_d, p_d) \in \mathbf{Q}^2$, encoded in binary.
*Output:* The coefficients of the polynomial $p(x)$ with $\deg(p) \leq d$ and $p(a_j) = p_j$.
Note that interpolation is not approximation preserving in general.

**Path-Cycle Decompositions.** The cover polynomials basically count a relaxed form of cycle covers, namely *path-cycle covers* or *path-cycle decompositions*. For a directed graph $D = (V, E)$ and some subset $\mathcal{PC} \subseteq E$, we denote the subgraph $(V, \mathcal{PC})$ again by $\mathcal{PC}$. A path-cycle cover of $D$ is a set $\mathcal{PC} \subseteq E$, such that, in $\mathcal{PC}$, every vertex $v \in V$ has an indegree and an outdegree of at most 1.

A path-cycle cover thus consists of disjoint simple paths and simple cycles. Note that also an independent vertex counts as a path, and an independent loop counts as a cycle.

We write $\rho(\mathcal{PC})$ and $\sigma(\mathcal{PC})$ for the number of paths and cycles of a path-cycle decomposition $\mathcal{PC}$. By the graph invariant $c_D(\rho, \sigma)$, we denote the number of path-cycle covers of $D$ that have exactly $\rho$ paths and $\sigma$ cycles. It is not hard to prove that the function $D \mapsto c_D(\rho, \sigma)$ is #**P**-complete: Counting Hamiltonian paths or cycles is #**P**-complete, that is $c_D(0, 1)$ and $c_D(1, 0)$ is #**P**-complete. We can reduce the computation of these two coefficients to the computation of $c_D(\rho, \sigma)$ by adding an appropriate number of independent vertices and/or self loops.

**Cover Polynomials.** We consider two variants of the cover polynomial: The original factorial cover polynomial defined by Chung & Graham (1995), and the more intuitive version by D'Antona & Munarini (2000).

The *factorial cover polynomial* or *Chung-Graham polynomial* is a graph polynomial in the variables $x$ and $y$, and it is defined by

$$C_{\text{fac}}(D; x, y) := \sum_{\rho=0}^{n} \sum_{\sigma=0}^{m} c_D(\rho, \sigma) x^{\underline{\rho}} y^{\sigma}, \tag{2.6}$$

where $x^{\underline{\rho}} := x(x-1)\dots(x-\rho+1)$ denotes the falling factorial. The Chung-Graham polynomial can also be written as a weighted sum over all covers of $D$:

$$C_{\text{fac}}(D; x, y) = \sum_{\substack{\text{path-cycle} \\ \text{cover } \mathcal{PC}}} x^{\underline{\rho(\mathcal{PC})}} y^{\sigma(\mathcal{PC})}.$$

The *geometric cover polynomial* or *D'Antona-Munarini polynomial* is a graph polynomial $C_{\text{geo}}(D; x, y)$ defined in a similar manner as the Chung-Graham polynomial, except that the falling factorial $x^{\underline{\rho}}$ is replaced with a usual power $x^{\rho}$. When clear from the context, we may use the notation $C(D; x, y)$ for both versions of the cover polynomial.

**Contraction-Deletion Identity.** By $D \setminus e$, we denote the *deletion*, and by $D/e$, the *contraction* of an edge $e$. If $e$ is incident to $u$ and $v$, then $D/e$ is obtained from $D$ by merging $u$ and $v$, and by removing all edges that either start at $u$ or end in $v$ (this *directed* contraction is different from a usual undirected contraction, see Figure 2.1). If $e$ is a loop then the convention is to *completely remove* the corresponding vertex when contracting $e$. An alternative way to

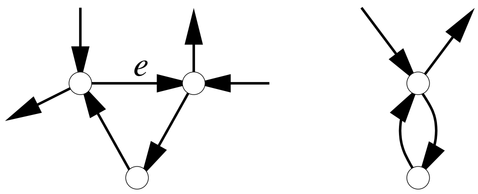



Figure 2.1: Directed contraction. The edge $e$ is contracted. The two edges are removed, because they would create a path in the new graph that is not present in the original graph.

define the cover polynomial $C(D) = C(D; x, y)$ is by a contraction-deletion identity:

$$C(D) = \begin{cases} C(D \setminus e) + C(D/e) & \text{if } e \text{ is not a loop in } D, \\ C(D \setminus e) + yC(D/e) & \text{if } e \text{ is a loop in } D, \\ x^n \text{ or } x^{\underline{n}} & \text{if there is no edge in } D. \end{cases} \tag{2.7}$$

In this recursion, $C(D \setminus e)$ counts those path-cycle covers that do not use $e$ while $C(D/e)$ counts the others. This way, every cycle shrinks to a loop and then vanishes, contributing a weight of $y$.

Using these identities, it is easy to see that, just like the Tutte polynomial, the geometric cover polynomial has a very useful product rule.

LEMMA 2.8. *Let $D_{1,2}$ be the disjoint union of two digraphs $D_1$ and $D_2$. For all $x, y \in \mathbf{Q}$, we have*

$$C_{geo}(D_{1,2}; x, y) = C_{geo}(D_1; x, y) C_{geo}(D_2; x, y).$$

## 3. Results

Our focus is to study the computational problem of evaluating the cover polynomials at fixed evaluation points $(x, y) \in \mathbf{Q}^2$. By abuse of notation, we denote by $C(x, y)$ the graph parameter $C(\,.\,, x, y)$, i.e., $x$ and $y$ are fixed and the inputs are digraphs $D$.

**Name** $C(x, y)$.

**Instance** Digraph $D = (V, E)$.

**Output** $C(D; x, y)$.

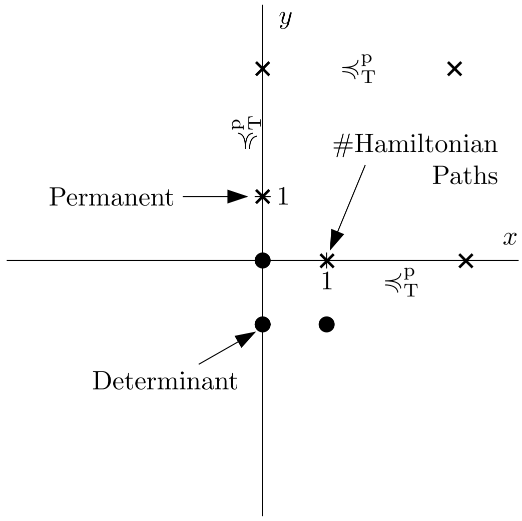



Figure 3.1: The Chung-Graham plane: three points (*black discs*) are easy to evaluate, and the rest of the plane is #**P**-hard. The *crosses* indicate the points for the reductions.

Formally, $C(x, y)$ is the function from $\mathcal{G} \to \mathbf{Q}$ with $D \mapsto C(D; x, y)$ where we assume graphs and rationals to be represented explicitly over a binary alphabet in a reasonable way. For both cover polynomials, we give a dichotomy theorem on the computational complexity of exact evaluation.

THEOREM 3.1. *The evaluations* $C(0, 0)$, $C(0, -1)$, *and* $C_{\text{fac}}(1, -1)$ *are computable in polynomial time.*

*All other evaluations* $C_{\text{geo}}(x, y)$ *and* $C_{\text{fac}}(x, y)$ *are* #**P**-*hard.*

PROOF (outline). The proof is in several steps (cf. Figure 3.1). We begin by classifying the polynomial-time computable points in Section 4.1. Furthermore, we point out that $C_{\text{fac}}(0, 1)$ is the permanent and $C_{\text{fac}}(1, 0)$ is the number of Hamiltonian paths, which both are #**P**-complete counting problems.

In Section 4.2, using elementary identities of the cover polynomial and interpolation, we reduce along horizontal lines, that means we prove $C_{\text{fac}}(0, y) \preccurlyeq_{\text{T}}^{\text{p}} C_{\text{fac}}(x, y)$ for all $x, y$ and $C_{\text{fac}}(1, 0) \preccurlyeq_{\text{T}}^{\text{p}} C_{\text{fac}}(x, 0)$ for all $x \neq 0$. This implies the hardness of $C_{\text{fac}}(x, 1)$ for all $x$ and of $C_{\text{fac}}(x, 0)$ for $x \neq 0$.

To prove the remaining hardness part where $y \notin \{-1, 0, 1\}$, we reduce the permanent to $C_{\text{fac}}(0, y)$. Section Section 4.3 is the core part of our proof, establishing this reduction $C_{\text{fac}}(0, 1) \preccurlyeq_{\text{T}}^{\text{p}} C_{\text{fac}}(0, y)$ along the $y$-axis. There we introduce and analyze the equality gadget, use it to establish a new identity for the *weighted* cover polynomial, and show how to derive a reduction for the standard cover polynomial from this.

In Section 4.4, we show how to carry over our result to the geometric version of the cover polynomial. □

REMARK 3.2. *It is easily verified that all the reductions that we use are algebraic, as defined in* Bläser et al. *(2008). Furthermore, we can reduce the evaluation at almost every point $(x', y')$ to almost every other point uniformly: First we reduce $C(x', y')$ to $C(0, y')$ using horizontal reductions and interpolation, then we reduce $C(0, y')$ to $C(0, y)$ using the vertical reduction, and then finally reducing $C(0, y)$ to $C(x, y)$.*

In the second part of this article, we analyze the inapproximability of the geometric cover polynomial. We say that $(x, y) \in \mathbf{Q}^2$ *has a root* if there exists a graph $D$ such that $C_{\text{geo}}(D; x, y) = 0$.

THEOREM 3.3. *Let $(x, y) \in \mathbf{Q}^2 \setminus \{(0, 0), (0, -1)\}$. The following holds.*

- *If $x \geq 0$ and $y = 1$, then there exists an FPRAS for $C_{\text{geo}}(x, y)$.*
- *If $1 \neq y > 0$ and $(x, y)$ has a root, then $C_{\text{geo}}(x, y)$ cannot be approximated within any polynomial factor unless* **RP** = **NP**.
- *If $y \leq 0$ and $(x, y)$ has a root, then $C_{\text{geo}}(x, y)$ cannot be approximated within any polynomial factor unless* **RFP** = **#P**.

Although we also have some results on the inapproximability of the factorial cover polynomial, they are not as nice as the theorem above, so we postpone such a theorem to the end of the article. The reason why it seems much more difficult to handle the factorial cover polynomial lies in the fact that in contrast to the geometric version, it lacks a proper product rule. As a result the gadgets designed for the geometric cover polynomial fail for the factorial version. Of course, we can obtain the factorial cover polynomial from the geometric cover polynomial by a change of basis and vice versa. For this, we need the coefficients of the polynomial which we can compute using interpolation. However, when we consider approximability, we do not get exact values but only very crude approximations to it. There is no hope of obtaining useful results by using these approximations for interpolation.

## 4. Complexity of the Cover Polynomial

**4.1. Special Points.** A *Hamiltonian path* is a path-cycle cover with exactly one path and zero cycles, and a *cycle cover* is a path-cycle cover without paths. The *permanent* $\mathrm{Perm}(D)$ is the permanent of the adjacency matrix $A$ of $D$, and it equals the number of cycle covers of $D$. The *determinant* $\det(D)$ is the determinant $\det(A)$. Remarkably, both the determinant and the permanent can be found in the cover polynomial.

LEMMA 4.1. *Let $D$ be a directed nonempty graph.*
*We have*

*(i)* $C_{\mathrm{fac}}(D;0,0) = 0$,

*(ii)* $C_{\mathrm{fac}}(D;1,0) = \#\textsc{HamiltonianPaths}(D)$,

*(iii)* $C_{\mathrm{fac}}(D;0,1) = \mathrm{Perm}(D)$,

*(iv)* $C_{\mathrm{fac}}(D;0,-1) = (-1)^n \det(D)$,

*(v)* $C_{\mathrm{fac}}(D;1,-1) = C_{\mathrm{fac}}(D;0,-1) - C_{\mathrm{fac}}(D';0,-1)$ *where $D'$ is a graph derived from $D$ by adding an apex $v_0$, that is, a fresh node and all edges to and from the nodes of $D$.*

PROOF. The proof of the first three claims is simple.
*(i)* Note that the empty graph $E_0 = (\emptyset,\emptyset)$ is the only graph that can be path-cycle covered without any paths or cycles, and that there is exactly one such cover. Because of $0^{\underline{\rho}}0^{\sigma} = 1$ if and only if $\rho = \sigma = 0$, we have

$$C(D;0,0) = \sum_{\rho,\sigma} c_D(\rho,\sigma) 0^{\underline{\rho}} 0^{\sigma} = c_D(0,0) = 0.$$

*(ii)* Using $1^{\underline{\rho}}0^{\sigma} = 1$ if and only if $\rho \in \{0,1\} \wedge \sigma = 0$, we get

$$\begin{aligned} C(D;1,0) &= \sum_{\rho,c} c_D(\rho,\sigma) 1^{\underline{\rho}} 0^{\sigma} = c_D(0,0) + c_D(1,0) \\ &= \#\textsc{HamiltonianPaths}(D). \end{aligned}$$

*(iii)* The property $0^{\underline{\rho}}1^{\sigma} = 1$ if and only if $\rho = 0$ reveals

$$C(D;0,1) = \sum_{\rho,\sigma} c_D(\rho,\sigma) 0^{\underline{\rho}} 1^{\sigma} = \sum_{\sigma} c_D(0,\sigma) = \mathrm{Perm}(D).$$

*(iv)* For the fourth claim, recall the Leibniz formula for the determinant:

$$\det(D) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_i A_{i\pi(i)}.$$

The permutations $\pi$ with $\prod_i A_{i\pi(i)} \neq 0$ stand in bijection with the cycle covers $C$ of $D$. For a cyclic permutation $\tau$ of length $\ell$, $\operatorname{sgn}(\tau) = (-1)^{\ell+1}$. Thus, $\operatorname{sgn}(\pi) = (-1)^{n+\sigma(C)}$ holds for each permutation $\pi$ and its corresponding cycle cover $C$ with $\sigma(C)$ cycles.

*(v)* For the last claim, notice that $C(D; 1, -1)$ counts all path-cycle covers with at most one path (weighted with $(-1)^{\sigma(C)}$), while the determinant $C(D; 0, -1)$ counts only cycle covers. The idea is that $C(D; 1, -1) - C(D; 0, -1)$ is the number of covers of $D$ with exactly one path, and can be expressed by $C(D'; 0, -1)$, the number of cycle covers of $D'$. This is because every path-cycle cover of $D$ with one path becomes a cycle cover in $D'$ where the path gets closed by taking a detour through the apex $v_0$ to form a cycle.

Let $D'$ be the graph obtained from $D$ by adding one additional fresh vertex $v_0$ to it, and by adding the edges $(v_0, v)$ and $(v, v_0)$ for every vertex $v$ of $D$. We claim that, for all $0 \neq y \in \mathbf{Q}$, we have

$$C(D; 1, y) \quad = \quad C(D; 0, y) + y^{-1} C(D'; 0, y).$$

First we notice that $c_{D'}(0, \sigma) = c_D(1, \sigma - 1)$ for all $\sigma > 0$: Every cycle cover $\mathcal{C}'$ of $D'$ with $\sigma$ cycles uses two edges $(v_0, u)$ and $(v, v_0)$. If we remove these two edges, we get a path-cycle cover $\mathcal{PC}$ of $D$ where the cycle that went through $v_0$ is broken up into a path from $u$ to $v$. This means that $\mathcal{PC}$ uses exactly one path and it uses one cycle less than $\mathcal{C}'$. On the other hand, assume that $\mathcal{PC}$ is a path-cycle cover of $D$ with $\sigma - 1$ cycles and 1 path. Assume the path is from $u$ to $v$. By adding the edges $(v_0, u)$ and $(v, v_0)$, we get a cycle cover of $D'$. Thus, there is a suitable bijection which implies $c_{D'}(0, \sigma) = c_D(1, \sigma - 1)$.

Now we note that $0^{\rho} = 1$ if and only if $\rho = 0$, and $1^{\rho} = 1$ if and only if $\rho = 0, 1$. And we note that $D'$ is nonempty, such that $c_{D'}(0, 0) = 0$. Then we

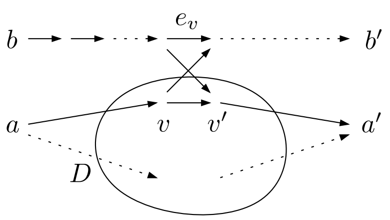



Figure 4.1: Shows the graph $D' := T(D)$ constructed in the proof of Lemma 4.3. The two edges $e_v, (v, v')$ together with the corresponding crossing edges form the *crossing gadget*.

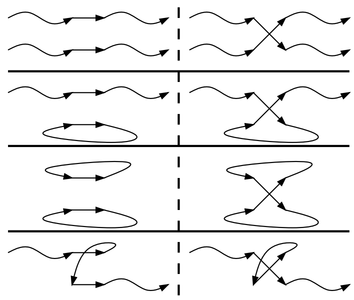


Figure 4.2: Shows the change in the number of paths and cycles in a path-cycle cover if we switch one crossing gadget.

can verify the following computation.

$$
\begin{aligned}
&C(D; 1, y) \\
&= \sum_{\rho,\sigma \geq 0} 1^{\underline{\rho}} y^{\sigma} c_D(\rho, \sigma) && \text{by definition} \\
&= \sum_{\sigma \geq 0} y^{\sigma} c_D(0, \sigma) + \sum_{\sigma \geq 0} y^{\sigma} c_D(1, \sigma) && \text{because } 1^{\underline{\rho}} = [\rho = 0 \text{ or } \rho = 1] \\
&= C(D; 0, y) + \sum_{\sigma \geq 0} y^{\sigma} c_{D'}(0, \sigma + 1) && \text{from } c_{D'}(0, \sigma) = c_D(1, \sigma - 1) \\
&= C(D; 0, y) + \sum_{\sigma \geq 1} y^{\sigma - 1} c_{D'}(0, \sigma) && \text{index transformation} \\
&= C(D; 0, y) + y^{-1} \sum_{\sigma \geq 0} y^{\sigma} c_{D'}(0, \sigma) && \text{because } c_{D'}(0, 0) = 0 \\
&= C(D; 0, y) + y^{-1} C(D'; 0, y).
\end{aligned}
$$

□

Corollary 4.2. *(i)* $C_{\text{fac}}(1, 0)$ *and* $C(0, 1)$ *are* #**P***-complete.*

*(ii)* $C(0, 0)$*,* $C(0, -1)$*, and* $C_{\text{fac}}(1, -1)$ *are polynomial-time computable.*

Proof. The first item follows from the results of Dyer *et al.* (1998) and Valiant (1979), respectively. $C(0, 0)$ is trivial and for the other two points, we can use Gaussian elimination. □

The #**P**-hardness of $C_{\text{fac}}(2, -1)$ follows from the following lemma.

LEMMA 4.3. #HAMILTONIANPATHS $\preccurlyeq^{\mathrm{p}}_{\mathrm{m}} C_{fac}(2,-1)$.

PROOF. We construct a graph transformation $T$ on the set of all directed graphs. Let $D$ be a directed graph with vertex set $V = \{1, \ldots, n\}$. $T(D)$ is constructed as follows (cf. Figure 4.1):

- uncontract all vertices $v$ from $D$, into two vertices $v, v'$ with one edge $(v, v')$ in between, and move all outgoing edges $(v, w)$ to $(v', w)$,
- add fresh vertices $a, a'$ to $D$, and add the edges $(a, v)$ and $(v', a')$ for all $v$,
- add an independent directed path of length $n$ with edges $e_1, \ldots, e_n$, and
- add the *crossing edges* $(v, w), (u, v')$ for all $(v, v')$ and $e_v = (u, w)$.

Note that uncontracting edges as above does not change the structure of the path-cycle covers. Therefore, we refer to the graph induced by the vertices $v$ and $v'$ again by $D$. Let $b$ be the tail of $e_1$, and let $b'$ be the head of $e_n$. Since $a$ and $b$ have indegree 0, they are start points of paths, and similarly $a'$ and $b'$ are end points of paths in every path-cycle cover.

For the cover polynomial of $D' := T(D)$, we have

$$C(D'; 2, -1) = 2 \cdot \sum_{\mathcal{PC}} (-1)^{\sigma(\mathcal{PC})},$$

where the sum is only over those path-cycle covers $\mathcal{PC}$ of $D'$ that have exactly two paths and an arbitrary number of cycles.

In the following, we prove $C(D'; 2, -1) = 2^{n+1} \cdot$ #HAMILTONIANPATHS$(D)$.

Let $\mathcal{PC}$ be a given path-cycle cover with two paths. In every crossing gadet, either the edges $(v, v')$ and $e_v$, or the corresponding crossing edges are in $\mathcal{PC}$. Furthermore, *switching* the presence of the edges of one gadget creates a related path-cycle cover $\mathcal{PC}'$.

Let $\mathcal{C}_0$ be the set of path-cycle covers $\mathcal{PC}_0$ of $D'$ that have exactly two paths, use *no* crossing edge, and have at least one cycle. We define the set of *bad* path-cycle covers $\mathcal{C}_b$ as the closure of $\mathcal{C}_0$ under switching arbitrary crossing gadgets. For $\vec{s} \in \{0, 1\}^n$, we denote by $\mathcal{PC}_{\vec{s}}$ the path-cycle cover obtained from $\mathcal{PC}_0 \in \mathcal{C}_0$ by switching those crossing gadgets at $u$ for which $s_u = 1$. We write $\vec{s} \oplus v$ for the vector $\vec{s}$ flipped at position $v$.

Let $\mathcal{PC}_{\vec{s}} \in \mathcal{C}_b$ be arbitrary. Let $v$ be the smallest vertex of $D$ for which $v$ and and the tail of $e_v$ do *not* belong to two different paths of $\mathcal{PC}_{\vec{s}}$. Note that there is such a vertex since otherwise $\mathcal{PC}_0$ does not contain any cycles: As depicted in Figure 4.2, the number of cycles stays unchanged if we switch crossing gadgets that are involved in two different paths. Further note that the numbers of cycles in $\mathcal{PC}_{\vec{s}}$ and $\mathcal{PC}_{\vec{s}\oplus v}$ differ by exactly 1. This implies

$$\sum_{\mathcal{PC}\in\mathcal{C}_b}(-1)^{\sigma(\mathcal{PC})} = \sum_{\mathcal{PC}_0\in\mathcal{C}_0}\sum_{\vec{s}}(-1)^{\sigma(\mathcal{PC}_{\vec{s}})} = 0,$$

since all terms $(-1)^{\sigma(\mathcal{PC}_{\vec{s}})} + (-1)^{\sigma(\mathcal{PC}_{\vec{s}\oplus v})} = 0$ cancel out.

As a result, $C(D'; 2, -1)$ is just 2 times the number of path covers of $D'$ with exactly two paths. Any such 2-path cover $\mathcal{PC}$ of $D'$ translates to an Hamiltonian path of $D$ (by switching all gadgets to $\mathcal{PC}_0$, recontracting the edges $(v, v')$, and removing $a, a'$ and the $b$-$b'$-path), and this procedure does not add any cycles. Since there are $2^n$ possible gadget states, we get

$$C(D'; 2, -1) = 2 \cdot 2^n \cdot \#\text{HamiltonianPaths}(D). \qquad \square$$

**4.2. Horizontal Reductions.** Let us consider reductions along the horizontal lines $L_y := \{(x, y) : x \in \mathbf{Q}\}$. For a directed graph $D$, let $D^{(r)}$ be the graph obtained by adding $r$ independent vertices. Corollary 4 in Chung & Graham (1995) is the core part of the horizontal-line reductions:

$$C_{\text{fac}}(D^{(r)}; x, y) = x^{\underline{r}} C_{\text{fac}}(D; x - r, y). \tag{4.4}$$

From this equation, a simple interpolation argument yields the following reduction.

Lemma 4.5. *For all $(x, y) \in \mathbf{Q}^2$, we have $C_{\text{fac}}(0, y) \preccurlyeq^{\text{p}}_{\text{T}} C_{\text{fac}}(x, y)$.*

Proof. For $x \in \mathbf{N}$, it follows directly because $C(D; 0, y) = C(D^{(x)}; y, x)/x^{\underline{x}}$.

For $x \notin \mathbf{N}$, we can compute the values $C(D; x - 1, y), \ldots, C(D; x - m, y)$ using the above identity. Since $C(D; x, b)$ is a polynomial in $x$ of degree at most $m$, this is enough to compute the coefficients of the polynomial exactly and in polynomial time. $\square$

In a similar fashion, one can also prove $C(1, 0) \preccurlyeq^{\text{p}}_{\text{T}} C(x, 0)$ for $x \neq 0$ and $C(2, -1) \preccurlyeq^{\text{p}}_{\text{T}} C(x, 0)$ for $x \neq 0, 1$ from (4.4). Please note that, together with Lemma 4.1, we obtain that $C(x, y)$ is #**P**-hard for every point $(x, y)$ on the lines $L_1$, $L_0$, and $L_{-1}$, except for the three easy points $(0, 0)$, $(0, -1)$, and $(1, -1)$.

**4.3. Vertical Reduction.** In this section, we reduce the permanent along the $y$-axis, as made explicit in the following theorem.

THEOREM 4.6. *Let $y \in \mathbf{Q}$ with $-1 \neq y \neq 0$. Then $C(0,1) \preccurlyeq^{\mathrm{p}}_{\mathrm{T}} C(0,y)$.*

PROOF (outline). For some input graph $D$, we compute $C(D;0,1)$ with oracle access to $C(0,y)$, and we use interpolation to do so.

In order to interpolate the polynomial $C(D;0,y)$, we need to compute some values $C(D;0,y_1),\dots,C(D;0,y_m)$. This can be done by using the oracle for some values $C(D_1;0,y),\dots,C(D_m;0,y)$ instead. More specifically, we construct graphs $D^\alpha$ containing $\alpha$ copies of a graph $D$, such that there is a simple relation between $C(D;0,y^\alpha)$ and $C(D^\alpha;0,y)$. Computing $C(D;0,y^\alpha)$ for $\alpha = 1,\dots,m$ and applying interpolation, we get the coefficients of $C(D;0,y)$.

Construction details are spelled out in the remainder of this section. □

The constructed graph $D^\alpha$ is a graph in which every cycle cover ideally has $\alpha$ times the number of cycles a corresponding cycle cover of $D$ would have. This way, the terms $y^c$ in the cover polynomial ideally become $y^{\alpha c}$, and some easily computable relation between $C(D;0,y^\alpha)$ and $C(D^\alpha;0,y)$ can be established.

In the construction, we duplicate the graph $\alpha$ times, and we connect the duplicates by equality gadgets. These equality gadgets make sure that every cycle cover of $D^\alpha$ is a cycle cover of $D$ copied $\alpha$ times, and thus has roughly $\alpha$ times the number of cycles. Let us construct the graph $D^\alpha$ explicitly.

- Start with the input graph $D$.
- Create $\alpha$ copies $D_1,\dots,D_\alpha$ of $D$.
- Let $e_i$ be the copy of $e$ in the graph $D_i$. Replace every tuple of edges $(e_1,\dots,e_\alpha)$ by a cascade of equality gadgets on $\alpha$ edges, which is depicted in Figure 4.4.

The $D_1$-part of every cycle cover of $D^\alpha$ is isomorphic to a cycle cover of $D$ and has to be imitated by the other subgraphs $D_i$ because of the equality gadgets. What we are left with is to prove that the equality gadget (Figure 4.3) indeed ensures that, in every cycle cover, two edges are either both present or both absent.

**4.3.1. The Weighted Cover Polynomial.** The equality gadget cannot enforce equality in *every* possible cycle cover since otherwise we could establish a parsimonious many-one reduction from #SAT to the permanent and thereby prove $\mathbf{P} = \mathbf{NP}$. We call those cycle covers *good* that satisfy equality (we later change this notion slightly). As you might have noticed, we introduce weights

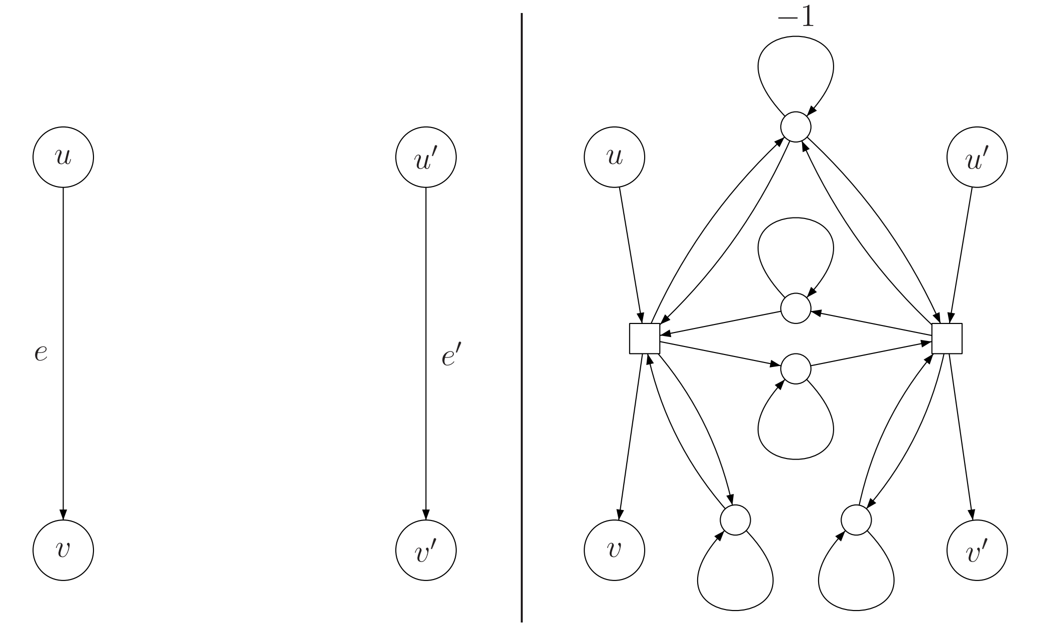



Figure 4.3: Shows how the equality gadget connects two edges $e, e'$. The smaller *circles* and *squares* represent fresh vertices.

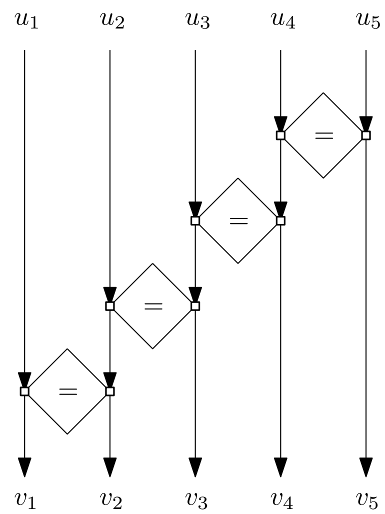



Figure 4.4: Cascade of equality gadgets connecting several edges.

$w_e \in \{-1, 1\}$ on the edges. These weights make sure that, in the *weighted* cover polynomial

$$C^w(D; 0, y) := \sum_{\text{cycle cover } C} w(C) y^{\sigma(C)} := \sum_{\text{cycle cover } C} y^{\sigma(C)} \prod_{e \in C} w_e,$$

the *bad* cycle covers sum up to 0, so *effectively* only the good cycle covers are visible. For fixed $(x, y) \in \mathbf{Q}^2$, we define the evaluation function $C^w(x, y)$ in the weighted case again as $D \mapsto C^w(D; x, y)$ and show that both evaluation complexities are equal.

LEMMA 4.7. *For all $y \in \mathbf{Q}$, it holds $C(0, y) \preccurlyeq_{\mathrm{m}}^{\mathrm{p}} C^w(0, y) \preccurlyeq_{\mathrm{T}}^{\mathrm{p}} C(0, y)$.*

PROOF. For the first part, we set all edge weights to $w_e = 1$ and notice $C(D; 0, y) = C^w(D; 0, y)$. For the second part, we are given a weighted graph $D$ as input. Conceptually, we now replace every $-1$-weight by a variable $z$, and we obtain a polynomial $p(z) = C^w(D; 0, y) = \sum_i c_i z^i$ with coefficients $c_i = c_i(y)$ and degree at most $m$. We are interested in the value $p(-1)$.

Although we cannot immediately simulate negative weights, we *can* simulate weights $z \geq 1$ by simply thickening all edges that have label $z$. Thus, we are able to compute the values of $p(z)$ for $z \in \{1, \dots, m+1\}$ using the oracle $C(0, y)$. Interpolation then enables us to compute $p(-1)$. □

**4.3.2. Partner Elimination.** In Figure 4.5, we have drawn all possible ways for an equality gadget to be covered by cycles if the surrounding graph allows it. The *state* of the gadget is the set of edges chosen to cover it. Notice that the GOOD states as well as $\text{STATE}_1^{\pm}$ and $\text{STATE}_2^{\pm}$ have the equality property, that is, the left path is completely present or absent if and only if the same holds for the right one.

As stated above, the problem is that the equality gadget does *not* prevent states that do not have the equality property. Therefore, using Lemma 4.7, we switch to the weighted cover polynomial for which our equality gadget is constructed in such a way, that the good cycle covers contribute a weight $\neq 0$ to the sum $C^w(D^\alpha; 0, y)$ while the bad cycle covers do not contribute to the sum, that is, the weights of the bad cycle covers sum up to zero. This is achieved by the fact that every bad cycle cover of weight $+1$ has a unique bad cycle cover of weight $-1$ as a partner. More specifically, every cycle cover $C$ has a partner $C'$ such that the corresponding summands in the weighted cover polynomial vanish, that is,

$$w(C) y^{\sigma(C)} + w(C') y^{\sigma(C')} = 0.$$

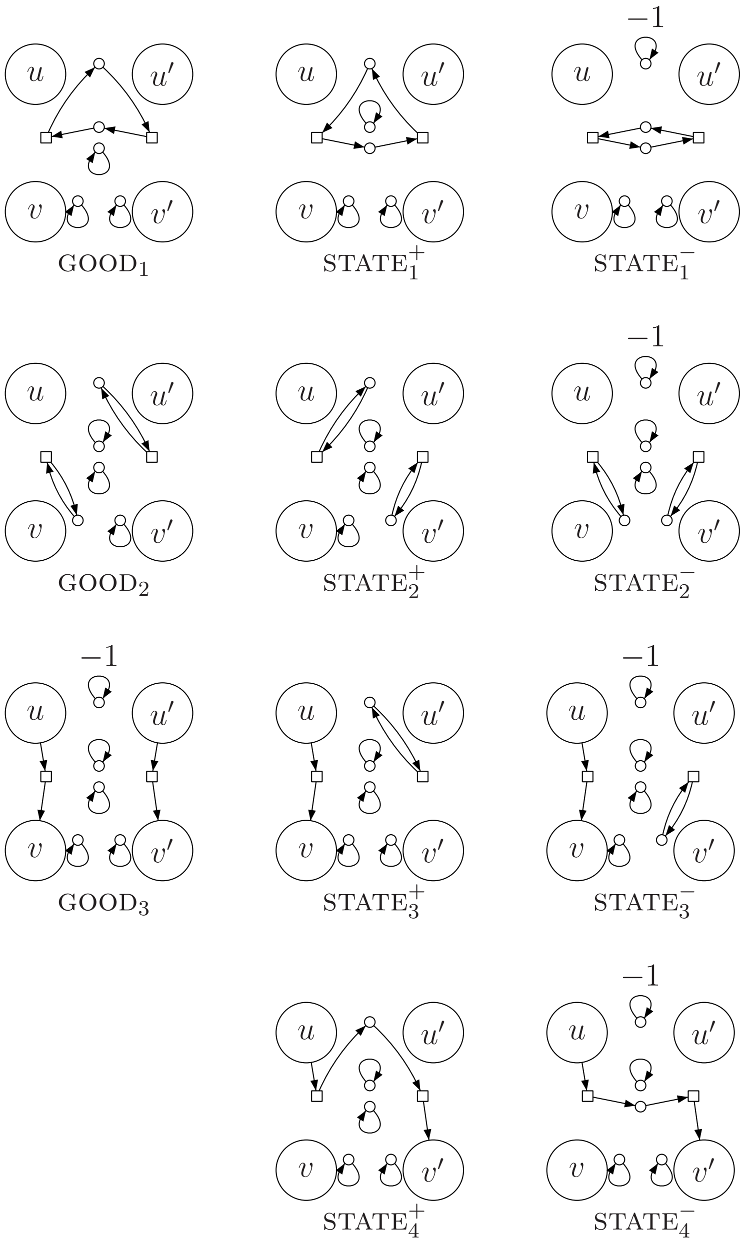



Figure 4.5: All possible states (=partial cycle covers) of the equality gadget except that the states STATE$_3^\pm$ and STATE$_4^\pm$ also have symmetric cases, which we have not drawn. We call GOOD$_1$, GOOD$_2$, and GOOD$_3$ *good* states as they have no partners and satisfy the equality property. Note that the choice of the good states is arbitrary as long as the equality property is satisfied and all STATE$_i^\pm$ have partners.

Note that not only the weights must be of different sign, but also the numbers of cycles must be equal! This is the crucial factor why we cannot simply adapt the XOR-gadget of Valiant to form an equality gadget for the cover polynomial. The number of cycles contributed by his XOR-gadget varies a lot, and thus the summands corresponding to the bad cycle covers do not cancel out. (If we plug in $y = 1$ to get the permanent, the condition on the cycles is not needed, and the XOR-gadget works, of course.)

Now let us quickly summarize and prove the properties of the equality gadget. We now call a cycle cover *bad* if an equality gadget is in a state $\text{STATE}_i^{\pm}$. The following lemma shows that there exists an involution on the set of bad cycle covers which switches the sign of the weights and leaves the number of cycles untouched.

LEMMA 4.8. *Every bad cycle cover $C$ of $D^\alpha$ has a partner $C'$ with the properties*

*(i) $C'$ is again bad, and its partner is $C$,*

*(ii) for the weights, it holds $w(C') = -w(C)$, and*

*(iii) for the number of cycles, it holds $\sigma(C') = \sigma(C)$.*

PROOF. We choose an arbitrary ordering on the equality gadgets of $D^\alpha$. Let $C$ be a bad cycle cover and $g$ be its smallest gadget in state $\text{STATE}_i^{\pm}$. We define its partner $C'$ to be the same cycle cover but with gadget $g$ in state $\text{STATE}_i^{\mp}$ instead. Verifying the three properties proves the claim. □

It immediately follows that only the good cycle covers of $D^\alpha$ remain in $C^w(D^\alpha)$:

$$C^w(D^\alpha; 0, y) = \sum_{\substack{\text{cycle cover } C,\\ C \text{ is good}}} w(C) y^{\sigma(C)}.$$

**4.3.3. Counting the Good Cycle Covers.** In order to finish the proof of Theorem 4.6, and thus also the proof of the Theorem 3.1, it remains to express $C^w(D; 0, y^\alpha)$ in an appropriate way in terms of $C^w(D^\alpha; 0, y)$.

LEMMA 4.9. *We have $C^w(D^\alpha; 0, y) = \left(y^{n+4m}(1+y)^{m-n}\right)^{\alpha-1} C^w(D; 0, y^\alpha)$.*

PROOF. Every cycle cover $C$ of $D$ induces several possible good cycle covers $C'$ in $D^\alpha$. Since $|C| = n$, a number of $(\alpha - 1)n$ gadgets have state $\text{GOOD}_3$ in $C'$. The other $m - n$ gadget have the free choice between the states $\text{GOOD}_1$

and $\text{GOOD}_2$. In addition to the $\alpha\sigma(C)$ cycles of $C'$, every gadget contributes small cycles to $C'$. More precisely, $\sigma(C') = \alpha\sigma(C) + 4\cdot\#\text{GOOD}_1 + 5\cdot\#\text{GOOD}_2 + 5\cdot\#\text{GOOD}_3$. Using the fact that there are $\binom{m-n}{i}$ possible $C'$ corresponding to $C$ and with the property $\#\text{GOOD}_1(C') = i$, the claim can be verified by the following computation.

Assuming $\#\text{GOOD}_2(C') = i$, we get

$$\begin{aligned}
\sigma(C') &= \alpha\sigma(C) + 5\cdot\#\text{GOOD}_3 + 5\cdot\#\text{GOOD}_2 + 4\cdot\#\text{GOOD}_1 \\
&= \alpha\sigma(C) + 5n(\alpha-1) + 5i + 4\left((m-n)(\alpha-1) - i\right) \\
&= \alpha\sigma(C) + i + (n+4m)(\alpha-1).
\end{aligned}$$

The weight $w(C')$ is either $-1$ or $+1$, depending on how many $-1$-loops there are. The number of these loops is just $\#\text{GOOD}_3 = (\alpha-1)n$. So for every good cycle cover $C'$ of $D^\alpha$, we have $w(C') = (-1)^{(\alpha-1)n}$. If $(\alpha-1)n$ is odd, we fix this by globally adding an independent $-1$-loop to the construction of $D^\alpha$, so that the weight of the good cycle covers is always 1.

The following sum is the part of $C^w(D^\alpha)$ that corresponds to a cycle cover $C$ of $D$. Let $\mathcal{C}(C)$ be the set of cycle covers $C'$ that correspond to $C$.

$$\begin{aligned}
\sum_{C'\in\mathcal{C}(C)} w(C')y^{\sigma(C')} &= \sum_{i=0}^{m-n} \sum_{\substack{C'\in\mathcal{C}(C),\\ \#\text{GOOD}_2(C')=i}} y^{\sigma(C')} \\
&= \sum_{i=0}^{m-n} \sum_{\substack{C'\in\mathcal{C}(C),\\ \#\text{GOOD}_2(C')=i}} y^{\alpha\sigma(C)+i+(n+4m)(\alpha-1)} \\
&= y^{\alpha\sigma(C)+(n+4m)(\alpha-1)} \sum_{i=0}^{m-n} \binom{m-n}{i} y^i \\
&= y^{\alpha\sigma(C)+(n+4m)(\alpha-1)}(1+y)^{m-n}.
\end{aligned}$$

Using this equation, we can see,

$$\begin{aligned}
C^w(D^\alpha;0,y) &= \sum_C \sum_{C'\in\mathcal{C}(C)} w(C')y^{\sigma(C')} \\
&= \sum_C y^{\alpha\sigma(C)+(n+4m)(\alpha-1)}(1+y)^{m-n} \\
&= y^{(n+4m)(\alpha-1)}(1+y)^{m-n}C^w(D;0,y^\alpha). \qquad \square
\end{aligned}$$

The easily computable relation between $C^w(D^\alpha;0,y)$ and $C^w(D;0,y^\alpha)$ from the last lemma proves (a weighted formulation of) Theorem 4.6.

Finally, the part of Theorem 3.1 concerned with the factorial cover polynomial follows, for $-1 \neq y \neq 0$, from the reduction chain

$$C(0,1) \preccurlyeq_{\mathrm{T}}^{\mathrm{p}} C^{w}(0,1) \preccurlyeq_{\mathrm{T}}^{\mathrm{p}} C^{w}(0,y) \preccurlyeq_{\mathrm{T}}^{\mathrm{p}} C(0,y) \preccurlyeq_{\mathrm{T}}^{\mathrm{p}} C(x,y).$$

**4.4. The Geometric Cover Polynomial.** The geometric cover polynomial $C_{\mathrm{geo}}(D;x,y)$ introduced by D'Antona & Munarini (2000) is the geometric version of the cover polynomial, that is, the falling factorial $x^{\underline{\rho}}$ is replaced by the usual power $x^{\rho}$ in (2.6).

$$C_{\mathrm{geo}}(D;x,y) := \sum_{\rho,\sigma} c_D(\rho,\sigma) x^{\rho} y^{\sigma}.$$

We denote by $D^{\alpha\text{-thick}}$ the $\alpha$-*thickening* of a graph $D$ in which every directed edge is replaced by $\alpha$ directed (multi-)edges. This graph operation gives a horizontal reduction for the geometric cover polynomial.

LEMMA 4.10. *For $\alpha \in \mathbf{N}_{>0}$, it holds $C_{\mathrm{geo}}(D^{\alpha\text{-thick}};x,y) = \alpha^n C_{\mathrm{geo}}(D;x/\alpha,y)$.*

PROOF. Let $\mathcal{C}$ be the set of all path-cycle covers of $D$. Every path-cycle cover $\mathcal{PC} \in \mathcal{C}$ satisfies $\rho(\mathcal{PC}) = n - |\mathcal{PC}|$ and gives rise to exactly $\alpha^{|\mathcal{PC}|} = \alpha^{n-\rho(\mathcal{PC})}$ path-cycle covers of $D^{\alpha\text{-thick}}$. Each of them has the same numbers of cycles and paths as $\mathcal{PC}$. Thus,

$$C_{\mathrm{geo}}(D^{\alpha\text{-thick}};x,y) \;=\; \sum_{\mathcal{PC}\in\mathcal{C}} \alpha^{n-\rho(\mathcal{PC})} x^{\rho(\mathcal{PC})} y^{\sigma(C)} = \alpha^n C_{\mathrm{geo}}(D;x/\alpha,y). \qquad \square$$

An immediate corollary of the lemma above, together with interpolation, is the reduction $C_{\mathrm{geo}}(x',y) \preccurlyeq_{\mathrm{T}}^{\mathrm{p}} C_{\mathrm{geo}}(x,y)$, for all $x,x',y \in \mathbf{Q}$ with $x \neq 0$. Using these horizontal reductions, we can prove a dichotomy theorem for the geometric cover polynomial.

THEOREM 4.11. *Let $(x,y) \in \mathbf{Q}^2$.*
*If $(x,y) \notin \{(0,0),(0,-1)\}$, then $C_{\mathrm{geo}}(x,y)$ is #**P**-hard.*
*Otherwise, $C_{\mathrm{geo}}(x,y)$ is computable in polynomial time.*

PROOF. On the $y$-axis, geometric and factorial cover polynomial coincide, $C_{\mathrm{fac}}(D;0,y) = C_{\mathrm{geo}}(D;0,y)$. Thus, for $x = 0$, the result follows from the factorial cover polynomial part of Theorem 3.1. For points with $x \neq 0$ and $y \neq 0,-1$, we use the horizontal reduction from above to reduce from the $y$-axis to $C_{\mathrm{geo}}(x,y)$.

For $y = 0, -1$, we again use thickenings as above to compute the polynomial $C_{\text{geo}}(D; x, y) = \sum_\rho c_\rho x^\rho$ with coefficients $c_\rho = \sum_\sigma c_D(\rho, \sigma) y^\sigma$. For $y = 0$, the coefficient $c_1 = c_D(1, 0) = \#\text{HamiltonianPaths}(D)$ is $\#\mathbf{P}$-hard. For $y = -1$, note that three coefficients can be used to compute a hard point of the *factorial* cover polynomial: $C_{\text{fac}}(D; 2, -1) = c_0 + 2c_1 + 2c_2$. □

## 5. Inapproximability of the Geometric Cover Polynomial

In this section, we study the approximability of $C_{\text{geo}}(x, y)$. In particular, we prove the following theorem.

THEOREM 5.1. *Let* $(x, y) \in \mathbf{Q}^2$ *with* $y \neq 1$ *and* $(x, y) \notin \{(0, 0), (0, -1)\}$ *such that there is a digraph* $D$ *with* $C_{\text{geo}}(D; x, y) = 0$.

*For* $y > 0$ *or* $y \leq 0$ *approximating* $C_{\text{geo}}(x, y)$ *is not possible within any polynomial factor unless* $\mathbf{RP} = \mathbf{NP}$ *or* $\mathbf{RFP} = \#\mathbf{P}$*, respectively.*

Both versions of the cover polynomial are self-reducible because the contraction-deletion identities (2.7) reduce the evaluation to smaller instances in polynomial time. Hence, we know by Theorem 2.4 that $C_{\text{geo}}(x, y)$ either has an FPRAS or is inapproximable within *any* polynomial factor.

The proof proceeds as follows. First, we establish the inapproximability of the $y$-axis. In particular, we prove that an FPRAS for the positive or negative $y$-axis would entail $\mathbf{RP} = \mathbf{NP}$ or $\mathbf{RFP} = \#\mathbf{P}$, respectively. We make an exception at three points: $C_{\text{geo}}(0, 0)$ is trivially zero, $C_{\text{geo}}(1, 0)$ is the permanent, and $C_{\text{geo}}(-1, 0)$ basically is the determinant. Second, we give an approximation preserving (horizontal) reduction from $(0, y)$ to any point $(x, y)$ for which there exists a digraph $D$ with $C_{\text{geo}}(D; x, y) = 0$. Hence, we can carry over the inapproximability results of the $y$-axis to these points. Since $C_{\text{geo}}(0, 0)$ and $C_{\text{geo}}(0, -1)$ are polynomial-time computable, we will separately establish the inapproximability of $C_{\text{geo}}(-1, 0)$ and $C_{\text{geo}}(-1, -1)$, and give an approximation preserving reduction on these two lines. Unfortunately, this approach fails for the factorial cover polynomial since the gadgets used in the reductions rely on the product rule of the geometric cover polynomial that the factorial version lacks in general.

We note that $C_{\text{geo}}(x, 1)$ can be approximated by an FPRAS for $x \geq 0$. For $x = 0$, this is due to the fact that the permanent $C_{\text{geo}}(0, 1)$ has an FPRAS, as recently discovered by Jerrum *et al.* (2004). For $x > 0$, $C_{\text{geo}}(x, 1)$ can be approximation-preservingly reduced to the *matching polynomial* $M(G; x)$, for which Jerrum & Sinclair (1997) gave an FPRAS in the case $x > 0$. Hence, we obtain the following lemma.

Lemma 5.2. *There exists an FPRAS for $C_{\mathrm{geo}}(x,1)$ if $x \geq 0$.*

**5.1. Inapproximability of the $y$-Axis.** As we will see, the $y$-axis exhibits different levels of inapproximability. Whereas its positive part is inapproximable under the reasonable assumption $\mathbf{RP} \neq \mathbf{NP}$, it turns out that approximating its negative part is as hard as $\#\mathbf{P}$. We will therefore consider both cases separately.

**5.1.1. The Positive $y$-Axis.** We consider the two cases $y \in (0,1)$ and $y > 1$ separately.

Lemma 5.3. *For $y \in (0,1)$, approximating $C_{\mathrm{geo}}(0,y)$ is not possible within any polynomial factor unless $\mathbf{RP} = \mathbf{NP}$.*

Proof. Consider $C_{\mathrm{geo}}(D;0,y) = \sum_\sigma c_D(0,\sigma) y^\sigma$. Note that $c_D(0,1)$ denotes the number of Hamiltonian cycles in $D$. As is well known, it is $\mathbf{NP}$-hard to decide whether $c_D(0,1) \neq 0$. Our goal is to amplify the contribution of this summand to the whole sum such that an approximation algorithm for $C_{\mathrm{geo}}(0,y)$ would allow us to decide whether $D$ contains a Hamiltonian cycle.

We will now demonstrate the idea of the amplification. Similar to the construction for the $\#\mathbf{P}$-hardness proof, we would want to build a graph $D^k$ containing $k$ copies of $D$ that are connected in such a way that every cycle cover of $D^k$ is (essentially) a cycle cover of $D$ copied $k$ times, and thus, has $k$ times as many cycles as the corresponding cycle cover of $D$. Assuming the existence of such a 'perfect cloning construction', we get

$$
\begin{aligned}
&\mathrm{PerfectCloning}(D,k;0,y) := \sum_\sigma c_D(0,\sigma) y^{k\sigma} \\
&= c_D(0,1) y^k + \sum_{\sigma>1} c_D(0,\sigma) y^{k\sigma} \\
&\leq y^k \big( c_D(0,1) + y^k \sum_{\sigma>1} c_D(0,\sigma) \big) && [y \in (0,1)] \\
&\leq y^k \big( c_D(0,1) + y^k 2^m \big). && [\#\text{cycle covers} \leq 2^m]
\end{aligned}
$$

By choosing $k \in \mathcal{O}(m)$ such that $y^k 2^m < \frac{1}{2}$, we have

$$\mathrm{PerfectCloning}(D,k;0,y) < y^k \big( c_D(0,1) + \tfrac{1}{2} \big).$$

Now we can see that a cloning construction amplifies Hamiltonian cycles in the following sense:

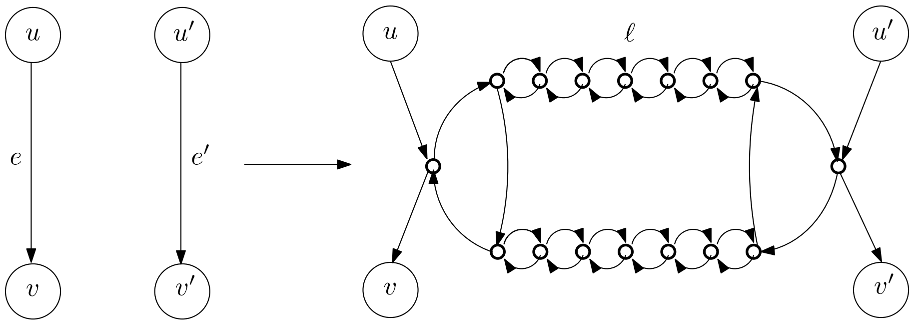



Figure 5.1: The equality gadget $E_6$ connecting two edges.

(a) If there is a Hamiltonian cycle in $D$, then $\mathrm{PerfectCloning}(D, k; 0, y) \geq y^k$.

(b) If there is no Hamiltonian cycle in $D$, then $\mathrm{PerfectCloning}(D, k; 0, y) < \frac{1}{2}y^k$.

In other words, the smallest value of $\mathrm{PerfectCloning}(D, k; 0, y)$ in case (a) exceeds the largest possible value in case (b) by a factor of 2. Hence, an FPRAS for $\mathrm{PerfectCloning}(D, k; 0, y)$ would be able to distinguish cases (a) and (b) with high probability, providing us with a polynomial time randomized algorithm for HamiltonianCycle. By Lemma 2.5, this implies **RP** = **NP**.

Unfortunately, it is unlikely that a 'perfect cloning construction' exists in the cover polynomial. However, it turns out that we can approximate it sufficiently well. In particular, we will connect the $k$ copies of $D$ by an (approximate) equality gadget which ensures that all cycle covers of $D^k$ that are not copies of a cycle cover of $D$ have a small weight contribution to $C_{\mathrm{geo}}(D^k; 0, y)$. Assuming an FPRAS for $C_{\mathrm{geo}}(0, y)$, this approximate cloning reduction provides an FPRAS for $\mathrm{PerfectCloning}(0, y)$.

It might be tempting to use the same equality gadget like in the #**P**-hardness proof. The reason why this does not work here is because the gadget there contains negative edges. In order to simulate the negative edges, we had to use interpolation. Unfortunately, this interpolation crucially relied on the exact evaluations of the weighted cover polynomial for different positive weights, i.e., it is not approximation preserving.

Instead, we use an equality gadget $E_\ell$ connecting two edges, as depicted in Figure 5.1, which ensures that (in a weighted sense) in almost all cycle covers either both edges are present or both are absent. Here, $\ell \in \mathbf{N}$ denotes the length of the upper and lower path of the gadget.

The construction of $D^k$ proceeds as follows:

1. Take $k$ copies $D_1, \dots, D_k$ of $D$.
2. Let $e_i$ be the edge in $D_i$ corresponding to $e$ in $D$. Connect all corresponding edges $\{e_1, \dots, e_k\}$ by a cascade of equality gadgets $E_\ell$, for a sufficiently (polynomially) large *even* $\ell$, as shown in Figure 4.4.

Note that each edge of $D_i$ is split into at most three edges in step 2 and that the construction uses $(k-1)m$ equality gadgets.

We now prove that the equality gadget indeed works sufficiently well to make sure that a cycle cover of $D_1$ carries over to all other $D_i$ in almost all cases (in a weighted sense). In Figure 5.2, the four possible situations of the equality gadget $E_\ell$ are drawn for even $\ell$, besides four impossible situations of bad states. It is easy to see that, for each possible configuration of the outer edges, there is at most one possible way to complete it to a cycle cover on the gadget. Since each edge of $D_i$ is split into at most three edges, it follows that there are at most $2^{3mk}$ cycle covers of $D^k$.

The crucial point is that, while in the good states (both outer paths absent or both outer paths present) each gadget only adds a single cycle to the existing cycle cover, we get $\ell + 1$ additional cycles in the bad states. So although the equality gadget cannot prevent the occurrence of a bad case completely, it will add a factor of $y^{\ell+1}$ to the weight of the corresponding cycle cover. We call cycle covers of $D^k$ *bad* if at least one gadget is in a bad state, and *good* otherwise.

Since the good cycle covers simulate perfect cloning, we get

$$\begin{aligned} C_{\text{geo}}(D^k; 0, y) &= \sum_{\text{bad cycle cover } C} y^{\sigma(C)} + \sum_{\text{good cycle cover } C} y^{\sigma(C)} \\ &\leq 2^{3mk} y^{\ell+1} + y^{(k-1)m} \cdot \text{PerfectCloning}(D, k; 0, y) \\ &< \big(1 + \frac{1}{4}\big) y^{(k-1)m} \cdot \text{PerfectCloning}(D, k; 0, y), \end{aligned}$$

by choosing $\ell \in \mathcal{O}(k^2 m^2)$ sufficiently large, such that $2^{3mk} y^{\ell+1} < \frac{1}{4} y^{km}$. If $D$ has at least one cycle cover, then $\text{PerfectCloning}(D, k; 0, y) \geq y^m$ and the second inequality above is true (if not, we just return 0). On the other hand, $C_{\text{geo}}(D^k; 0, y) \geq y^{(k-1)m} \text{PerfectCloning}(D, k; 0, y)$ since $y > 0$. Thus, an FPRAS for $C_{\text{geo}}(0, y)$ would entail an FPRAS for $\text{PerfectCloning}(0, y)$. But this implies $\mathbf{RP} = \mathbf{NP}$ as we have seen before. This finishes the case $y \in (0, 1)$. □

REMARK 5.4. *Note that the 'approximate cloning reduction' also proves that there cannot be a randomized polynomial time algorithm that approximates*

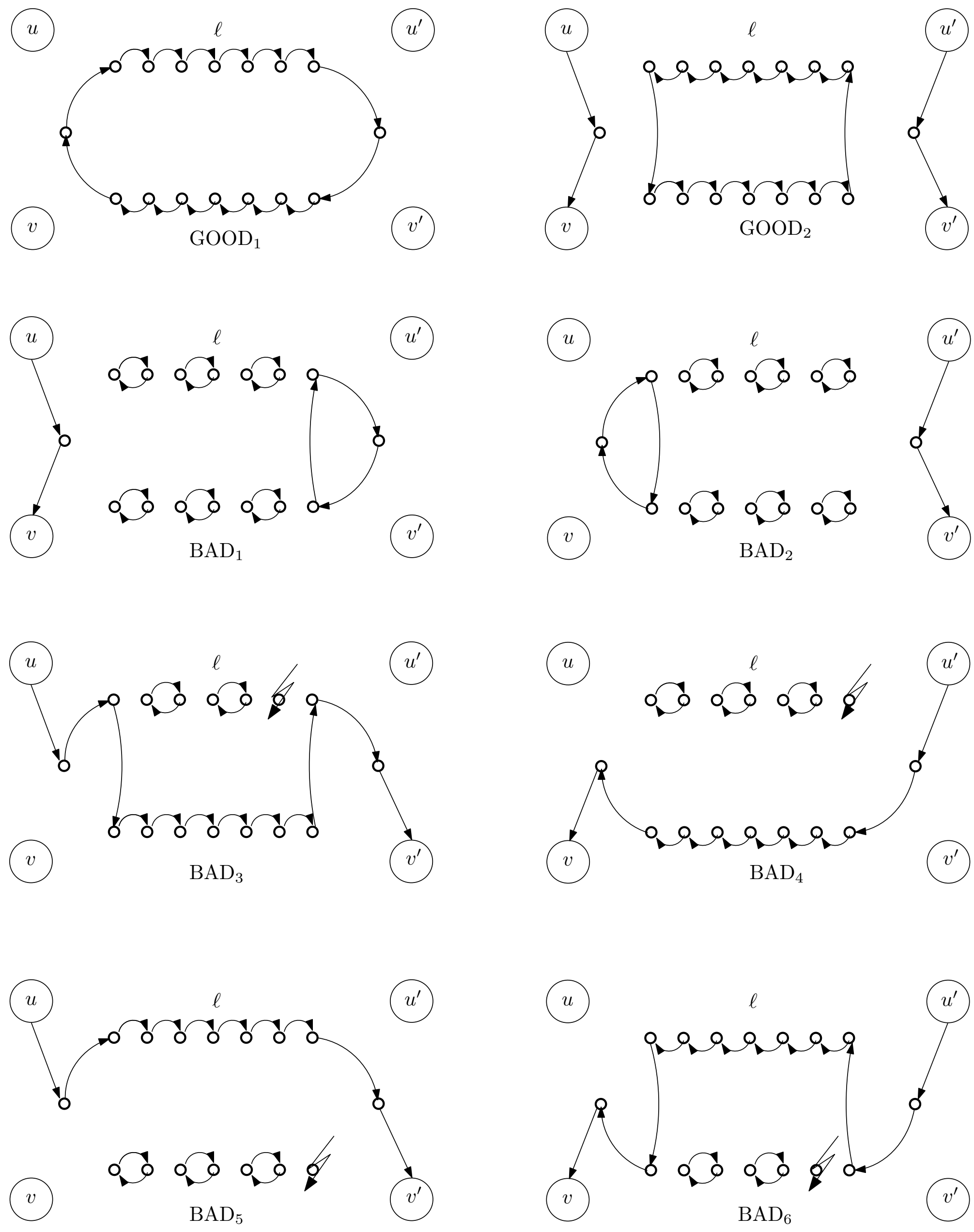



Figure 5.2: Exhaustive case analysis of equality gadget $E_\ell$ for even $\ell$. Note that bad states are either impossible or result in many cycles.

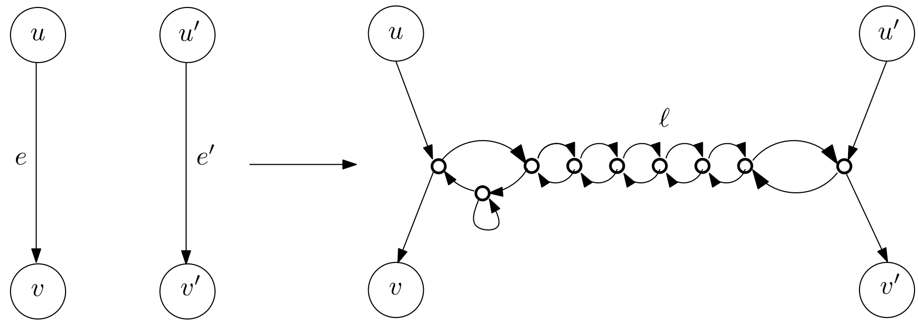



Figure 5.3: The approximate equality gadget $I_5$ connecting two edges.

*$C_{geo}(0, y)$ for $y \in (0, 1)$, even within any fixed* exponential *factor. To see this, simply set $k \in \mathcal{O}(m)$ s.t. $y^k 2^m < 2^{-cm}$ with $c > 1$ and $\ell \in \mathcal{O}(m^2)$ s.t. $2^{3mk} y^{\ell+1-(k-1)m} < y^k 2^{-cm}$.*

Assuming the existence of an equality gadget for the case $y > 1$, one might be tempted to try the same reduction as above. However, while for $y \in (0, 1)$ the cloning reduction amplifies the weight of the Hamiltonian cycles relative to the cycle covers consisting of more than one cycle, we achieve quite the opposite for $y > 1$. In particular, if $s$ is the maximum number of cycles in any cycle cover of $D$, then the summand $c_D(0, s)$ would be amplified the most. Unfortunately, $c_D(0, s)$ is not known to be hard to approximate. Instead we will reduce from Partition into Triangles:

**Name.** Partition into Triangles.

**Instance.** A simple undirected Graph $G = (V, E)$ without loops, with $|V| = 3q$ for some positive integer $q \in \mathbf{N}$.

**Question.** Can the vertices of $G$ be partitioned into $q$ disjoint sets $V_1, \ldots, V_q$, each containing exactly three vertices that are connected with each other in $G$, i.e., form a triangle?

Partition into Triangles is a classical **NP**-complete problem, see (Garey & Johnson 1979, GT11).

Lemma 5.5. *For $y > 1$, approximating $C_{geo}(0, y)$ is not possible within any polynomial factor unless* $\mathbf{RP} = \mathbf{NP}$.

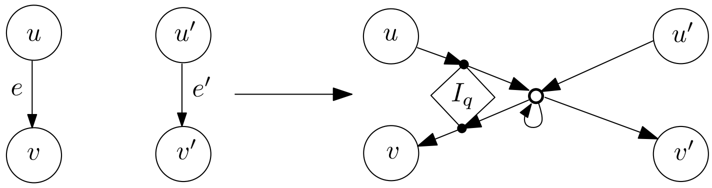



Figure 5.4: The approximate NAND gadget $N_\ell$ connecting two edges.

Proof. Let $G = (V, E)$ be an instance of Partition into Triangles. In the remainder of the proof we denote $|V|$ by $n$ and $|E|$ by $m$. Note that $G$ can be partitioned into triangles if and only if it contains a cycle cover with $\frac{n}{3}$ cycles. Such a cycle cover would have the maximum possible number of cycles since, in a simple undirected graph, each cycle has length at least 3. Thus, in order to decide whether such a cycle cover exists, we can apply a cloning reduction similar to the one used in the case $y \in (0, 1)$. Still, we need to bypass the problem that the cover polynomial is defined for directed graphs whereas $G$ is undirected. In other words, we want to construct a directed graph $D = (V', E')$ such that a polynomial approximation to $C_{\text{geo}}(D; 0, y)$ for any $y > 1$ would enable us to decide whether $G$ has a cycle cover with $\frac{n}{3}$ cycles, i.e., can be partitioned into triangles. For that purpose, we direct $G$ in the natural way by introducing one forward and one backward edge for each undirected edge of $G$. In order to make sure that each cycle cover of this directed version corresponds to a cycle cover of $G$, we connect each pair of backward and forward edges by a NAND gadget. This gadget enforces that (in a weighted sense) almost no cycle cover contains the connected pair of edges simultaneously. Finally, we apply in essence the same cloning reduction as for the case $y \in (0, 1)$ with a suitable equality gadget connecting the different copies of the directed graph. Note that in contrast to the case $y \in (0, 1)$, we need to use an equality gadget in which the good states are covered by many cycles to guarantee a large weight contribution. On the other hand, the bad states should be covered only by few cycles to guarantee a relatively small weight contribution. See Figure 5.3 and Figure 5.4 for the equality and NAND gadget, respectively. The equality gadget $I_\ell$ is parameterized by the length of its middle path. As we will see later, the loop in the equality gadget makes sure that, in the good states, all cycle covers of the gadget have the same number of cycles. Although not crucial, this property of the gadget will simplify our calculations. Note that each edge $(u, v)$ connected by such a gadget is transformed into a path $u \rightsquigarrow v$ of length 2 or 4 (left path of NAND gadget). Furthermore, it is easy to see that the cycle cover of the gadgets is uniquely determined by the presence or absence of their outer paths, see Figure 5.5 and Figure 5.6. Hence, it is sufficient for the cloning reduction to connect the corresponding outer paths of each copy of $D$ by equality gadgets. Formally, the construction of $D^k$ proceeds as follows:

1. Start with $D = (V', E')$ where $V' = V$ and $E' = \bigcup_{\{u,v\}\in E}(u, v) \cup (v, u)$
2. Connect each pair $\{(u, v), (v, u)\} \subseteq E'$ by the NAND gadget $N_\ell$ for sufficiently (polynomially) large *odd* $\ell \in \mathbf{N}$.
3. Construct $D^k$ as follows:
   (a) Take $k$ copies $D_1, \ldots, D_k$ of $D$ for a sufficiently (polynomially) large $k \in \mathbf{N}$.
   (b) Let $e_i$ be an edge in $D_i$ corresponding to an edge $e$ on an outer path of a gadget in $D$. Connect all corresponding edges $\{e_1, \ldots, e_k\}$ by a cascade of equality gadgets $I_\ell$ for a sufficiently (polynomially) large *odd* $\ell$, see Figure 4.4.

Each undirected edge $\{u, v\}$ of $G$ is transformed into two directed edges $(u, v)$ and $(v, u)$ in step 1, that are split into at most four outer edges in step 2, which in turn are copied $k$ times where each copy is split into at most three edges in step 3. In total, $D^k$ has at most $24km$ outer edges. Since the cycle cover of every gadget is uniquely determined by the outer edges of its gadgets, it follows that $D^k$ has at most $2^{24km}$ cycle covers.

We will now prove that both types of used gadgets work sufficiently well for our purposes. It is easy to see that the NAND gadget works as desired under the assumption that the equality gadget does so. In Figure 5.6, all states that are compatible with good states of the equality gadget are shown. In particular, note that only one of the two paths $u \rightsquigarrow v$ and $u' \rightsquigarrow v'$ can be present in a cycle cover since they share a common vertex in the middle. Moreover the equality gadget $I_\ell$ enforces that cycle covers in which the invalid pathes $u \rightsquigarrow v'$ or $u' \rightsquigarrow v$ are present contribute only a relatively small weight. Note that the NAND gadget provides one additional cycle if its outer paths are absent.

As for the proposed equality gadget, we prove that bad states of cycle covers where only one of the two connected edges is present have a relatively small weight. In Figure 5.5 all possible states of $I_\ell$ are illustrated for odd $\ell$, besides two impossible bad states. Note that in the good states (both edges present or absent), the equality gadget $I_\ell$ provides $\frac{\ell+3}{2}$ additional cycles to the cycle cover. On the other hand, at most one additional cycle occurs in the bad states.

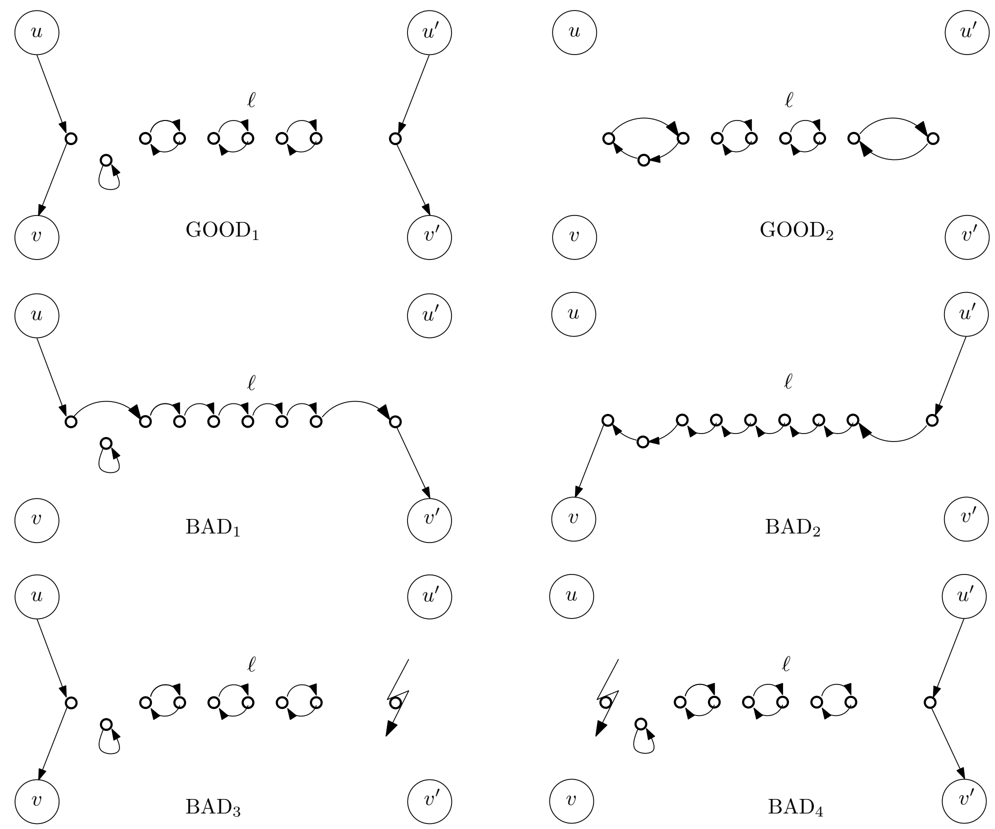



Figure 5.5: Exhaustive case analysis of equality gadget $I_\ell$ for odd $\ell$. Note that good states incur many cycles, wheras bad states are either impossible or have at most one cycle.

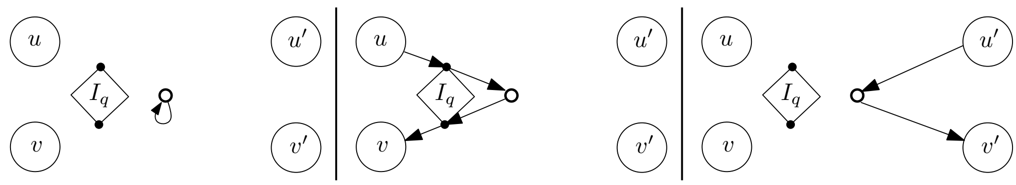



Figure 5.6: Case analysis of NAND gadget $N_\ell$ for odd $\ell$. Bad states resulting from bad states of the equality gadget are omitted.

As before, we call cycle covers of $D^k$ *good* if all gadgets are in a good state and *bad* otherwise. Hence, good cycle covers of $D^k$ (essentially) consist of $k$ copies of cycle covers of $D$ where each cycle has length at least 3.

First, we note that $D^k$ contains $k \cdot m$ $N_\ell$ gadgets and $(k-1)\cdot 2m$ additional $I_\ell$ gadgets. Since each $N_\ell$ gadget is made up of one $I_\ell$ gadgets, we have in total $3km - 2m$ $I_\ell$ gadgets. Thus, on the one hand, a good cycle cover gets a weight factor of exactly $y^{(3km-2m)\frac{\ell+3}{2}}$ from its equality gadgets. Furthermore, it gets a weight factor of $y^{k(m-n)}$ from its NAND gadgets since there are $m-n$ pairs of connected edges that are absent in a cycle cover. On the other hand, a bad cycle cover gets at most a weight factor of $y^{(3km-2m-1)\frac{\ell+3}{2}+1}$ from its equality gadgets since, by definition, at least one of its equality gadgets is in a bad state. Apart from that, it gets at most a weight factor of $y^{km}$ from its NAND gadgets. Besides the cycles from the gadgets, a bad cycle cover can have at most $\frac{1}{2}kn$ cycles, since we assumed that $G$ has no loops.

Following the now familiar path,

$$\begin{aligned}
C_{\mathrm{geo}}(D^k;0,y) &= \sum_{\text{bad cycle cover C}} y^{\sigma(C)} + \sum_{\text{good cycle cover C}} y^{\sigma(C)} \\
&\le 2^{24km} y^{(3km-2m-1)\frac{\ell+3}{2}+1} y^{km} y^{\frac{1}{2}kn} && \big[\text{\# of cycle covers} \\
&\quad + y^{(3km-2m)\frac{\ell+3}{2}} y^{k(m-n)} \sum_{\sigma} c_D(0,\sigma) y^{k\sigma} && \text{is} \le 2^{24km}\big] \\
&= y^{(3km-2m)\frac{\ell+3}{2}+k(m-n)} \big(2^{24km} y^{\frac{1}{2}(3kn-\ell-1)} && \big[\text{good cycle cover} \\
&\quad + c_D(0,\tfrac{n}{3}) y^{\frac{1}{3}kn} + \sum_{\sigma<\frac{n}{3}} c_D(0,\sigma) y^{k\sigma}\big) && \text{has} \le \tfrac{n}{3} \text{ cycles}\big] \\
&\le y^{(3km-2m)\frac{\ell+3}{2}+k(m-n)} \big(2^{24km} y^{\frac{1}{2}(3kn-\ell-1)} && \big[\text{\# of cycle covers} \\
&\quad + c_D(0,\tfrac{n}{3}) y^{\frac{1}{3}kn} + 2^{2m} y^{k(\frac{n}{3}-1)}\big) && \text{of } D \text{ is} \le 2^{2m}\big] \\
&= y^{(3km-2m)\frac{\ell+3}{2}+k(m-n)+\frac{1}{3}kn} \big(2^{24km} y^{\frac{1}{2}(\frac{7}{3}kn-\ell-1)} \\
&\quad + c_D(0,\tfrac{n}{3}) + 2^{2m} y^{-k}\big).
\end{aligned}$$

By setting $k \in \mathcal{O}(m)$ s.t. $2^{2m}y^{-k} < \frac{1}{4}$ and $\ell \in \mathcal{O}(m^2)$ s.t. $2^{24km}y^{\frac{1}{2}(\frac{7}{3}kn-\ell-1)} < \frac{1}{4}$, we have

$$C_{\mathrm{geo}}(D^k;0,y) < y^{(3km-2m)\frac{\ell+3}{2}+k(m-n)+\frac{1}{3}kn}\big(\tfrac{1}{2} + c_D(0,\tfrac{n}{3})\big).$$

We consider the two complementary situations:

(a) There exists a partition of $G$ into triangles. Then $c_D(0,\frac{n}{3}) \ge 1$ and

$$C_{\mathrm{geo}}(D^k;0,y) \ge y^{(3km-2m)\frac{\ell+3}{2}+k(m-n)+\frac{1}{3}kn}.$$

(b) There exists no partition of $G$ into triangles. Then $c_D(0, \frac{n}{3}) = 0$ and

$$C_{\text{geo}}(D^k; 0, y) < \tfrac{1}{2} y^{(3km-2m)\frac{\ell+3}{2}+k(m-n)+\frac{1}{3}kn}.$$

By the same argument as in the case $y \in (0, 1)$, the existence of an FPRAS for $C_{\text{geo}}(0, y)$ for $y > 1$ would entail $\mathbf{RP} = \mathbf{NP}$. □

**5.1.2. The Negative $y$-Axis.** We can easily adapt the above proof for the corresponding negative cases. In particular, we can use the same gadgets for our reductions. We refrain from doing so since, for $y < 0$, we will prove the even stronger statement that approximating $C_{\text{geo}}(0, y)$ is as hard as $\#\mathbf{P}$. Assuming that $\#\mathbf{P}$ is a much bigger class than $\mathbf{RFP}$, this is a very different kind of inapproximability. As noted before (see Corollary 2.2), every problem in $\#\mathbf{P}_{\mathbf{Q}}$ has a randomized polynomial approximation scheme using an oracle for an $\mathbf{NP}$ predicate. Based on this fact, our result implies that, for negative $y$, $C_{\text{geo}}(0, y)$ is not in $\#\mathbf{P}_{\mathbf{Q}}$, under a reasonable complexity assumption.

THEOREM 5.6. *For $y \in \mathbf{Q}^- \setminus \{-1\}$, approximating $C_{geo}(0, y)$ is not possible within any polynomial factor unless $\mathbf{RFP} = \#\mathbf{P}$.*

PROOF (sketch). Assuming the existence of an FPRAS for $C_{\text{geo}}(0, y)$, we will show how to compute the exact value of $C_{\text{geo}}(0, y)$ which is known to be $\#\mathbf{P}$-hard. This will establish Theorem 5.6.

The proof proceeds as follows. For two given directed graphs $D_1, D_2$, we will show how to construct a digraph $D^\ell_{1,2}$ such that, for some polynomially large $\ell \in \mathbf{N}$,

$$C_{\text{geo}}(D^\ell_{1,2}; 0, y) \approx C_{\text{geo}}(D_1; 0, y) + C_{\text{geo}}(D_2; 0, y).$$

Given such a construction, we will show how, for $z \in \mathbf{Q}$, we can build a digraph $D_z$ such that $C_{\text{geo}}(D_z; 0, y) \approx z$. Both these building blocks will enable us to perform a binary search over a precomputed interval of possible values of $C_{\text{geo}}(D; 0, y)$ for any directed graph $D$. □

Let us now turn to our first goal. We construct $D^\ell_{1,2}$ by first taking the disjoint union of $D_1$ and $D_2$. We want to connect both subgraphs such that ideally the cycle covers of $D^\ell_{1,2}$ are in one-to-one correspondence with the cycle covers of $D_1$ and $D_2$. For that purpose, we connect both subgraphs in such a way that (in a weighted sense) almost all cycle covers of $D^\ell_{1,2}$ consist of cycle covers of $D_1$ and one designated cycle cover on $D_2$, or vice versa, consist of cycle covers of $D_2$ and one designated cycle cover on $D_1$. Thus, the cycle covers of

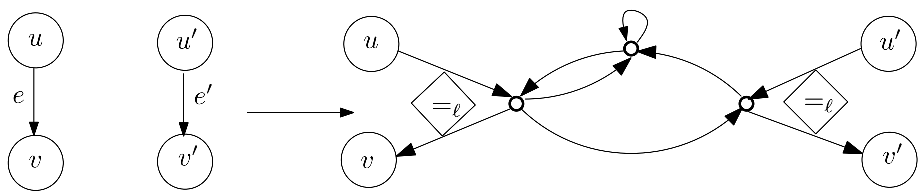



Figure 5.7: The implies gadget $Imp_\ell$.

$D^\ell_{1,2}$ can be seen as disjoint unions of the cycle covers of $D_1$ and $D_2$. It follows that $C_{\mathrm{geo}}(D^\ell_{1,2}; 0, y)$ is (almost) the sum of $C_{\mathrm{geo}}(D_1; 0, y)$ and $C_{\mathrm{geo}}(D_2; 0, y)$. The disjoint union is achieved by adding a Hamiltonian cycle $H_1$ to $D_1$ and connecting its edges by equality gadgets, and adding a Hamiltonian cycle $H_2$ to $D_2$ and also connecting its edges by equality edges. Finally, each original edge of $D_1$ gets connected with an arbitrary edge $e_0$ of the new Hamiltonian cycle $H_2$ by an 'implies gadget'. Such a gadget enforces that, (in a weighted sense) in almost all cycle covers, whenever an edge in $D_1$ is taken (other than from $H_1$), $e_0$ is also taken, and in turn, because of the equality gadgets, $H_2$ is used to cover $D_2$. Vice versa, the construction makes sure that, (in a weighted sense) in almost all cycle covers, if an edge in $D_2$ is taken (other than from $H_2$), no edge in $H_2$ can be taken simultaneously and thus $D_1$ must be covered by $H_1$. As equality gadgets, we use the same gadgets as in the previous proof, $E_\ell$ and $I_\ell$, for the cases $y \in (-1, 0)$ and $y < -1$, respectively. Summarizing the construction, we have,

1. Start with $D^\ell_{1,2} = D_1 \uplus D_2$ to be the disjoint union of $D_1$ and $D_2$. By a slight abuse of notation, we will now denote by $D_1$ and $D_2$ the corresponding subgraphs of $D^\ell_{1,2}$.

2. Add two arbitrary Hamiltonian cycles $H_1$ and $H_2$ to $D_1$ and $D_2$, respectively, consisting of newly introduced edges. Note that this might result in multiple edges.

3. Connect every pair of consecutive edges in $H_1$ and $H_2$ by equality gadgets $E_\ell$ (resp. $I_\ell$), see Figure 4.4.

4. Connect each edge in $D_1$ but not in $H_1$ with a designated edge of $H_2$ by an implies gadget $Imp_\ell$.

The implies gadget is drawn in Figure 5.7. It is based upon two equality gadgets. We can easily see that $Imp_\ell$ has the desired behavior, that is, when

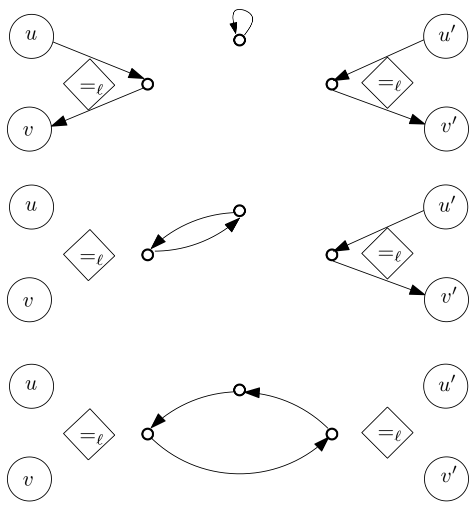



Figure 5.8: Case analysis of approximate implies gadget $Imp_\ell$. Bad states resulting from bad states of equality gadgets are omitted.

its left path $u \rightsquigarrow v$ is present in a cycle cover, then the right path $u' \rightsquigarrow v'$ does also have to be present (apart from the negligible bad states due to bad states of the equality gadgets), see Figure 5.8. Notice that all good states of the gadget add exactly one additional cycle to the corresponding cycle cover.

The following lemma assures that our construction works sufficiently well.

Lemma 5.7. *Let $D_1$, $D_2$ be two digraphs, $y \in \mathbf{Q}$, and $r \in \mathbf{N}$ an error parameter. There exists $k, \ell \leq poly(|D_1|, |D_2|, r)$ such that, for $D^\ell_{1,2}$ as constructed above, it holds that*

$$C_{geo}(D^\ell_{1,2}; 0, y) = y^k\big(C_{geo}(D_1; 0, y) + C_{geo}(D_2; 0, y) + \epsilon\big),$$

*where $|\epsilon| < 2^{-r}$. Moreover, $k$ and $\ell$ are computable in polynomial time.*

Proof. Let $\ell \in \mathbf{N}$ be a positive integer that will be specified later. Let $n_1 := n(D_1)$, $n_2 := n(D_2)$, $m_1 := m(D_1)$, $m_2 := m(D_2)$. First note that our construction uses $n_1$ equality gadgets for $H_1$, $n_2$ equality gadgets for $H_2$ and $2m_1$ equality gadgets for the $m_1$ implies gadgets. Furthermore, every original edge of $D_1$ and $H_1$ is split into at most four and three edges in $D^\ell_{1,2}$, respectively. Likewise, every original edge of $H_2$ is split into into at most three edges except for the designated edge of $H_2$ which is split into $4m_1 + 2$ edges. Since the cycle cover of every gadget is uniquely determined by its outer edges, we thus have in total at most $2^{4(2m_1+n_1+n_2)+2}$ cycle covers of $D^\ell_{1,2}$. We will now consider the two cases, $y \in (-1,0)$ and $y < -1$, separately. In each case, we determine the weight contribution of the good cycle covers of $D^\ell_{1,2}$, and bound the absolute weight contribution of the bad cycle covers of $D^\ell_{1,2}$.

**Case $\boldsymbol{y \in (-1,0)}$:** In the good cycle covers, defined as before, the weight contribution of the equality gadgets $E_\ell$ (with even $\ell$) is $y^{n_1+n_2+2m_1}$, since each equality gadget adds one cycle. Besides, each implies gadget also adds one cycle, incurring a factor of $y^{m_1}$. Finally, each good cycle cover of $D^\ell_{1,2}$ is a cycle cover of $D_1$ or $D_2$ with an additional Hamiltonian cycle on the other subgraph, respectively. In other words, there is a bijection between the cycle covers of $D^\ell_{1,2}$ and the union of the cycle covers of $D_1$ and $D_2$ which almost preserves the number of cycles. Hence, we get,

$$\begin{aligned}\sum_{\substack{\text{good}\\ \text{cycle cover } C}} y^{\sigma(C)} &= y^{n_1+n_2+2m_1} \cdot y^{m_1} \cdot \big(yC_{\text{geo}}(D_1;0,y) + yC_{\text{geo}}(D_2;0,y)\big)\\ &= y^{n_1+n_2+3m_1+1} \cdot \big(C_{\text{geo}}(D_1;0,y) + C_{\text{geo}}(D_2;0,y)\big).\end{aligned}$$

In the bad cycle covers, at least one equality gadget is in a bad case, adding $\ell+1$ cycles. It follows,

$$\begin{aligned}\Bigg|\sum_{\substack{\text{bad}\\ \text{cycle cover } C}} y^{\sigma(C)}\Bigg| &\le \left|y^{n_1+n_2+2m_1-1} \cdot y^{\ell+1}\right| \cdot 2^{4(2m_1+n_1+n_2)+2}\\ &= \left|y^{n_1+n_2+3m_1+1} \cdot y^{\ell-m_1-1}\right| \cdot 2^{4(2m_1+n_1+n_2)+2}.\end{aligned}$$

In total, we get

$$\begin{aligned}C_{\text{geo}}(D^\ell_{1,2};0,y) &= \sum_{\substack{\text{good}\\ \text{cycle cover } C}} y^{\sigma(C)} + \sum_{\substack{\text{bad}\\ \text{cycle cover } C}} y^{\sigma(C)}\\ &= y^{n_1+n_2+3m_1+1} \cdot \big(C_{\text{geo}}(D_1;0,y) + C_{\text{geo}}(D_2;0,y) + \epsilon\big),\end{aligned}$$

where $|\epsilon| \le |y|^{\ell-m_1-1} \cdot 2^{4(2m_1+n_1+n_2)+2}$.

It is now easy to choose an even $\ell \in \mathcal{O}(m_1+n_1+n_2+r)$ such that $|\epsilon| < 2^{-r}$.

**Case $y < -1$:** Noting that each equality gadget $I_\ell$ adds $\frac{\ell+3}{2}$ cycles in the good cases (for odd $\ell$), we get for the good cycle covers by the same line of reasoning as in the previous case,

$$\sum_{\substack{\text{good}\\ \text{cycle cover } C}} y^{\sigma(C)} = y^{(n_1+n_2+2m_1)\frac{\ell+3}{2}} \cdot y^{m_1} \cdot (yC_{\text{geo}}(D_1;0,y) + yC_{\text{geo}}(D_2;0,y))$$

$$= y^{(n_1+n_2+2m_1)\frac{\ell+3}{2}+m_1+1} \cdot \big(C_{\text{geo}}(D_1;0,y) + C_{\text{geo}}(D_2;0,y)\big).$$

In the bad cycle covers, at least one equality gadget is in a bad case, adding at most one cycle. Apart from that, each implies gadget adds at most one cycle. It follows,

$$\Bigg|\sum_{\substack{\text{bad}\\ \text{cycle cover } C}} y^{\sigma(C)}\Bigg| \leq \left|y^{(n_1+n_2+2m_1-1)\frac{\ell+3}{2}} \cdot y \cdot y^{m_1}\right| \cdot 2^{4(2m_1+n_1+n_2)+2}$$

$$= \left|y^{(n_1+n_2+2m_1)\frac{\ell+3}{2}+m_1+1} \cdot y^{-\frac{\ell+3}{2}}\right| \cdot 2^{4(2m_1+n_1+n_2)+2}.$$

In total, we have

$$C_{\text{geo}}(D^{\ell}_{1,2};0,y) = \sum_{\substack{\text{good}\\ \text{cycle cover } C}} y^{\sigma(C)} + \sum_{\substack{\text{bad}\\ \text{cycle cover } C}} y^{\sigma(C)}$$

$$= y^{(n_1+n_2+2m_1)\frac{\ell+3}{2}+m_1+1} \cdot \big(C_{\text{geo}}(D_1;0,y) + C_{\text{geo}}(D_2;0,y) + \epsilon\big),$$

where $|\epsilon| \leq |y|^{-\frac{\ell+3}{2}}\, 2^{4(2m_1+n_1+n_2)+2}$.

Again, for some odd $\ell \in \mathcal{O}(m_1+n_1+n_2+r)$, we have $|\epsilon| < 2^{-r}$. □

In the following, we denote by $(D_1 + D_2)(r)$ the graph $D^{\ell}_{1,2}$ with error parameter $r$, as constructed above.

We will now turn to our second building block of the proof of Theorem 5.6. Our goal is to construct a directed graph $D_z$ such that $C_{\text{geo}}(D_z;0,y) \approx z$, where $z \in \mathbf{Q}$.

In case $y$ is integer and z a non-integer value, we cannot hope to build a digraph $D_z$ with $C_{\text{geo}}(D_z,0,y) \approx z$. Instead, we will relax our original goal and show how to construct a digraph $D_z$ such that $C_{\text{geo}}(D_z;0,y) = y^c \cdot z$ with $c$ being an easily computable value. This relaxation is also necessary for a generic construction of $D_z$ for non-integer $y$.

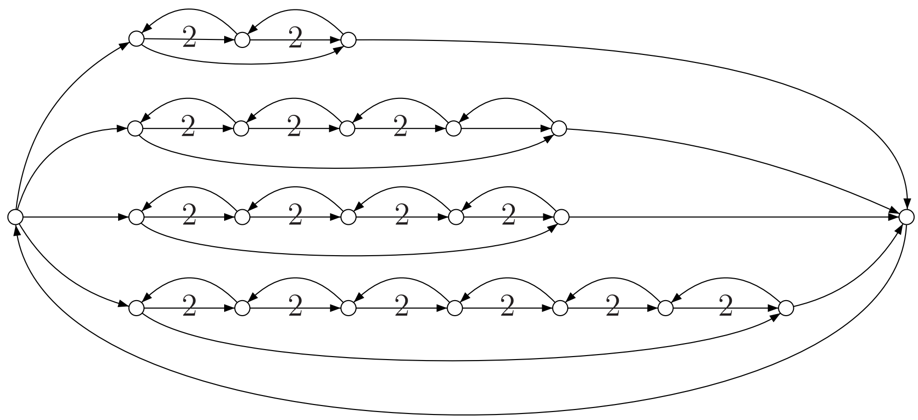



Figure 5.9: A digraph $D_{92}$ with $92 = 2^2 + 2^3 + 2^4 + 2^6$ cycle covers. The edge labels denote multiplicities.

As a first step in this direction, we construct a digraph $D_k$ such that $D_k$ has exactly $k \in \mathbf{N}$ cycle covers, each of which consists of the same number of cycles.

LEMMA 5.8. *Let $y \in \mathbf{Q}$ and $k \in \mathbf{N}$ be a non-negative integer. There exists a digraph $D_k$ such that*

$$C_{geo}(D_k; 0, y) = y^c \cdot k,$$

*for some positive integer $c \leq \log_2 k$. In particular, $D_k$ and $c$ can be computed in time logarithmic in $k$.*

PROOF. Consider the unique binary expansion $k = \sum_{i=0}^{\lfloor \log_2 k \rfloor} a_i 2^i$. Let $c$ be the number of non-zero coordinates $a_i$ and let $\big(I(1), \ldots, I(c)\big)$ be their corresponding indices. Figure 5.9 illustrates how the graph $D_k$ is built that contains exactly $k$ cycle covers, each of which consists of $c$ cycles. Notice that the $i$-th upper path contains a path of length $I(i)$ whose edges have multiplicity 2. In cases, where $I(i)$ is odd, the path is extended by one more edge. Finally, each path gets embedded in a separate cycle of opposite direction. It is easy to check that every cycle cover of $D_k$ consists of one outer cycle going through one of the $c$ pathes and $c-1$ inner cycles. In particular, the path chosen for the outer cycle uniquely determines the cycle cover of $D_k$. Furthermore, since the $i$-th path can be traversed in $2^{I(i)}$ different ways, $D_k$ has exactly $k$ cycle covers. □

For $y \in \mathbf{Z}^-$, the above construction yields the following corollary.

Corollary 5.9. *Let $y \in \mathbf{Z}^-$ and $z \in \mathbf{Z}$. There exists a digraph $D_z$, such that*

$$C_{\text{geo}}(D_z; 0, y) = y^c \cdot z,$$

*for some positive integer $c \in \mathcal{O}(\log |z|)$. In particular, $D_z$ and $c$ can be computed in time logarithmic in $|z|$.*

Proof. If $z \geq 0$, the claim directly follows from Lemma 5.8. Otherwise, apply Lemma 5.8 to get a graph $D_{yz}$ such that $C_{\text{geo}}(D_{yz}; 0, y) = y^c \cdot yz = y^{c+1} \cdot z$. □

For $y \in \mathbf{Q}^- \setminus \mathbf{Z}$, we can extend the above construction to any $z \in \mathbf{Q}$.

Lemma 5.10. *Let $y \in \mathbf{Q}^- \setminus \mathbf{Z}$, $z \in \mathbf{Q}$ be a rational value, and $\ell \in \mathbf{N}$ be an error parameter. There exists a digraph $D_z$, such that*

$$C_{\text{geo}}(D_z; 0, y) = y^c(z + \epsilon),$$

*where $c \leq \mathcal{O}(\ell + \log |z|)$ and $2^{-\ell} > \epsilon \geq 0$. In particular, $D_z^\ell$ can be build in time polynomial in $\ell$ and $\log z$.*

Proof.
Again, we distinguish the cases $y \in (-1, 0)$ and $y < -1$.

**Case $y \in (-1, 0)$:** Assume for the moment that $z \geq 0$. Our goal is to construct a digraph $D_z$ such that $C_{\text{geo}}(D_z; 0, y) = y^c y^k \cdot \lceil z \cdot y^{-k} \rceil$, where $c \in \mathbf{Z}^+$, and $k \in \mathbf{N}$ is an even integer to be specified later. Applying Lemma 5.8, we first construct a graph $D_z$ with exactly $\lceil z \cdot y^{-k} \rceil$ cycle covers. Let $c \in \mathcal{O}(k + \log z)$ be the number of cycles of each of these cycle covers. Then, we add $k$ disjoint loops to $D_z$. Thus, we have $C_{\text{geo}}(D_z; 0, y) = y^c y^k \cdot \lceil z \cdot y^{-k} \rceil$. By choosing $k \in \mathcal{O}(\ell)$ such that $y^k < 2^{-\ell}$, the claim follows.

If $z < 0$, we basically use the same construction, but choose $k$ to be odd and build $D_z$ such that it has $\lfloor z \cdot y^{-k} \rfloor$ cycle covers.

**Case $y < -1$:** We use a similar construction as in case $y \in (-1, 0)$. For that purpose, we construct a generic digraph $D_r$ such that $C_{\text{geo}}(D_r; 0, y) \in (-1, 0)$. This graph $D_r$ will take the role of the independent loops used in the previous case.

Let $y \in (-i, -i + 1)$ for some $i \in \mathbf{Z}^+$. Clearly, we have $y + i \in (-1, 0)$. Unfortunately, there exists no digraph whose cover polynomial is $y + i$. In particular, every cover polynomial has an absolute term of zero. We can fix

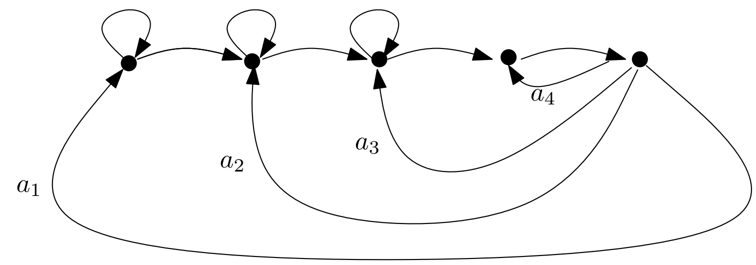



Figure 5.10: A digraph $D_r$ with $C_{\text{geo}}(D_r; 0, y) = a_1 \cdot y + a_2 \cdot y^2 + a_3 \cdot y^3 + a_4 \cdot y^4$. The edge labels denote multiplicities.

this problem by considering the polynomial $y(y+i)$. While it is easy to find digraphs having $y(y+i)$ as their cover polynomial, it is not the case that $y(y+i) \in (-1, 0)$. However, we can choose $j$ logarithmic in $y$ such that $y(y+i)^j \in (-1, 0)$. Set $r := y(y+i)^j$ and let $r = \sum_{i=1}^{j+1} a_i y^i$ be the polynomial expansion of $r$. Note that for fixed $y$, $j$ and hence $r$ are constant. In particular, all coefficients $a_i$ of the expansion are constants. We build a digraph $D_r$ with $C_{\text{geo}}(D_r; 0, y) = \sum_{i=1}^{j+1} a_i y^i$ as illustrated in Figure 5.10 for $j = 3$.

Assume that $z \geq 0$. Similar to the previous case, we now build a graph $D_z$ with exactly $\lceil z \cdot r^{-k} \rceil$ cycle covers, each of which contains $c \in \mathcal{O}(k + \log z)$ cycles, for some even $k \in \mathbf{N}$. Adding $k$ disjoint graphs $D_r$ as constructed above, we get $C_{\text{geo}}(D_z; 0, y) = y^c r^k \cdot \lceil z \cdot r^{-k} \rceil$, since $C_{\text{geo}}(x, y)$ is multiplicative (see Lemma 2.8). By choosing $k \in \mathcal{O}(\ell)$ such that $r^k < 2^{-\ell}$, the claim follows.

As before, for $z < 0$, we just choose $k$ to be odd and build $D_z$ accordingly such that $C_{\text{geo}}(D_z; 0, y) = y^c r^k \cdot \lfloor z \cdot r^{-k} \rfloor$.

□

We are now ready to prove Theorem 5.6.

Proof (Proof of Theorem 5.6). Let $y = \frac{p}{q}$, with $p \in \mathbf{Z}^-$ and $q \in \mathbf{N}$ relatively prime.

For any directed graph $D = (V, E)$, we have $|C_{\text{geo}}(D; 0, y)| \leq (|y| + 1)^n \cdot 2^m$. In other words,

$$C_{\text{geo}}(D; 0, y) \in [-(|y| + 1)^n \cdot 2^m, (|y| + 1)^n \cdot 2^m].$$

We want to perform a binary search over this interval in order to determine $C_{\text{geo}}(D; 0, y)$ exactly. Since $C_{\text{geo}}(D; 0, y)$ is an integer multiple of $\frac{1}{q^n}$, it suffices to consider all $\frac{c}{q^n}$ for $c \in \mathbf{Z}$ that lie in this interval. For that purpose, we will show how, for a given pivot element $z = \frac{c}{q^n}$, we can determine whether $z$ is less, equal, or greater than $C_{\text{geo}}(D;0,y)$. If $z=0$, this distinction is equivalent to determine the sign of $C_{\text{geo}}(D;0,y)$ which we can easily do having an FPRAS at hand. Otherwise, let $D_{-z}$ be a graph as constructed in Corollary 5.9 or Lemma 5.10 (depending on whether $y$ is integer or non-integer, respectively), such that

$$C_{\text{geo}}(D_{-z};0,y) = y^c(-z+\epsilon), \tag{5.11}$$

where $c \in \mathbf{N}$ is a polynomially computable integer and

$$0 \le \epsilon \le \tfrac{1}{4}q^{-n}. \tag{5.12}$$

If $y$ is integer we have $\epsilon = 0$ by Corollary 5.9. If $y$ is not integer, the last inequality holds by Lemma 5.10 for $\ell \in \mathcal{O}(n)$ such that $2^{-\ell} < \frac{1}{4}q^{-n}$. In any case, we have $c \in \mathcal{O}(\ell + \log z) = \mathcal{O}(m+n)$.

Let $D'$ be a copy of $D$ with $c$ additional independent disjoint loops. Thus,

$$C_{\text{geo}}(D';0,y) = y^c C_{\text{geo}}(D;0,y). \tag{5.13}$$

W.l.o.g. assume that $c$ is even. Otherwise, we add one more independent loop to $D_{-z}$ and $D'$, respectively.

We consider the 'sum' of $D'$ and $D_{-z}$ with error parameter $r \in \mathcal{O}(m+n)$ such that

$$2^{-r} < \tfrac{1}{4}y^c q^{-n}. \tag{5.14}$$

We have,

$$\begin{aligned} C_{\text{geo}}\big((D'+D_{-z})(r);0,y\big) &= y^k\big(C_{\text{geo}}(D';0,y) \\ &\qquad + C_{\text{geo}}(D_{-z};0,y) + \epsilon'\big) \quad \text{by Lemma 5.7} \\ &= y^k\big(y^c C_{\text{geo}}(D;0,y) \\ &\qquad + y^c(-z+\epsilon) + \epsilon'\big) \qquad \text{by (5.13) and (5.11)} \end{aligned}$$

for some polynomially computable $k \in \mathbf{N}$ and by (5.14),

$$|\epsilon'| < \tfrac{1}{4}y^c q^{-n}. \tag{5.15}$$

Equivalently,

$$y^{-k}\, C_{\text{geo}}((D'+D_{-z})(r);0,y) = y^c\big(C_{\text{geo}}(D;0,y) - z + \epsilon\big) + \epsilon'. \tag{5.16}$$

Now consider the three cases:

**Case $z \geq C_{\mathrm{geo}}(D; 0, y) + q^{-n}$:**

$$\begin{aligned} y^{-k}\, C_{\mathrm{geo}}\big((D' + D_{-z})(r); 0, y\big) &\leq y^c(-q^{-n} + \epsilon) + \epsilon' && \text{by (5.16)} \\ &< y^c(-\tfrac{3}{4}q^{-n}) + \tfrac{1}{4}y^c q^{-n} && \text{by (5.12) and (5.15)} \\ &= -\tfrac{1}{2}y^c q^{-n}. \end{aligned}$$

**Case $z \leq C_{\mathrm{geo}}(D; 0, y) - q^{-n}$:**

$$\begin{aligned} y^{-k}\, C_{\mathrm{geo}}\big((D' + D_{-z})(r); 0, y\big) &\geq y^c(q^{-n} + \epsilon) + \epsilon' && \text{by (5.16)} \\ &\geq y^c q^{-n} - \tfrac{1}{4}y^c q^{-n} && \text{by (5.12) and (5.15)} \\ &= \tfrac{3}{4}y^c q^{-n}. \end{aligned}$$

**Case $z = C_{\mathrm{geo}}(D; 0, y)$:**

$$\begin{aligned} y^{-k}\, C_{\mathrm{geo}}\big((D' + D_{-z})(r); 0, y\big) &= y^c\epsilon + \epsilon' && \text{by (5.16)} \\ &\leq \tfrac{1}{2}y^c q^{-n} && \text{by (5.12) and (5.15)} \end{aligned}$$

and

$$\begin{aligned} y^{-k}\, C_{\mathrm{geo}}\big((D' + D_{-z})(r); 0, y\big) &= y^c\epsilon + \epsilon' && \text{by (5.16)} \\ &\geq -\tfrac{1}{4}y^c q^{-n} && \text{by (5.12) and (5.15).} \end{aligned}$$

Note that this case analysis is exhaustive, since $z$ is an integer multiple of $q^{-n}$.

Clearly, an FPRAS can distinguish between the three cases with high probability. There are at most $2(|y| + 1)^n \cdot 2^m \cdot q^n$ multiples of $q^{-n}$ in the search interval $[-(|y|+1)^n \cdot 2^m, (|y|+1)^n \cdot 2^m]$. Hence, the binary search takes at most $\mathcal{O}(m + n)$ bisections until $C_{\mathrm{geo}}(D; 0, y)$ is determined. So, given an FPRAS for $C_{\mathrm{geo}}(0, y)$, we can evaluate $C_{\mathrm{geo}}(0, y)$ exactly with high probability, which would entail **RFP** = #**P**. □

**5.2. Inapproximability at Points with Graphs that Evaluate to Zero.** In this section, we consider points $(x, y) \in \mathbf{Q}^2$ for which there exists a graph $D$ such that $C_{\mathrm{geo}}(D; x, y) = 0$. We call such graphs *roots*. We will show how to reduce $C_{\mathrm{geo}}(0, y)$ to $C_{\mathrm{geo}}(x, y)$ whenever $C_{\mathrm{geo}}(x, y)$ has roots. Since this reduction will be approximation preserving, it follows that we can extend the inapproximability results of the $y$-axis to all these points.

Specifically, we get the following theorem.

Theorem 5.17. *Let* $(x, y) \in \mathbf{Q}^2$ *with* $y \notin \{1, 0, -1\}$ *such that there is a root in* $(x, y)$*. If* $y > 0$ *or* $y < 0$*, then approximating* $C_{\text{geo}}(x, y)$ *is not possible within any polynomial factor unless* $\mathbf{RP} = \mathbf{NP}$ *or* $\mathbf{RFP} = \#\mathbf{P}$*, respectively.*

The above theorem directly follows from the following main lemma of this section.

Lemma 5.18. *Let* $(x, y) \in \mathbf{Q}^2$*. If there is a root in* $(x, y)$*, then we have*

$$C_{\text{geo}}(0, y) \leq_{\text{AP}} C_{\text{geo}}(x, y).$$

Before we prove Lemma 5.18, we need the notion of a *minimal* graph.

Let $D = (V, E)$ be a digraph and $(x, y) \in \mathbf{Q}^2$. We call $D$ *edge-minimal* in $(x, y)$, if there is no nonempty subset $E' \subseteq E$, such that

$$C_{\text{geo}}(D - E'; x, y) = C_{\text{geo}}(D; x, y)$$

where $D - E' = (V, E \setminus E')$. We call $D$ *node-minimal* in $(x, y)$, if there is no nonempty subset $V' \subseteq V$ such that

$$C_{\text{geo}}\big(D - V'; x, y\big) = C_{\text{geo}}(D; x, y)$$

where $D - V'$ is the subgraph of $D$ induced by $V \setminus V'$. Finally, we call a graph *minimal*, if it is both edge-minimal and node-minimal.

Fact 5.19. *Let* $(x, y) \in \mathbf{Q}^2$*. If there is a root in* $(x, y)$*, then there is also a minimal root in* $(x, y)$*.*

We are now ready to prove Lemma 5.18.

Proof (Proof of Lemma 5.18). Let $R$ be a minimal root in $(x, y)$ and $D = (V, E)$ be a digraph. Our goal is to compute $C_{\text{geo}}(D; 0, y)$ from $C_{\text{geo}}(D; x, y)$ using an approximation preserving reduction. For that purpose, we connect each vertex in $D$ with a suitable gadget such that the weight contribution of the true path-cycle covers (i.e., those that have at least one path) of $D$ to $C_{\text{geo}}(D; x, y)$ becomes zero, whereas the weight contribution of each *cycle* cover of $D$ remains basically untouched. It turns out that the following two conditions are sufficient for such a gadget $G$:

1. $G$ contains a vertex $s$ with no incoming edges.
2. $G$ is a root, but $G - s$ is not. In particular, we have $\alpha := C_{\text{geo}}(G - s; x, y) \neq 0$.

To see why these conditions are sufficient for our purpose, consider the following construction.

1. Start with $D$.
2. Connect each vertex $v \in V$ with a (separate) copy of $G$, by identifying $v$ with the vertex $s$ of $G$. For each vertex $v \in V$, we denote its gadget by $G_v$.
3. Denote the resulting graph by $D'$.

What is the value of $C_{\text{geo}}(D'; x, y)$? For each path-cycle cover $\mathcal{PC}$ of $D$, let $D'(\mathcal{PC})$ denote the set of all path-cycle covers of $D'$ that agree with $\mathcal{PC}$ on $D$, i.e., yield $\mathcal{PC}$ when restricted to $D$. Assume $C$ is a cycle cover. Then, because of the multiplicity of the cover polynomial (see Lemma 2.8), the total weight contribution of path-cycle covers from $D'(C)$ to $C_{\text{geo}}(D'; x, y)$ is $y^{\sigma(C)} \cdot \alpha^n$ where $\alpha = C_{\text{geo}}(G - s; x, y)$ as defined above. On the other hand, let $\mathcal{PC}$ be a path-cycle cover containing a path that ends in some $v \in V$. We can extend this path by any path-cycle cover on the gadget $G_v$ because $G_v$ contains no incoming edges into $v$. Now, since $C_{\text{geo}}(G_v; x, y) = 0$ and by multiplicity of the cover polynomial (see Lemma 2.8), the total weight contribution of the path-cycle covers from $D'(\mathcal{PC})$ to $C_{\text{geo}}(D'; x, y)$ is zero. Note that $\{D'(\mathcal{PC}) \mid \text{path-cycle cover } \mathcal{PC} \text{ of } D\}$ is a partition of all path-cycle covers of $D'$.

Hence, we get

$$
\begin{aligned}
C_{\text{geo}}(D'; x, y) &= \sum_{\text{path-cycle cover } \mathcal{PC} \text{ of } D} x^{\rho(\mathcal{PC})} y^{\sigma(\mathcal{PC})} \alpha^{n-\rho(C)} 0^{\rho(C)} \\
&= \sum_{\text{cycle cover } C \text{ of } D} y^{\sigma(C)} \alpha^n \\
&= \alpha^n C_{\text{geo}}(D; 0, y).
\end{aligned}
$$

It remains to show the existence of such a gadget that satisfies the two above mentioned conditions given a minimal root $R$. Note that $R$ must have edges, since otherwise $C_{\text{geo}}(R; x, y)$ cannot be zero for $x \neq 0$. Consider the graph $G(R)$ depicted in Figure 5.11. Here, the subgraph $R$ is connected with the rest of $G(R)$ through any of its vertices $v$ that has incoming edges.

We will prove that, for a particular choice of the multiplicities $p, q \in \mathbf{N}$, $G(R)$ fulfills both conditions. Denote by $R'$ the digraph that is obtained from $R$ when all incoming edges to $v$ are deleted. By edge-minimality of $R$, we have

$$
\delta := C_{\text{geo}}(R'; x, y) \neq 0. \tag{5.20}
$$

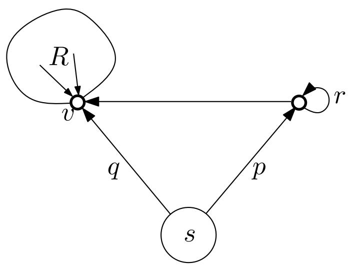



Figure 5.11: The gadget $G(R)$ containing a minimal root $R$ with incoming edges at all nodes. The edge labels denote multiplicities.

We now prove that $G(R)$ is a root for a particular choice of $p$ and $q$. Applying the contraction-deletion identities, we get

$$C_{\text{geo}}(G(R); x, y) = q\delta(x + ry) + \delta x + p\delta. \tag{5.21}$$

It is easy to see that for

$$x = -\frac{p + qry}{q + 1}, \tag{5.22}$$

we get $C_{\text{geo}}(G(R); x, y) = 0$.

Let $x = \frac{a}{b}$ with $a \in \mathbf{Z}$ and $b \in \mathbf{N} \setminus \{0\}$ relatively prime.

Assume that $a \neq 0$, as otherwise $x = 0$ and we are already at the $y$-axis. If $a < 0$, we set $r := 0$, $p := -a$ and $q := b - 1$. If $a > 0$, there can only be a root in $(x, y)$, if $y < 0$. In that case, we set $q := 2b - 1 > 0$, $r := \lceil \frac{2a}{|y|} \rceil$ and $p := -2a - qry$. Note that because of our choice of $r$, it holds $p > 0$.

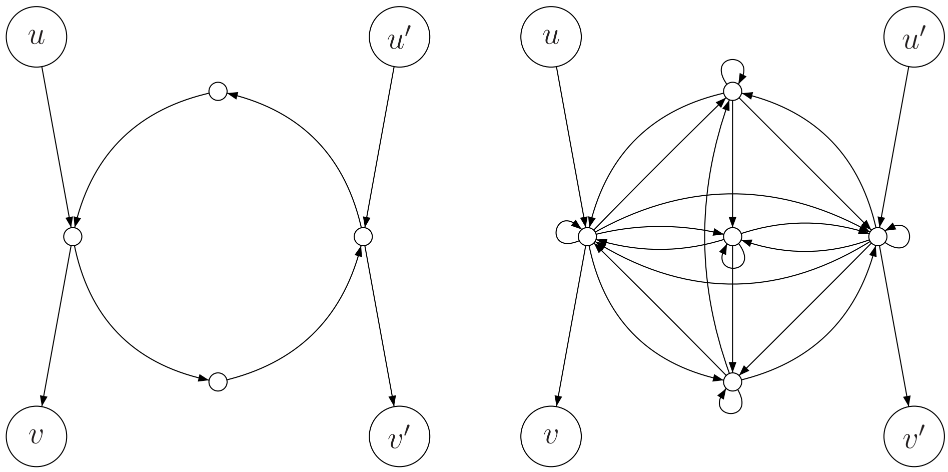
It is easy to check that, with this choice, Equation (5.22) is satisfied. Furthermore, $p$, $q$, and $r$ are constant for fixed $x$ and $y$.

It remains to check the second condition, namely that $G(R) - s$ is not a root. By (5.20) and since $R$ is a root, we get

$$C_{\text{geo}}(G(R) - s; x, y) = \delta \neq 0. \tag{5.23}$$

□

Remark 5.24. *Many families of points have roots, e.g., every point $(c \cdot x, -x)$ for all $c \in \mathbb{N}$ and $x \in \mathbb{Q}$ has a root. On the other hand, not every point $(x, y) \in \mathbf{Q}^2$ has a root. Consider, e.g., the point $(-\frac{1}{2}, 0)$. Let $D$ be a digraph.*




(a) (b)

Figure 5.12: Main components of equality gadgets. (a) is used for $C_{\text{geo}}(-1, 0)$. (b) is used for $C_{\text{geo}}(-1, -1)$.

*Note that the weight of each path-cycle cover of $D$ is a multiple of $2^{-n}$. Since only the path-cycle cover that consists of $n$ paths (independent vertices) has weight exactly $2^{-n}$ and all other path-cycle covers have either weight zero or a weight larger than $2^{-n}$, it follows $C_{\text{geo}}(D; -\frac{1}{2}, 0) \neq 0$.*

**5.3. Extending to $y = 0$ and $y = -1$.** Unfortunately, the above inapproximability results do not directly carry over to the two lines $y = 0$ and $y = -1$ since the corresponding problems on the $y$-axis are computable in polynomial time. Thus, we will establish the inapproximability of one point on each of these two lines and give an approximation preserving reduction from these two points to every point on the lines that has a root. Specifically, we consider the problems $C_{\text{geo}}(-1, 0)$ and $C_{\text{geo}}(-1, -1)$. The proof mainly follows the proof of the inapproximability of the negative $y$-axis. So we will only give a rough proof sketch.

LEMMA 5.25. *For $y \in \{0, -1\}$, approximating $C_{\text{geo}}(-1, y)$ is not possible within any polynomial factor unless* **RFP** $= \#$**P**.

PROOF (sketch). The essential difference to the proof for the $y$-axis lies in the equality gadgets. The main components of the gadgets are depicted in Figure 5.12. On the one hand, they ensure that the total weight contribution

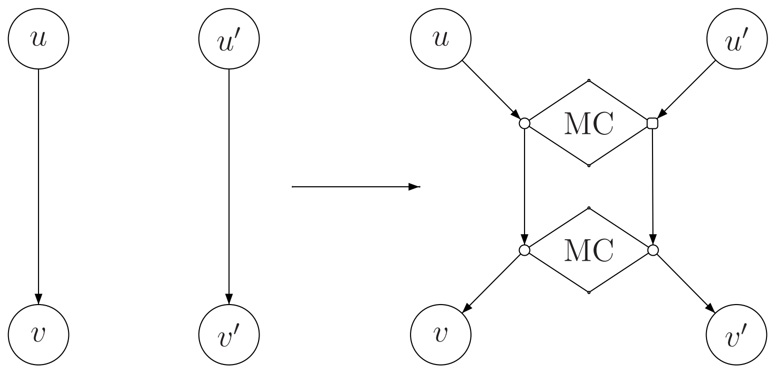



Figure 5.13: Equality gadget construction. Use graphs of Figure 5.12 in place of diamond.

of all good cycle covers in which either both outer paths $u \rightsquigarrow v$ and $u' \rightsquigarrow v'$ are present or both are absent have a non-zero weight contribution in total. On the other hand, even though these graphs do not prohibit bad states completely, all the other bad cycle covers sum up to zero. The reason why we still cannot directly use these graphs as equality gadgets lies in the fact that the weight factor stemming from all good states of the graphs where both outer paths are present differs from the one stemming from all good states where both outer paths are absent - but luckily only in sign. To overcome this problem we connect two copies of these graphs in series to form the desired equality gadgets, see Figure 5.13. Having an equality gadget, we can proceed to build an implies gadget as well as a NAND gadget as done before, see Figure 5.7 and Figure 5.4. It is easy to adapt the inapproximability proof of the negative $y$-axis using these newly constructed gadgets. □

We can now carry over this inapproximability result to all points on the lines $y = 0$ and $y = -1$ that have roots.

Lemma 5.26. *Let* $y \in \{0, -1\}$ *and* $x \neq 0$*. If there is a root in* $(x, y)$*, then* $C_{geo}(x, y)$ *is hard to approximate within any polynomial factor unless* **RFP** $=$ #**P**.

Proof. We give an approximation preserving reduction similar to the one used in Section 5.2. For that, we construct a digraph $G$ with the following properties:

1. $G$ contains a vertex $s$ with no incoming edges.

2. $C_{\text{geo}}(G;x,y) = -C_{\text{geo}}(G-s;x,y)$ with $\alpha := C_{\text{geo}}(G;x,y) \neq 0$.

Using $G$ in the same construction as in Section Section 5.2 and by the same reasoning, we get a digraph $D'$ with

$$\begin{aligned} C_{\text{geo}}(D';x,y) &= \sum_{\text{path-cycle cover } \mathcal{PC} \text{ of } D} \alpha^{\rho(\mathcal{PC})} y^{\sigma(\mathcal{PC})} (-\alpha)^{n-\rho(\mathcal{PC})} \\ &= \alpha^n \sum_{\text{path-cycle cover } \mathcal{PC} \text{ of } D} (-1)^{\rho(\mathcal{PC})} y^{\sigma(\mathcal{PC})} \\ &= \alpha^n C_{\text{geo}}(D;-1,y). \end{aligned}$$

Given a minimal root $R$, we construct a digraph $G(R)$ as shown in Figure 5.11. As before, denote by $R'$ the graph that is obtained from $R$ when all incoming edges to $v$ are deleted. By edge-minimality of $R$, we have

$$\delta := C_{\text{geo}}(R';x,y) \neq 0. \tag{5.27}$$

We now prove $C_{\text{geo}}(G(R);x,y) = -C_{\text{geo}}(G(R)-s;x,y) \neq 0$ for a particular choice of the multiplicities $p, q \in \mathbf{N}$. By (5.21) we have,

$$C_{\text{geo}}(G(R);x,y) = q\delta(x+ry) + \delta x + p\delta. \tag{5.28}$$

Let $x = \frac{a}{b}$ with $a \in \mathbf{Z} \setminus \{0\}$ and $b \in \mathbf{N} \setminus \{0\}$ relatively prime.

If $y = 0$, there is no root in $(x,y)$ for $x > 0$. It follows $a < 0$. By choosing $r := 0$, $q := b-1$ and $p := -a+1$, we get $C_{\text{geo}}(G(R);x,y) = -\delta$. If $y = -1$, let $q := 2b-1 > 0$, $r := \lceil \frac{2|a|}{q} \rceil$ and $p := qr - 2a - 1$. Note that, because of our choice of $r$, it holds $p > 0$ and $C_{\text{geo}}(G(R);x,y) = -\delta$.

As for $G(R) - s$, we get by (5.23),

$$C_{\text{geo}}(G(R)-s;x,y) \neq 0. \qquad \square$$

## 6. Inapproximability of the Factorial Cover Polynomial

Although the cover polynomial agrees with the geometric cover polynomial on the $y$-axis and satisfies the same contraction-deletion identities (2.7), except for the base case, it seems much more difficult to handle than its geometric version. The reason is that while the geometric cover polynomial is multiplicative (see Lemma 2.8), we only have a rudimentary product rule for the cover polynomial.

Lemma 6.1 (Chung & Graham 1995, Corollary 2). *Let $D_{1,2}$ be the graph obtained as the disjoint union of two digraphs $D_1 = (V_1, E_1)$ and $D_2 = (V_2, E_2)$ together with all edges $(v_1, v_2)$ for $v_1 \in V_1$ and $v_2 \in V_2$.*[1] *Then for all $x, y \in \mathbf{Q}$, we have*

$$C(D_{1,2}, x, y) = C(D_1, x, y)C(D_2, x, y).$$

Interestingly, there is a nice relationship between the cover polynomial of a simple digraph $D = (V, E)$ and its complement $\overline{D} = (V, V \times V \setminus E)$.

Lemma 6.2 (Chung & Graham 1995, Chow 1996). *Let $D$ be a simple digraph and $\overline{D}$ its complement. It holds,*

$$C(\overline{D}; x, y) = (-1)^n C(D; -x - y, y). \tag{6.3}$$

The lack of proper multiplicativity makes the task of finding gadgets for approximation preserving reductions harder than for the geometric cover polynomial. Still, since $C$ agrees with the geometric cover polynomial on the $y$-axis, we get the following corollary.

Corollary 6.4. *Let $y \in \mathbf{Q}$ with $y \notin \{1, 0, -1\}$.*

*For $y > 0$ or $y < 0$ approximating $C(0, y)$ is not possible within any polynomial factor unless* **RP** = **NP** *or* **RFP** = #**P**, *respectively.*

Note that this result also holds even if $C(0, y)$ is restricted to simple graphs, since all reductions used either do not make any use of graphs with multiple edges or can be easily adjusted appropriately. We can extend it a bit by using the 'horizontal reduction' of (4.4) and the reciprocity formula of Lemma 6.2.

Corollary 6.5. *Let $y \in \mathbf{Q} \setminus \{1, 0, -1\}$ and $c \in \mathbf{Z}$. For $y > 0$ or $y < 0$, $C(c-y, y)$ and $C(c, y)$ is not approximable within any polynomial factor, unless* **RP** = **NP** *or* **RFP** = #**P**, *respectively.*

Proof. Let $r \in \mathbf{N}$. If $C$ is restricted to simple graphs, we have

$$\begin{aligned} C(0, y) &\leq_{\mathrm{AP}} C(r, y) && \text{by (4.4)} \\ &\leq_{\mathrm{AP}} C(-r - y, y) && \text{by Lemma 6.2,} \end{aligned}$$

and furthermore,

[1] This is a kind of directed join.

$$
\begin{aligned}
C(0,y) &\leq_{\mathrm{AP}} C(-y,y) && \text{by Lemma 6.2}\\
&\leq_{\mathrm{AP}} C(r-y,y) && \text{by (4.4)}\\
&\leq_{\mathrm{AP}} C(-r,y) && \text{by Lemma 6.2.}
\end{aligned}
$$

By setting $r := c$ if $c \geq 0$ and $r := -c$ if $c < 0$, the corollary follows. □

## 7. Conclusion and Further Work

In this paper, we completely characterized the complexity of evaluating the cover polynomial and its geometric version in the rational plane. Our reductions should also work for complex or algebraic numbers, but we do not have any interpretation for such points yet.

We also gave a succinct characterization of a large class of points at which approximating the geometric cover polynomial within any polynomial factor is not possible. Furthermore, we extended this result to a grid of points for the cover polynomial.

Further work is to be done until a full classification of inapproximable regions is achieved. Although we conjecture all points apart from the positive quadrant to be non-approximable, it is not clear at all whether such a result can be established by a reduction from the $y$-axis. As for the positive quadrant, the question of whether there exists, in general, an FPRAS seems wide open. Specifically, it is unclear whether the FPRAS for the line $y = 1$ (with $x \geq 0$) also works for the whole positive quadrant. We tend to conjecture that apart from the line $y = 1$, only the line $y = 0$ is also approximable in the positive quadrant. As a step in this direction, one could try to prove that any FPRAS similar to the one used for the line $y = 1$ is doomed to fail elsewhere in this quadrant, in line with such a result for the independent set counting problem by Dyer *et al.* (2002).

One might conjecture that the connection between the Tutte and the cover polynomial is solved because both are #**P**-hard to evaluate at most points, so they can be computed from each other (where they are in #**P**). From a combinatorial point of view, however, **P**-reductions are not really satisfying steps towards a deeper understanding of the connection between the two graph polynomials.

One promising approach might be to consider generalizations of the cover polynomial. Very recently, Courcelle (unpublished) analyzed general contraction-deletion identities analogous to those of the coloured Tutte polynomial, and he found conditions for the variables which are necessary and sufficient for the well-definedness of the polynomial. Unfortunately, these conditions seem too restrictive.

The contraction-deletion identities of Tutte and cover polynomial look astonishingly similar, but contractions work differently for directed and for undirected graphs. Explaining the connection between Tutte and cover polynomial means overcoming this difference.

## Acknowledgements

We would like to thank Thomas Jansen and Moritz Hardt for fruitful discussions and valuable comments on earlier versions of this paper.

Research of the second author supported by the Deutsche Forschungsgemeinschaft within the research training group Methods for Discrete Structures (GRK 1408).

## References

Artur Andrzejak (1998). An algorithm for the Tutte polynomials of graphs of bounded treewidth. *Discrete Mathematics* **190**(1-3), 39–54.

Ilia Averbouch, Benny Godlin & Johann A. Makowsky (2008). A Most General Edge Elimination Polynomial. In *Graph-Theoretic Concepts in Computer Science, 34th International Workshop, WG 2008, Durham, UK, June 30 - July 2, 2008. Revised Papers*, volume 5344 of *Lecture Notes in Computer Science*, 31–42.

Andreas Björklund, Thore Husfeldt, Petteri Kaski & Mikko Koivisto (2008). Computing the Tutte polynomial in vertex-exponential time.

Markus Bläser, Holger Dell & Johann A. Makowsky (2008). Complexity of the Bollobás-Riordan Polynomial. In *3rd International Computer Science Symposium in Russia (CSR)*, 86–98.

Markus Bläser & Christian Hoffmann (2008). On the Complexity of the Interlace Polynomial. In *25th International Symposium on Theoretical Aspects of Computer Science (STACS 2008)*, Susanne Albers & Pascal Weil, editors, 97–108. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany. URL `http://drops.dagstuhl.de/opus/volltexte/2008/1337`.

Markus Bläser & Christian Hoffmann (2009). Fast Evaluation of Interlace Polynomials on Graphs of Bounded Treewidth. In *Algorithms - ESA 2009, 17th Annual European Symposium*, volume 5757 of *Lecture Notes in Computer Science*, 623–634. Springer.

Lenore Blum, Felipe Cucker, Michael Shub & Steve Smale (1998). *Complexity and real computation.* Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 0-387-98281-7.

Béla Bollobás & Oliver Riordan (1999). A Tutte Polynomial for Coloured Graphs. *Combinatorics, Probability and Computing* **8**(1-2), 45–93.

Andrei Bulatov & Martin Grohe (2005). The complexity of partition functions. *Theoretical Computer Science* **348**(2), 148–186.

Timothy Y. Chow (1996). A Short Proof of the Rook Reciprocity Theorem. *Electr. J. Comb.* **3**(1).

Fan R.K. Chung & Ronald L. Graham (1995). On the cover polynomial of a digraph. *Journal of Combinatorial Theory Series B* **65**(2), 273–290.

Bruno Courcelle, Johann A. Makowsky & Udi Rotics (2001). On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics* **108**(1-2), 23–52.

Ottavio M. D'Antona & Emanuele Munarini (2000). The cycle-path indicator polynomial of a digraph. *Adv. Appl. Math.* **25**(1), 41–56.

Martin Dyer, Alan Frieze & Mark Jerrum (2002). On Counting Independent Sets in Sparse Graphs. *SICOMP: SIAM Journal on Computing* **31**.

Martin E. Dyer, Alan M. Frieze & Mark Jerrum (1998). Approximately Counting Hamilton Paths and Cycles in Dense Graphs. *SIAM Journal on Computing* **27**(5), 1262–1272.

Martin E. Dyer, Leslie Ann Goldberg & Mike Paterson (2006). On Counting Homomorphisms to Directed Acyclic Graphs. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, ICALP 2006*, volume 4051 of *Lecture Notes in Computer Science*, 38–49. Springer.

Michael R. Garey & David S. Johnson (1979). *Computers and Intractability.* Freeman, San Francisco.

Omer Giménez & Marc Noy (2006). On the Complexity of Computing the Tutte Polynomial of Bicircular Matroids. *Combinatorics, Probability and Computing* **15**(3), 385–395.

Leslie Ann Goldberg & Mark Jerrum (2008). Inapproximability of the Tutte polynomial. *Inform. Comput.* **206**(7), 908–929.

Christian Hoffmann (2010). A Most General Edge Elimination Polynomial - Thickening of Edges. *Fundam. Inform.* **98**(4), 373–378.

François Jaeger, Dirk L. Vertigan & Dominic J.A. Welsh (1990). On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society* **108**(1), 35–53.

Mark Jerrum & Alistair Sinclair (1989). Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains. *Information and Computation* **82**(1), 93–133.

Mark Jerrum & Alistair Sinclair (1997). The Markov Chain Monte Carlo Method: An Approach to Approximate Counting and Integration. In *Approximation Algorithms for NP-hard Problems*, D. Hochbaum, editor. PWS Publishing Company.

Mark Jerrum, Alistair Sinclair & Eric Vigoda (2004). A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *JACM: Journal of the ACM* **51**.

Nathan Linial (1986). Hard enumeration problems in geometry and combinatorics. *SIAM J. Algebraic Discrete Methods* **7**(2), 331–335.

Martin Lotz & Johann A. Makowsky (2004). On the algebraic complexity of some families of coloured Tutte polynomials. *Advances in Applied Mathematics* **32**(1), 327–349.

Johann A. Makowsky (2008). From a zoo to a zoology: Towards a general theory of graph polynomials. *Theory of Computing Systems* **43**(3), 542–562.

S. D. Noble (1998). Evaluating the Tutte Polynomial for Graphs of Bounded Tree-Width. *Combinatorics, Probability & Computing* **7**(3), 307–321.

Christos M. Papadimitriou (1994). *Computational Complexity*. Addison-Wesley, Reading, Massachusetts.

Leslie G. Valiant (1979). The complexity of computing the permanent. *Theoretical Computer Science* **8**(2), 189–201.

Leslie G. Valiant & Vijay V. Vazirani (1986). NP is as easy as detecting unique solutions. *Theoretical Computer Science* **47**(1), 85–93.

Manuscript received

Markus Bläser
Department of Computer Science
Saarland University
Saarbrücken, Germany
mblaeser@cs.uni-saarland.de

Holger Dell
Department of Computer Science
Humboldt University of Berlin
Berlin, Germany
dell@informatik.hu-berlin.de

Mahmoud Fouz
Department of Computer Science
Saarland University
Saarbrücken, Germany
mfouz@cs.uni-saarland.de