The Complexity of Counting in Sparse, Regular, and Planar Graphs

Salil P. Vadhan*

May 9, 1997

Abstract

We show that a number of graph-theoretic counting problems remain $\mathcal{NP}$-hard, indeed $\#\mathcal{P}$-complete, in very restricted classes of graphs. In particular, it is shown that the problems of counting matchings, vertex covers, independent sets, and extremal variants of these all remain hard when restricted to planar bipartite graphs of bounded degree or regular graphs of constant degree. To achieve these results, a new interpolation-based reduction technique which preserves properties such as constant degree is introduced. In addition, the problem of *approximately* counting minimum cardinality vertex covers is shown to remain $\mathcal{NP}$-hard even when restricted to graphs of maximal degree 3. Previously, restricted-case complexity results for counting problems were elusive; we believe our techniques may help obtain similar results for many other counting problems.

1 Introduction

Ever since the introduction of $\mathcal{NP}$-completeness in the early 1970's, the primary focus of complexity theory has been on decision problems — that is, problems of determining whether an object (*e.g.* a graph) has a given property (*e.g.* has a Hamiltonian circuit). However, many problems of mathematical and practical interest are not decision problems. For a large fraction of these, computationally equivalent decision problems can be found. But it is possible that studying certain classes of problems separately might provide more insight into the nature of their difficulty than disguising them as decision problems would.

Counting problems form one class whose separate study has proven to be both natural and fruitful. From the time that Valiant [Val79a, Val79b] introduced the class $\#\mathcal{P}$ of counting problems and gave a complexity-theoretic

*Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139. Email: salil@math.mit.edu. Some of this work originally appeared in the author's undergraduate thesis [Vad95], done at Harvard University under the supervision of Leslie Valiant. The author is currently supported by a National Defense Science and Engineering Graduate Fellowship.

explanation for the apparent difficulty of enumeration, counting has held an important place in theoretical computer science. Although several researchers continued Valiant's work by adding to the list of $\#\mathcal{P}$-complete problems [PB83, DF89, BW90, Rot96, Gol93, BD97, MW92, Lin86, JVW90], our understanding of the complexity of counting still pales in comparison to our understanding of decision problems.

This is unfortunate, for counting, aside from being mathematically interesting, is closely related to important practical problems. For instance, reliability problems are often equivalent to counting problems. Computing the probability that a graph remains connected given a probability of failure on each edge is essentially equivalent to counting the number of ways that the edges could fail without losing connectivity. Another application of counting arises in Artificial Intelligence research. As explained by Roth [Rot96], the Bayesian approach to reasoning requires evaluating the probability that a formula in the propositional calculus is true, and this is tantamount to counting the number of satisfying assignments. Thus, understanding the types of propositional formulae for which counting satisfying assignments is feasible tells us the extent to which Bayesian reasoning might be useful. Graph-theoretic counting problems such as the ones we consider also appear often in statistical physics [Har67, LV97, JS93].

Perhaps the most significant deficency in our understanding of counting is that, in most cases, we do not know whether hard counting problems remain hard when additional restrictions are placed on the problem instances. A quick glance at Garey and Johnson's famous catalogue of $\mathcal{NP}$-complete problems [GJ79] reveals that the restricted-case complexity of most difficult decision problems is understood in detail. This information is useful, because a complexity-theoretic hardness result often leads us to ask whether the instances we are interested in possess special properties which make the problem tractable. Restricted-case complexity results tell us when such special properties do not make a problem any easier, closing the gap between what we can do and what we know we cannot.

One of the reasons restricted-case complexity results are so elusive for counting problems is that many of the reductions used employ 'blow-up' techniques, which destroy special properties of the original problem instance. This makes it difficult to deduce additional restricted-case results from known restricted-case results. For example, although Dagum and Luby [DL92] have shown that counting perfect matchings remains $\#\mathcal{P}$-complete when restricted to 3-regular bipartite graphs, the standard reduction from counting perfect matchings to counting matchings in [Val79b] blows up the degree of the graph and does not enable us to conclude that counting matchings remains difficult in either regular or bounded-degree graphs.

In this paper, we introduce a new reduction technique that yields restricted-case complexity results for many problems of interest. In particular, we show in Theorem 1 that counting matchings, vertex covers, and variants of these structures remains difficult in planar bipartite graphs of bounded degree and in regular graphs of constant degree. Our technique, like some of those in [Val79b], is based on polynomial interpolation. However, in contrast to the reductions in most earlier papers on the complexity of counting, our reductions preserve graph properties such as regularity and degree-boundedness. Moreover, some of our reductions appear to require an unbounded number of oracle calls, unlike those in [Val79b].

Roughly speaking, our technique involves using a sequence of simple gadgets in which the structures we are trying to count can be described by linear recurrence relations, such as the Fibonacci recurrence. Attaching these gadgets to the original graph results in an interpolation at a sequence of rapidly converging rational points which we can prove does not repeat using some fairly general techniques described in the appendix. Previous approaches often attempted to obtain interpolations at distinct integer points, which it is difficult to do without losing special properties of the original graph. In addition, our methods for proving that the interpolation points do not repeat does not require an exact closed form for the sequence, enabling us to perform reductions that were not previously possible.

When the search for efficient restricted-case counting algorithms fails, the natural alternative is to seek approximate counting algorithms. However, restricted-case complexity results for approximate counting are even harder to come by than ones for exact counting. In Theorem 2, we obtain such a result, as we show that counting minimum cardinality vertex covers is $\mathcal{NP}$-hard even in graphs of maximal degree 3. Perhaps this could be used as a starting point for achieving other such results.

2 Preliminaries

Nearly all of the counting problems we will be considering are in Valiant's class $\#\mathcal{P}$, and the remainder are closely related to $\#\mathcal{P}$. Following [JVV86], $\#\mathcal{P}$ can be defined in terms of p-relations.

Let Σ be a finite alphabet. A relation $R \subset \Sigma^* \times \Sigma^*$ is said to be a ***p*-relation** iff it is polynomially-balanced, *i.e.* there exists a polynomial p such that $\langle x, y\rangle \in R \Rightarrow |y| \leq p(|x|)$; and it can be 'checked quickly,' *i.e.* the language $L = \{\langle x, y\rangle \in R\}$ can be decided in polynomial time. The counting problem $\#R$ associated with R is: Given $x \in \Sigma^*$, output $|R(x)| = |\{y \in \Sigma^* : \langle x, y\rangle \in R\}|$. $\#\mathcal{P}$ is the class of all such counting problems.

In the above definition, think of x as an instance of a problem, such as a boolean formula F, and $R(x)$ as the set of solutions associated with x, such as the satisfying assignments to F. It is easy to see that $\mathcal{NP}$ consists exactly of problems of the form: Given x, decide whether $R(x)$ is nonempty. Thus $\#\mathcal{P}$ is the set of counting problems naturally associated with $\mathcal{NP}$ languages.

In contrast to $\mathcal{NP}$, it turns out that mapping reductions are not sufficient to describe the relative difficulty of counting problems. It is easy to construct counting problems which are obviously equivalent in difficulty, but for which there can be no one-to-one correspondence between solution sets. Hence, as in [Val79a], we consider a problem Π to be as hard as a problem Γ iff Γ can be solved by a polynomial-time Turing machine with a Π-oracle, and we denote this by $\Gamma \propto \Pi$. Such a reduction is known as a **(polynomial-time) Turing reduction** [GJ79], and this is the only form of reduction we will refer to in this paper. A problem Π is said to be $\#\mathcal{P}$-hard iff all problems in $\#\mathcal{P}$ reduce to it; if, in addition, $\Pi \in \#\mathcal{P}$ it is called **$\#\mathcal{P}$-complete**. Lastly, a problem is said to be $\#\mathcal{P}$-easy if it can be reduced to some problem in $\#\mathcal{P}$. Occasionally, we will be able to reduce one problem to another via a mapping of problem instances that preserves the number of solutions. Such a reduction is called a **parsimonious** reduction and these are important because they preserve inapproximability. [1]

Having defined all the complexity-theoretic notions we will need, we proceed to define the combinatorial objects we will be studying. Let $G = (V, E)$ be an undirected graph. The **degree** of G is the maximum number of edges incident to any vertex. A **vertex cover** in G is a subset S of V such that every edge in E has at least on endpoint in S. An **independent set** in G is a subset S of V such that no two vertices in S are connected by an edge in E. A **clique** in G is a subset S of V such that every two vertices in S are connected by an edge in E. It is well-known that cliques, vertex covers, and independent sets are intimitely related objects. Their relationship is formalized by the following proposition:

Proposition 2.1 *Let $G = (V, E)$ be an undirected graph and let $\overline{G} = (V, \overline{E})$ be its (edge-)complement. Let F be the monotone 2-CNF formula on variables V given by $F = \bigwedge_{(u,v)\in E}(u \vee v)$. For $S \subset V$, let $\chi_S : V \to \{0, 1\}$ be the assignment which maps $v \in V$ to 1 iff $v \in S$. Then the correspondence $S \leftrightarrow (V - S) \leftrightarrow (V - S) \leftrightarrow \chi_S$ establishes bijections between the vertex covers in G, the independent sets in G, the cliques in $\overline{G}$, and the satisfying assignments of F.*

Proof By the definitions. $\Box$

The above proposition will enable us to immediately deduce hardness results for all of the above problems given a hardness result for one of them. Therefore, we will concentrate on just the vertex cover problem.

There are several reasons for studying these problems. One is that many $\mathcal{NP}$-completeness results have come via reduction from decision versions of these problems, so it reasonable to guess that many restricted-case $\#\mathcal{P}$-completeness results could come via reduction from restricted versions of these counting problems. Another reason is that these problems are closely related to other important problems in computer science. For example, Roth [Rot96] has shown that counting the number of satisfying assignments in a propositional formula is interreducible with computing the 'degree of belief' in a Bayesian Belief Network, an important Artificial Intelligence problem. His reduction shows that if counting satisfying assignments is hard for restricted classes of formulae, then computing the degree of belief is hard even for restricted network topologies. He proceeds to show that for some restricted classes of propositional formulae, counting the number of satisfying assignments is hard. Our results show that

[1]For a more detailed discussion of $\#\mathcal{P}$, the reader is referred to [Vad95] or [Pap94, Chapter 18].

counting satisfying assignments in even more restricted classes of monotone 2-CNF formulae is hard. These results are stronger than any others known for counting satisfying assignments.

We will also study extremal variants of all of the above problems. A vertex cover S is said to be **minimal** iff no proper subset of S is a vertex cover. It is said to be of **minimum cardinality** iff there is no vertex cover with fewer vertices. Similarly, we speak of maximal and maximum cardinality independent sets or cliques.

By Proposition 2.1, it is easy to see that minimal vertex covers correspond to **minterms** of a monotone 2-CNF formula — that is, satisfying assignments for which changing any variable from true to false would no longer satisfy the formula. It is clear that the smallest DNF form for a monotone formula F is simply the disjunction of all minterms, writing an individual minterm M as the conjunction of the variables in M. Hence, restricted-case hardness results for counting minimal vertex covers immediately imply that determining the size of the minimal DNF form is hard even for restricted classes of CNF formulae.

It is clear that if the decision problem associated with a p-relation is $\mathcal{NP}$-complete, then the associated counting problem is also $\mathcal{NP}$-hard. Thus, complexity of counting results are only interesting when the related decision problem is easy. The first nontrivial result of this form, due to Valiant, involved another type of graph-theoretic structure, known as a perfect matching. A **matching** in an undirected graph $G = (V, E)$ is a set $M \subset E$ of edges, no two of which share an endpoint. A **perfect matching** is a matching M in which every vertex in V is the endpoint of an edge in M. Valiant's theorem [Val79a] states that counting perfect matchings in bipartite graphs is $\#\mathcal{P}$-complete. Matchings can be related to the other structures we mentioned via the following construction:

Let $G = (V, E)$ be an undirected graph. The **line graph of G** is the undirected graph $L(G) = (E, H)$, where $(e_1, e_2) \in H$ iff e_1 and e_2 share an endpoint in G.

Lemma 2.2 *Let $G = (V, E)$ be an undirected graph. Then $M \leftrightarrow E - M$ establishes a bijective correspondence between matchings in G and vertex covers in the line-graph of G. In particular the number if matchings in G equals the number of vertex covers in $L(G)$ and the number of maximum cardinality matchings in G equals the number of minimum cardinality vertex covers in G.*

Proof Notice that $M \subset E$ is a matching in G iff M is an independent set in $L(G)$. The relationship with vertex covers follows from Proposition 2.1. $\Box$

One of restricted classes of graphs that we will examine is the class of planar graphs. A graph is said to be **planar** iff there exists an embedding of the graph in the plane (where the vertices are points and the edges are curves connecting the points) in which no two edges intersect. The bijection of Proposition 2.1 suggests how to define planarity for CNF formulae. If F is a formula in conjunctive normal form, we define $G(F)$ to be the graph whose vertices are the variables of F, where two vertices are connected if they lie in a common clause. We call F **planar** iff $G(F)$ is planar.

For their simplicity, their relationships to other important problems, and their potential to serve as starting points for further results, the problems of counting matchings, vertex covers, and independent sets are important ones to consider. So motivated, we now proceed to classify the cases in which these problems are and are not tractable.

3 Statement of Results

Our main results are that the problems of counting vertex covers and matchings and variants of these remain hard even in very restricted cases. In particular, we show that the problems of counting matchings, maximal matchings, vertex covers, and minimum cardinality vertex covers all remain $\#\mathcal{P}$-complete when restricted to planar bipartite graphs whose degree is bounded by a small constant or to k-regular graphs for constant k. The same results hold for minimal vertex covers, except that there is no restriction on the degree in the regular graphs. As a corollary, we obtain analogous results for independent sets, cliques, and satisfying assignments to propositional formulae. Previously, it was only known that, except for matchings, these problems are $\#\mathcal{P}$-complete for unrestricted bipartite graphs [Val79b, PB83]. Counting matchings was known to be hard in both unrestricted bipartite graphs and unrestricted planar graphs [Val79b, Jer87, Jer90]. It may stand out that there is no hardness result for counting *perfect* matchings in *planar* graphs. However, efficient algorithms, due to Fisher [Fis61], Kasteleyn [Kas63], and Temperley and Fisher [TF61], are known for this problem, so a $\#\mathcal{P}$-completeness result would be very unexpected.

We also prove that counting minimum cardinality vertex covers within a ratio of $2^{n^{1-\epsilon}}$ remains $\mathcal{NP}$-hard even in graphs of degree 3. Other approximate counting problems that were previously known to be hard include counting cycles in an undirected graph [JVV86] and counting independent sets in a graph [Rot96, Sin93]. The main interest of our result is that it holds for such a restricted class of graphs.

We begin with a statement of results, postponing the proofs until the next two sections. Theorem 1 contains our exact counting results, and Theorem 2 contains our approximate counting result. Note that all of the problems listed become polynomial-time solvable when restricted to graphs of maximal degree 2: Such a graph has connected components which are cycles and chains and all of the objects we study multiply across connected components, so it suffices to count in cycles and chains. It is not hard to write down closed forms or recurrences which can be used to count any of these structures in chains and cycles. This is done explicitly in [Vad95].

Theorem 1 *The following problems are $\#\mathcal{P}$-complete:*

1. $\#k$-REGULAR BIPARTITE PERFECT MATCHINGS, ANY FIXED $k \geq 3$ [2]
   Input: A k-regular bipartite graph G.
   Output: The number of perfect matchings in G.

[2] This result is due to Dagum and Luby [DL92]. We include it for completeness.

2. #4Δ-Bipartite Matchings
   Input: A bipartite graph G of degree ≤ 4.
   Output: The number of matchings in G.
3. #6Δ-Planar Bipartite Matchings
   Input: A planar bipartite graph G of degree ≤ 6.
   Output: The number of matchings in G.
4. #k-Regular Matchings, any fixed $k \geq 5$
   Input: A k-regular graph G.
   Output: The number of matchings in G.
5. #5Δ-Bipartite Maximal Matchings
   Input: A bipartite graph G of degree ≤ 5.
   Output: The number of maximal matchings in G.
6. #7Δ-Planar Bipartite Maximal Matchings
   Input: A planar bipartite graph G of degree ≤ 7.
   Output: The number of maximal matchings in G.
7. #Planar Bipartite Vertex Cover
   Input: A planar bipartite graph G.
   Output: The number of vertex covers in G.
8. #3Δ-Planar Bipartite Minimum Cardinality Vertex Cover[3]
   Input: A planar bipartite graph G of degree ≤ 3.
   Output: The number of minimum cardinality vertex covers in G.
9. #k-Regular Minimum Cardinality Vertex Cover, any $k \geq 4$ [4]
   Input: A k-regular graph G.
   Output: The number of minimum cardinality vertex covers in G.
10. #4Δ-Planar Bipartite Vertex Cover
    Input: A planar bipartite graph G of degree ≤ 4.
    Output: The number of vertex covers in G.
11. #k-Regular Vertex Cover, any fixed $k \geq 5$
    Input: A k-regular graph G.
    Output: The number of vertex covers in G.
12. #5Δ-Planar Bipartite Minimal Vertex Cover
    Input: A planar bipartite graph G of degree ≤ 5.
    Output: The number of minimal vertex covers in G.
13. #Regular Minimal Vertex Cover

[3] In contrast to the other problems, it is not obvious that this problem or problem 8 is in $\#\mathcal{P}$, as the general problem of checking whether a vertex cover is of minimum cardinality is $\mathcal{NP}$-hard. However, using Hall's marriage theorem, it can be shown that the size of the minimum cardinality vertex cover equals the size of the maximum cardinality matching in *bipartite* graphs. The latter quantity can be found using any of the standard maximum cardinality matching algorithms. See [Pap94, Problem 9.5.25].

[4] Actually, this problem is not likely to be in $\#\mathcal{P}$, as testing whether a vertex cover is of minimum cardinality is $\mathcal{NP}$-hard [GJ79, Thm 3.3]. But we will reduce it to a $\#\mathcal{P}$ problem, proving that it is $\#\mathcal{P}$-easy. The failure of $\#\mathcal{P}$ to be closed under reductions and even simpler operations is discussed in [OTTW96].

Input: *A regular graph G.*

Output: *The number of minimal vertex covers in G.*

Corollary 1 *The following problems are $\#\mathcal{P}$-complete:*

1. $\#3\Delta$-PLANAR BIPARTITE MAXIMUM CARDINALITY INDEPENDENT SET
   Input: *A planar bipartite graph G of degree ≤ 3.*
   Output: *The number of maximum cardinality independent sets in G.*
2. $\#k$-REGULAR MAXIMUM CARDINALITY INDEPENDENT SET, ANY $k \geq 4$ [5]
   Input: *A k-regular graph G.*
   Output: *The number of maximum cardinality independent sets in G.*
3. $\#4\Delta$-PLANAR BIPARTITE INDEPENDENT SET
   Input: *A planar bipartite graph G of degree ≤ 4.*
   Output: *The number of independent sets in G.*
4. $\#k$-REGULAR INDEPENDENT SET, ANY FIXED $k \geq 5$
   Input: *A k-regular graph G.*
   Output: *The number of independent sets in G.*
5. $\#5\Delta$-PLANAR BIPARTITE MAXIMAL INDEPENDENT SET
   Input: *A planar bipartite graph G of degree ≤ 5.*
   Output: *The number of maximal independent sets in G.*
6. #REGULAR MAXIMAL INDEPENDENT SET
   Input: *A regular graph G.*
   Output: *The number of maximal independent sets in G.*
7. #REGULAR MAXIMUM CARDINALITY CLIQUE
   Input: *A regular graph G.*
   Output: *The number of maximum cardinality independent sets in G.*
8. #REGULAR CLIQUE
   Input: *A regular graph G.*
   Output: *The number of cliques in G.*
9. #REGULAR MAXIMAL CLIQUE
   Input: *A regular graph G.*
   Output: *The number of maximal cliques in G.*
10. $\#4\Delta$-PLANAR BIPARTITE MONOTONE 2SAT
    Input: *A planar monotone formula $F = (c_{1,1} \vee c_{1,2}) \wedge \cdots \wedge (c_{r,1} \vee c_{r,2})$ on variables $X \cup Y$, where $X \cap Y = \emptyset$; $c_{i,1} \in X$, $c_{i,2} \in Y$ for each i; and each variable appears at most 4 times.*
    Output: *The number of satisfying assignments to F.*
11. $\#k$-REGULAR MONOTONE 2SAT, ANY FIXED $k \geq 5$
    Input: *A monotone formula $F = (c_{1,1} \vee c_{1,2}) \wedge \cdots \wedge (c_{r,1} \vee c_{r,2})$ on variables X; $c_{i,j} \in X$ for each i, j; and each variable in X appears exactly k times.*
    Output: *The number of satisfying assignments to F.*

[5] Again, this problem and the analogous one for cliques are technically not in $\#\mathcal{P}$, but they are $\#\mathcal{P}$-easy.

12. #5Δ-PLANAR BIPARTITE MINTERMS

Input: *As in 10, except each variable appears at most 5 times.*

Output: *The number of minterms of F, i.e. the number of clauses in the smallest DNF formula equivalent to F.*

13. #REGULAR MINTERMS

Input: *A monotone formula $F = (c_{1,1} \vee c_{1,2}) \wedge \cdots \wedge (c_{r,1} \vee c_{r,2})$ on variables X; $c_{i,j} \in X$ for each i, j; and each variable in X appears the same number of times.*

Output: *As in 12.*

Theorem 2 *The following problems are $\mathcal{NP}$-hard for every $\epsilon > 0$:*

1. $2^{n^{1-\epsilon}}$-APPROX #3Δ-MINIMUM CARDINALITY VERTEX COVER

Input: *An undirected graph G of degree ≤ 3 on n nodes.*

Output: *An approximation to the number of minimum cardinality vertex covers in G within ratio $2^{n^{1-\epsilon}}$.*

2. $2^{n^{1-\epsilon}}$-APPROX #3Δ-MAXIMUM CARDINALITY INDEPENDENT SET

Input: *An undirected graph G of degree ≤ 3 on n nodes.*

Output: *An approximation to the number of maximum cardinality independent sets in G within ratio $2^{n^{1-\epsilon}}$.*

4 Proof of Theorem 1

In the course of these reductions, we will need several technical lemmas. For the sake of readability, the precise statements and proofs of most of these are deferred to the appendix. We state three facts here that we will use repeatedly. Let K be a finite extension of $\mathbb{Q}$. For any $z \in K$, let $||z||$ denote the number of bits needed to represent z. For a polynomial f in several variables over K, let $||f||$ be the number of bits needed to represent f, which is the sum of $||a||$ for the coefficients a of f. For a rational function $q = f/g$, let $||q|| = ||f|| + ||g||$.

Fact 4.1 *Let $F = K(y_1, y_2, \ldots, y_k)$ be the field of rational functions over K in k variables for some constant k. Let $f(x) = \sum_{i=0}^{d} a_i x^i$ be a polynomial with coefficients in F. If $(\alpha_0, \beta_0), \ldots, (\alpha_d, \beta_d)$ such that $f(\alpha_i) = \beta_i$ are known for distinct $\alpha_i \in F$, then the coefficients of f can be recovered in time polynomial in $\max_i ||\alpha_i||$, $\max_i ||\beta_i||$, and d.*

Fact 4.2 *Let $f(x, y) = \sum_{i+j \leq n} a_{ij} x^i y^j$ be a polynomial in two variables with coefficients in K. If for each of $n+1$ distinct $x_i \in K$, $n+1$ distinct $y_{ij} \in K$ along with the values $z_{ij} = f(x_i, y_{ij})$ are known, then the coefficients of f can be recovered in time polynomial in $\max_{ij} ||a_{ij}||, \max_i ||x_i||, \max_{ij} ||y_{ij}||$, and $\max_{ij} ||z_{ij}||$.*

Fact 4.3 *Let $f(x) = \sum_{i=0}^{d} a_i x^i$ be a polynomial with nonnegative integer coefficients. If (α, β) is known where $f(\alpha) = \beta$ and α is a rational number satisfying $\alpha \geq (a_i + 1)$ for each i, then the coefficients of f can be recovered in time polynomial in $||\alpha||$, $||\beta||$, and d.*

Proof (of Fact 4.1) Here we just use the Lagrange interpolation formula:

$$f(x) = \sum_{i=0}^{d} \beta_i \left(\frac{(x-\alpha_1)\cdots(x-\alpha_{i-1})(x-\alpha_{i+1})\cdots(x-\alpha_d)}{(\alpha_i-\alpha_1)\cdots(\alpha_i-\alpha_{i-1})(\alpha_i-\alpha_{i+1})\cdots(\alpha_i-\alpha_d)} \right)$$

This is a polynomial which agrees with f at $d+1$ distinct points, so it must be the same polynomial. By multiplying out and collecting terms, we can obtain the coefficients of f all at once, in polynomial time. $\Box$

Proof (of Fact 4.2) Define, for each $0 \le r \le n$, $g_r(y) = \sum_{i,j} a_{ij} x_r^i y^j$. For each r, we know the evaluation of $g_r(y)$ at the $n+1$ points $y_{r0}, \ldots, y_{rn}$, so we can recover the polynomials $g_r(y)$ by Fact 4.1. These are the evaluations of $f(x,y)$, considered as a polynomial in x with coefficients in $K(y)$, at the points $x_0, \ldots, x_n$. By Fact 4.1, we can recover the coefficients of f. $\Box$

Proof (of Fact 4.3) All we need to do is write β as a number in base α and the digits are our coefficients. In more detail, note that $\sum_{i=0}^{d-1} a_i \alpha^i \le \sum_{i=0}^{d-1} (\alpha-1)\alpha^i = \alpha^d - 1 < \alpha^d$, so $a_d = \lfloor \beta/\alpha^d \rfloor$, which we can obtain quickly by just integer multiplication and division. Now consider $f_1(x) = f(x) - a_d x^d$. We can repeat this process for f_1, obtaining the coefficients in sequence. $\Box$

Proof (of Theorem 1)
1. Proven by Dagum and Luby [DL92] using a direct parsimonious reduction from #Perfect Matchings.

2. #3-Regular Perfect Matchings $\propto$ #4Δ-Matchings
Let G be a 3-regular bipartite graph with vertices $v_1, \ldots, v_n$. For each $s = 0, \ldots, n$, construct G_s as follows: Add nodes $v_{i,j}$ to G for $1 \le i \le n$, $1 \le j \le s$. Add edges to create disjoint chains $v_i - v_{i,1} - v_{i,2} - \cdots - v_{i,s}$ of $s+1$ nodes. Note that G_s has degree at most 4, since G has degree at most 3, and that G_s remains bipartite.

Now, for every matching M in G, we can obtain matchings in G_s by adding some of the new edges to M. For each vertex v_i of G which is matched by M, $(v_i, v_{i,1})$ cannot be added to the matching, but we can choose an arbitrary matching for the chain $v_{i,1} - \cdots - v_{i,s}$, and there are x_s ways to do this, where x_s is the number of matchings in a chain of s nodes. For each vertex v_i which is not matched by M, we can add an arbitrary matching of the chain $v_i - v_{i,1} - \cdots - v_{i,s}$ to M, and there are x_{s+1} ways to do this. Additionally, every matching of G_s is obtained by extending a matching of G in this fashion, because the restriction of any matching in G_s must be a matching in G. Let A_i be the number of matchings in G in which exactly i vertices are matched. Then the number of matchings in G_s is exactly $\sum_{i=0}^{n} A_i (x_s)^i (x_{s+1})^{n-i}$. We can obtain these values for $s = 0, \ldots, n$ with $n+1$ oracle calls. As observed in Lemma A.4, the x_s's are simply the Fibonacci numbers, which we can compute. Dividing by $(x_{s+1})^n$, we obtain the evaluation of $f(x) = \sum_{i=0}^{n} A_i x^i$ at the points (x_s/x_{s+1}) for $s = 0, \ldots, n$. By Lemma A.1, we evaluate f at $n+1$ distinct points, and by Fact 4.1 we can efficiently recover the coefficients of f. A_n is the number of perfect matchings in G.

This is the simplest incarnation of our reduction technique: we use a sequence of gadgets (in this case, chains) which preserves some graph property (in this case, bounded-degree bipartiteness) and in which the number of matchings is easily computed (in this case, Fibonacci numbers) to obtain interpolations at points which we can prove do not repeat. To apply this to other problems and obtain results for other types of graphs, we will have to use different gadgets. Proving that the interpolation points do not repeat may require a bit more work, but the essential idea is the same.[6]

3. #6Δ-Planar Bipartite Matchings

The starting point for this proof is the work of Jerrum [Jer87, Jer90] which shows that counting matchings in planar graphs is $\#\mathcal{P}$-complete. As is, his reduction produces graphs that are neither bipartite nor of bounded degree. We show how an additional step added in the middle of his reduction can transform the graphs produced into bipartite ones. We then show how a reduction like the one in Reduction 2 can replace the final step of his reduction so that the degree does not blow up.

In the course of his reduction, Jerrum considers a weighted form of #Matchings: Let $G = (V, E)$ be a graph in which each vertex $v \in V$ is assigned a weight $w(v) \in \mathbb{C}$. If $M \subset E$ is a matching in G, then we let $\boldsymbol{C(M)}$ be the set of vertices in V which are covered by M, *i.e.* are endpoints of edges in M. Then the **weight** of M is $\mathbf{w(M)} = \prod_{v \notin C(M)} w(v)$, *i.e.* the product of the weights of all vertices *not* covered by M. The **weighted matching sum $\boldsymbol{W(G)}$** of G is $\sum_M w(M)$, where this sum is taken over all matchings in G. Thus if all vertices have weight 1, $W(G)$ is simply the number of matchings in G. We say that G has **$\boldsymbol{k}$ weights** if the number of distinct values other than 1 occurring as weights in G is at most k. The $\#k\lambda$-Weighted Matchings problem is the following:

$\#k\lambda$-Weighted Matchings
Input: A vertex-weighted graph G with k weights.
Output: $W(G)$.

As usual, we also consider variants of this problem for bipartite, planar, and bounded degree graphs. The prefix $d\Delta$ is used to restrict to graphs of degree at most d. Our reduction will proceed in three stages. The first will show that computing $W(G)$ is hard for planar graphs of bounded degree. The second will transform any graph into a bipartite graph without changing $W(G)$, losing planarity, or increasing the degree. The third will remove weights one by one without losing any of the graph properties, showing that counting matchings in

[6] It appears that the number of oracle calls in this reduction and some of our others cannot be reduced to a constant. This is in contrast to the reductions done in [Val79b]. There, Valiant asserts that all the reductions can be done with a single oracle call, because the arithmetic can be simulated by operations on the graph or formula in question. Here that does not appear to work, because the graph manipulations necessary all blow up the degree.

bipartite planar graphs of bounded degree is hard.

3a. #3-Regular Perfect Matchings $\propto$
#5Δ-2λ-Planar Weighted Matchings

Jerrum [Jer87, Jer90] gives a reduction from #Perfect Matchings to #2λ-Planar Weighted Matchings. We observe that his reduction produces a graph of degree 5 when applied to a graph of degree 3.

3b. #5Δ-2λ-Planar Weighted Matchings $\propto$
#5Δ-4λ-Planar Bipartite Weighted Matchings

Consider the weighted graph H with vertex set $\{v_1, v_2, v_3, v_4\}$, edges (v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4), and vertex weights $w(v_1) = w(v_4) = 1$, $w(v_2) = \zeta$, $w(v_3) = \zeta^2$, where $\zeta = e^{2\pi i/3}$ is a primitive cube root of unity. (See Figure 1.)

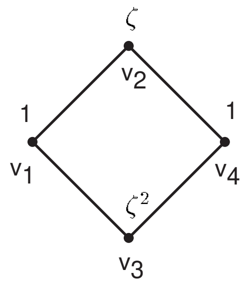

Figure 1: The graph H in Reduction 3b.

Straightforward calculations show the following:

1. $w(H) = w(H \setminus \{v_1, v_4\}) = 1$

2. $w(H \setminus \{v_1\}) = w(H \setminus \{v_4\}) = 0$

Above, the notation $H \setminus S$ denotes the graph formed by removing from H all vertices in S and any edges incident to them.

Now let G be a planar weighted graph with 2 weights and let $e = (u, v)$ be any edge in G. Consider the graph G' obtained from G by removing edge e; adding a disjoint copy of H; and adding edges $e_1 = (u, v_1)$ and $e_2 = (v, v_4)$. (See Figure 2.) We will show that $W(G') = W(G)$. Let M be any matching in G' that doesn't contain any of the edges of H. Then observation 2 above tells us that if M contains exactly one of e_1 and e_2, the net contribution to $W(G')$ of all matchings formed by adding H-edges to M will be zero. If M contains neither e_1 nor e_2, then, since $w(H) = 1$, the net contribution of all matchings formed by adding H-edges to M will be $\prod w(x)$ where the product is taken over all vertices $x \notin C(M) \cup \{v_1, v_2, v_3, v_4\}$. This is the same as the G-weight of

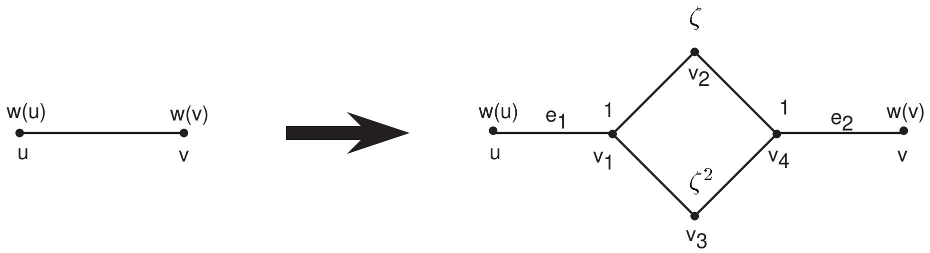

Figure 2: The transformation of Reduction 3b.

M. Similarly, since $w(H \setminus \{v_1, v_4\}) = 1$, if M contains both e_1 and e_2 the net contribution will be the same as the G-weight of $(M - \{e_1, e_2\}) \cup \{e\}$. Thus, the net weight of all matchings in G' accounts exactly for the net weight of all matchings in G and $W(G') = W(G)$. Therefore, if we do this procedure to all edges in G, we will end up with a planar bipartite graph $\hat{G}$ with 4 weights (ζ and ζ^2 are the only new weights) such that $W(\hat{G}) = W(G)$. It is clear that this reduction can be carried out in polynomial time.

In the next part of the reduction, we remove weights one by one, only increasing the degree by 1. We use the prefix $\boldsymbol{k\mu}$ to restrict to graphs with k weights in which all vertices with weight 1 have degree at most 6 and all other vertices have degree at most 5. In particular, an instance of #5Δ-4λ-Planar Bipartite Weighted Matchings satisfies this condition for $k = 4$.

3c. #$k\mu$-Planar Bipartite Weighted Matchings $\propto$
#$(k-1)\mu$-Planar Bipartite Weighted Matchings

Jerrum [Jer87, Jer90] gives a reduction which removes weights one by one, but his reduction blows up the degree. To replace his reduction, we use the same technique as in Reduction 2 — attaching chains in order to interpolate at ratios of Fibonacci numbers.

Let G be a bipartite planar graph with k weights, in which all vertices with weight 1 have degree at most 6 and all other vertices have degree at most 5. Let $\alpha \neq 1$ be a vertex weight that occurs in G and let $v_1, \ldots, v_m$ be the vertices with weight α. For each $s = 0, \ldots, m$, construct a graph G_s as follows: Add nodes $v_{i,j}$ to G for $1 \leq i \leq m$. Add edges to create disjoint chains $v_i - v_{i,1} - v_{i,2} - \cdots - v_{i,s}$ of $s+1$ nodes. Assign each of the new vertices weight 1, assign $v_1, \ldots, v_m$ weight 1, and keep all other vertex weights the same as in G. Notice that G_s is a valid instance of #$(k-1)\mu$-Planar Bipartite Weighted Matchings.

Now, for every matching M in G, we can obtain matchings in G_s by adding some of the new edges to M. Suppose M covers exactly i of $v_1, \ldots, v_m$. Then the net contribution to $W(G_s)$ of the matchings formed by adding new edges to M is $x_s^i (x_{s+1}/\alpha)^{m-i} w_G(M)$, where x_t is the number of matchings in a chain of t vertices. Let $A_i = \sum w_G(M)$, where the sum is taken over all matchings in G which match exactly i of $v_1, \ldots, v_m$. Then $W(G_s) = (x_{s+1}/\alpha)^m \sum_{i=0}^{m} A_i(\alpha x_s/x_{s+1})^i$. We can obtain these values for $s = 0, \ldots, m$ with $m+1$ oracle calls. As observed in Lemma A.4, the x_s's are simply the Fibonacci numbers, which we can compute. Dividing by $(x_{s+1}/\alpha)^m$, we obtain the evaluation of $f(x) = \sum_{i=0}^{m} A_i x^i$ at the points $(\alpha x_s/x_{s+1})$ for $s = 1, \ldots, m+1$. By Lemma A.1, we evaluate f at $m+1$ distinct points, and by Fact 4.1 we can efficiently recover the coefficients of f. Then it is easy to compute $W(G) = \sum_{i=0}^{m} A_i$.

It may seem odd that the graphs G_s constructed in this reduction do not depend on the value of α. This is because this reduction works even if α is regarded as an indeterminate — $W(G)$ is then a polynomial in α, and we are essentially recovering this polynomial.

Finally, note that $\#0\mu$-Planar Bipartite Weighted Matchings is exactly $\#6\Delta$-Planar Bipartite Matchings.

4. $\#(k-2)$-Regular Bipartite Perfect Matchings $\propto$ $\#k$-Regular Matchings, any fixed $k \geq 5$

Let H be the complete graph on $k+1$ vertices with one edge removed. Consider the graph L_n^k formed by taking $n+1$ disjoint copies of H, labelled $H_0, \ldots, H_n$, and attaching y_i to x_{i+1} for each $0 \leq i < n$, where x_i and y_i are the two vertices in H_i of degree $k-1$. (See Figure 3.)

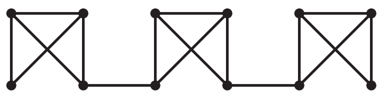

Figure 3: The graph L_2^3 in Reduction 4.

Now let G be any $(k-2)$-regular graph on n nodes $u_1, u_2, \ldots, u_n$. For any $0 \leq s \leq n$, consider the disjoint union of G with n copies of L_s^k. Let the vertices of degree $k-1$ in the ith copy of L_s^k be labelled v_i and w_i. We can form a k-regular graph G_s by connecting u_i to both v_i and w_i for each i. Let A_i be the number of matchings in G in which exactly i vertices are unmatched. Then the number of matchings in G_s is $\sum_{i=0}^{n} A_i(x_s + 4y_s + 3z_s)^i(x_s + 2y_s + z_s)^{n-i}$, where x_s is the number of matchings in L_s^k in which v and w — the two vertices of degree $k-1$ — are both matched, y_s is the number in which v but not w is matched, and z_s is the number in which neither v nor w is matched. By Lemma A.5, we can compute x_s, y_s, and z_s in polynomial time and the sequence $(y_s + z_s)/(x_s + 2y_s + z_s)$ never repeats. Thus, with $n+1$ oracle calls, we can evaluate the polynomial $f(x) = \sum_{i=0}^{n} A_i x^i$ at the $n+1$ distinct points $(x_s + 4y_s + 3z_s)/(x_s + 2y_s + z_s) = 1 + 2(y_s + z_s)/(x_s + 2y_s + z_s)$, $s = 0, \ldots, n$. By Fact 4.1, we can recover the coefficients of f in polynomial time. A_0 is the number of perfect matchings in G.

5. $\#4\Delta$-Matchings $\propto$ $\#5\Delta$-Maximal Matchings

The reduction we use is from [PB83], though they use it to prove something different. Given a bipartite undirected graph $G = (V, E)$ of degree ≤ 4, construct $G' = (V', E')$, where $V' = V \cup \{v' : v \in V\}$ and $E' = E \cup \{(v, v') : v \in V\}$. Note that every matching W in G yields a unique maximal matching $W' = W \cup \{(v, v') : v \text{ not matched by } W\}$ in G' and every maximal matching of G' can be obtained in this fashion. Observe that G' is bipartite and of degree ≤ 5.

6. #6Δ-Planar Bipartite Matchings $\propto$
#7Δ-Planar Bipartite Maximal Matchings.
Notice that Reduction 5 preserves planarity, so it applies here, too.

7. #Planar Bipartite Vertex Cover
This result will come via a sequence of reductions beginning with #Vertex Cover, whose #$\mathcal{P}$-completeness follows immediately from Lemma 2.2 and the #$\mathcal{P}$-completeness of #Matchings. In the spirit of Reduction 3, the intermediate problems involve a generalization of vertex covers which we call an **edge-weighted set**. Suppose G is a graph in which each edge $e = (u, v)$ is labelled with a triple $d_e/s_e/n_e$ and S is a set of vertices in G. Then define the **weight of S with respect to e** as:

$$w_e(S) = \begin{cases} d_e & \text{if } |\{u, v\} \cap S| = 2, \\ s_e & \text{if } |\{u, v\} \cap S| = 1, \\ n_e & \text{if } |\{u, v\} \cap S| = \emptyset, \end{cases}$$

so that d_e, s_e, and n_e correspond to weights if e is "doubly covered," "singly covered," or "not covered" by S, respectively. The **weight of S with respect to G** is then defined as $w_G(S) = \prod_e w_e(S)$. We now define the **edge-weighted sum $\mathbf{EW}(G)$** to be $\sum_{S \subset V} w_G(S)$. Notice that if all the edges in G are labelled $1/1/0$, then $\mathrm{EW}(G)$ is simply the number of vertex covers in G, so we have indeed generalized #Vertex Cover. With this in mind, we call an edge of such a labelled graph **normal** if its label is $1/1/0$. For technical reasons, we will restrict to graphs with only a constant number of distinct labels. We will say a graph labelled as above has **k labels** if the number of distinct labels other than $1/1/0$ is at most k.

Our first aim in reducing #Vertex Cover to planar graphs is to simplify the types of graphs we deal with. We call an embedding of a labelled graph G in the plane **simple** iff only normal edges are involved in crossings and each edge is in at most one crossing. Consider the following computational problem:

$k\lambda$-Simple Edge-Weighted Sum
Input: A labelled graph G with k labels and a simple embedding of G in the plane.
Output: $\mathrm{EW}(G)$.

We now reduce #Vertex Cover to this problem.

7a. #Vertex Cover $\propto$ #1λ-Simple Edge-Weighted Sum

Let m be the number of edges in $G = (V, E)$. For any $s \geq 2m$ and any $t \geq 0$ consider the graph $G_{s,t} = (E_{s,t}, V_{s,t})$ formed by removing each edge $e = (u, v)$ of G and replacing it with the $s + 2$ vertices u^e, v^e, and $w_1^e, \ldots, w_s^e$. Also add the edges (u, u^e), (v, v^e), (u, w_1^e), (v, w_s^e) and (w_i^e, w_{i+1}^e) for $i = 1, \ldots, s-1$. Furthermore, label the edges (u, u^e) and (v, v^e) with the label $(tf_{s-1}/f_s)/0/1$, where f_s is the sth Fibonacci number, as defined in Lemma A.1. Label all other edges $1/1/0$. Notice that it is easy to obtain a simple embedding of $G_{s,t}$ from any embedding of G in the plane, as we have broken each edge of G into $\geq 2m$ pieces and the edges of the form (u, u^e) clearly make no difference.

For each set of vertices S in G, let us consider the (weighted) number of ways that we may extend S to $G_{s,t}$, *i.e.* let us compute

$$\sum_{T : T \cap V = S} w_{G_{s,t}}(T).$$

If an edge $e = (u, v)$ of G is doubly covered by S, then there are $x_s(tf_{s-1}/f_s)^2$ (weighted) ways of adding vertices u^e, v^e, and $w_1^e, \ldots, w_s^e$ to S, where x_k is the number of vertex covers in a chain of k vertices. If e is singly covered by S, then there are $x_{s-1}tf_{s-1}/f_s$ ways to add these vertices. Finally, if e is not covered by S, there are x_{s-2} ways. Thus, if A_{ij} is the number of subsets of V which doubly cover i edges in G and which singly cover j edges, then

$$\mathrm{EW}(G_{s,t}) = \sum_{i,j} A_{ij}(x_s(tf_{s-1}/f_s)^2)^i (x_{s-1}tf_{s-1}/f_s)^j (x_{s-2})^{m-i-j}.$$

Writing r_k for f_k/f_{k-1} and using the fact that $x_k = f_{k+1}$ (from Lemma A.4), we get

$$\mathrm{EW}(G_{s,t}) = \sum_{i,j} A_{ij}((f_{s+1}/f_{s-1})(t/r_s)^2)^i t^j f_{s-1}^m.$$

Using the relation $r_{s+1} = 1 + 1/r_s$, we get $f_{s+1}/f_{s-1} = r_{s+1}r_s = r_s + 1$. Substituting this above, we get

$$\mathrm{EW}(G_{s,t}) = f_{s-1}^m \sum_{i,j} A_{ij}(t^2(r_s^{-1} + r_s^{-2}))^i t^j.$$

Lemma A.1 tells us that the sequence $\{r_s\}$ does not repeat. Since $x^{-1} + x^{-2} \neq y^{-1} + y^{-2}$ for any two distinct positive real numbers x and y, the sequence $\{r_s^{-1} + r_s^{-2}\}$ also does not repeat. Thus, by Fact 4.2, evaluating $\mathrm{EW}(G_{s,t})$ for each $t = 0, \ldots, m$ and for each $s = 2m, \ldots, 3m+1$ enables us to recover the coefficients A_{ij}. $\sum_{i+j=m} A_{ij}$ is the number of vertex covers of G.

We've reduced the problem to dealing with simple embeddings of graphs; the next step is to the problem of computing EW for planar graphs, as defined below:

$k\lambda$-Planar Edge-Weighted Sum

Input: A labelled graph G with k labels and a planar embedding of G.

Output: $\mathrm{EW}(G)$.

The aim of the next reduction will be to replace crossings with planar gadgets without changing the value of $EW(G)$.

7b. #1λ-Simple Edge-Weighted Sum $\propto$
#5λ-Planar Edge-Weighted Sum
First we make planar gadgets to compute elementary boolean formulae. We will write sets of vertices as functions from the set of all vertices to $\{0,1\}$, where $S(v)=1$ indicates that $v \in S$ and $S(v)=0$ indicates that $v \notin S$. Consider the AND gadget in Figure 4.

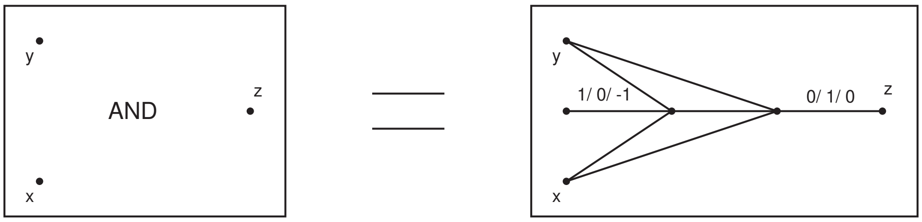

Figure 4: The AND gadget in Reduction 7b.

It is easy to see that for any $a, b, c \in \{0,1\}$,

$$\sum_{S:S(x)=a,S(y)=b,S(z)=c} w_{\mathrm{AND}}(S) = \begin{cases} 1 & \text{if } z = x \wedge y. \\ 0 & \text{otherwise.} \end{cases}$$

Thus, this gadget "forces" z to be $x \wedge y$. Similarly, the OR gadget of Figure 5 forces z to be $x \vee y$. Observe that an edge labelled $1/0/1$ forces its endpoints to

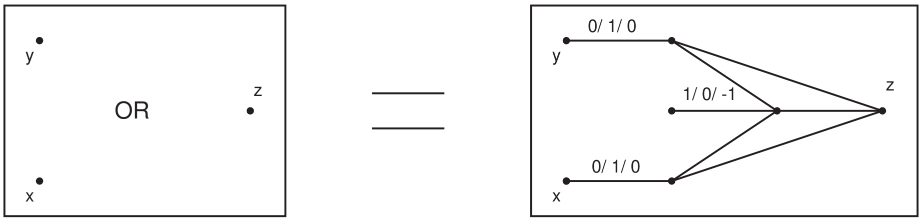

Figure 5: The OR gadget in Reduction 7b.

take on the same value and an edge labelled $1/0/0$ forces both its endpoints to take on the value 1. The AND and OR gadgets constructed above along with these observations, enable us to form a complex gadget to replace crossings: Take any simple embedding of $G=(V,E)$ in the plane. Consider any two (normal) edges $e_1=(a,c)$ and $e_2=(b,d)$ which cross, where a,b,c,d is the order of the endpoints going clockwise around the crossing starting with a. Let G' be the graph with these edges removed. For any $S \subset V$, $w_{G'}(S)=w_G(S)$ if S contains at least one endpoint of both e_1 and e_2 and $w_G(S)=0$ otherwise. The key observation is that S contains at least one endpoint of both e_1 and e_2

iff S contains some pair of vertices in $\{a, b, c, d\}$ which are *adjacent*, when these vertices are considered in clockwise order. Thus, if we replace the crossing with a gadget which simply forces $(a \wedge b) \vee (b \wedge c) \vee (c \wedge d) \vee (d \wedge a) = 1$, then the edge-weighted sum does not change. This can be done with the planar gadget of Figure 6.

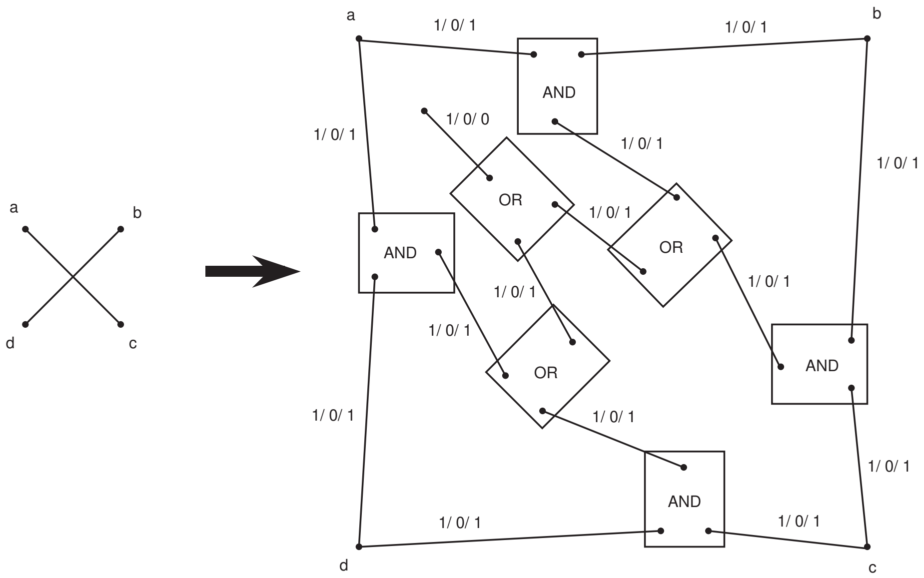

Figure 6: Replacing crossings in Reduction 7b.

If we replace all crossings of G in this manner, we obtain a planar labelled graph G' such that $\mathrm{EW}(G) = \mathrm{EW}(G')$. The gadgets only use the labels $1/0/0$, $0/1/0$, $1/0/{-1}$, and $1/0/1$ in addition to the labels already present in G, so G' uses 5 labels. Finally, notice that the transformation can be performed in polynomial time.

Now we show that we can reduce the number of labels one at a time until there are none, showing that planar vertex cover is $\#\mathcal{P}$-complete.

7c. $\#k\lambda$-Planar Edge-Weighted Sum $\propto$
$\#(k-1)\lambda$-Planar Edge-Weighted Sum
Let $G = (V, E)$ be any planar graph. Pick any label $\lambda = (a, b, c)$ used in G, let L be the set of edges with label λ and let $\ell = |L|$. Let $G_{s,t}$ be the graph formed by replacing all edges $e = (u, v)$ with label λ in the following fashion: Remove e, add vertices

$$V_e = \{w_1^e, \ldots, w_s^e, u_1^e, \ldots, u_t^e, v_1^e, \ldots, v_t^e\},$$

and add (normal) edges (u, w_i^e), (v, w_i^e), (u, u_j^e), (u_j^e, v_j^e), (v_j^e, v) for each $0 \leq i \leq s, 0 \leq j \leq t$.

Let S be a subset of V. For each edge $e \in L$ doubly covered by S, there are $2^s 3^t$ (weighted) ways to add the vertices of V_e to S. For each edge $e \in L$

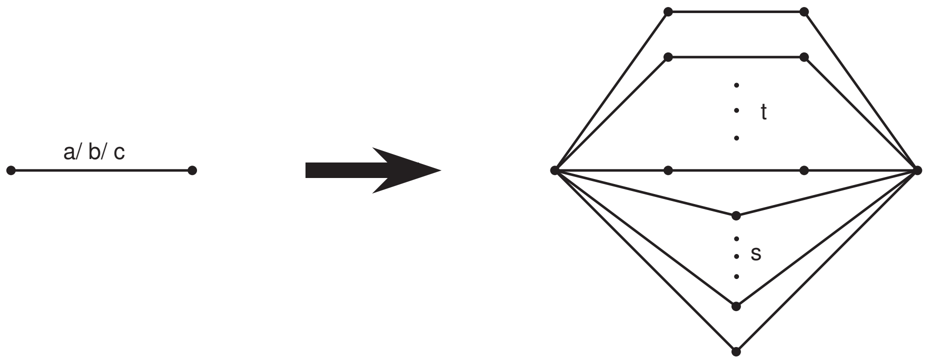

Figure 7: The transformation of Reduction 7c.

singly covered, there are 2^t ways. For each $e \in L$ not covered, there is only 1 way. Thus, if C_{ij} is the collection of subsets of V which doubly cover i edges of L and singly cover j edges of L, and

$$A_{ij} = \frac{1}{a^i b^j c^{\ell-i-j}} \sum_{S \in C_{ij}} w_G(S),$$

then

$$\mathrm{EW}(G_{s,t}) = \sum_{i,j} A_{ij} (2^s 3^t)^i (2^t)^j .$$

By Fact 4.2, if we can compute $\mathrm{EW}(G_{s,t})$ for $t = 0, \ldots, \ell$, $s = 0, \ldots, \ell$, we can recover the coefficients A_{ij} and thereby compute

$$\mathrm{EW}(G) = \sum_{i,j} A_{ij} a^i b^j c^{\ell-i-j} .$$

Notice that #0λ-Planar Edge-Weighted Sum is exactly Planar Vertex Cover.

7d. #Planar Vertex Cover $\propto$ #Planar Bipartite Vertex Cover
Observe that the reduction of Provan and Ball [PB83] from #Vertex Cover to #Bipartite Vertex Cover preserves planarity.

8. #Planar Bipartite Vertex Cover
#3Δ-Planar Bipartite Minimum Cardinality Vertex Cover
Let G be a bipartite planar graph. Consider the graph G' formed by taking each vertex v in G, replacing it with a cycle C_v of $2d(v)$ vertices where $d(v)$ is the degree of v in G, and connecting the neighbors of v to alternate vertices on C_v. In order for this to preserve planarity, the neighbors must be connected with the same orientation as they have in a planar embedding of G. Let M_v be the vertices in C_v which are connected to vertices outside C_v, so $|C_v| = 2|M_v|$. See Figure 8 for an example.

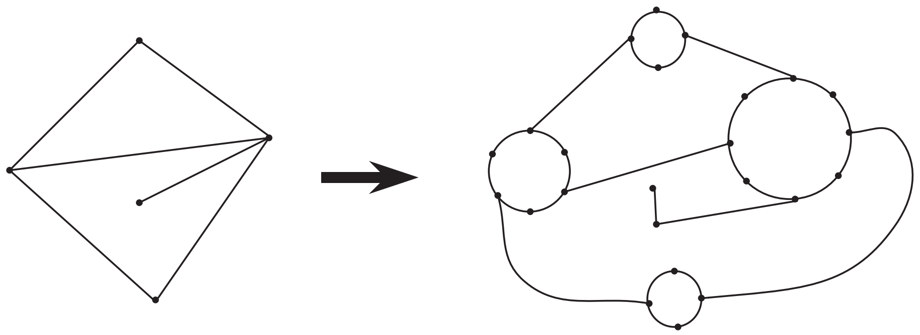

Figure 8: The tranformation of Reduction 8.

Notice that every vertex cover in G' must be of size at least $s = \sum_{v \in V} d(v)$ to cover each cycle. Further notice that the vertex covers in G are in bijective correspondence with the covers of size s in G, under the map

$$S \mapsto \bigcup_{v \in S} M_v \cup \bigcup_{v \notin S} (C_v \setminus M_v).$$

Finally, observe that G' is bipartite (since G is), planar, and of degree at most 3.

9a. #k-Regular Bipartite Perfect Matchings $\propto$
#$(2k-2)$-Regular Minimum Cardinality Vertex Cover, any fixed $k \geq 3$
This follows immediately from Lemma 2.2, noting that the line-graph of a k-regular graph is a $(2k-2)$-regular graph.

9b. #$(k-1)$-Regular Minimum Cardinality Vertex Cover $\propto$
#k-Regular Minimum Cardinality Vertex Cover, any odd $k \geq 5$
Let H be the complete graph on $k+1$ vertices with one edge removed. Consider the sequence of graphs $\{H_n^k : n \geq 0\}$ defined as follows: H_0 is the complete graph on $k+2$ vertices, labelled $u_1, v_1, \ldots, u_{(k-1)/2}, v_{(k-1)/2}, u, v, w$, with the edges $(u_1, v_1), \ldots, (u_{(k-1)/2}, v_{(k-1)/2}), (u, v), (v, w)$ removed. H_{n+1}^k is formed by taking the disjoint union of H_n^k and a new copy of H and and connecting one of the vertices of degree $k-1$ in H to the unique vertex in H_n^k of degree $k-1$.

Let G be any $(k-1)$-regular graph on n nodes $u_1, u_2, \ldots, u_n$. For any $s \geq 0$, consider the disjoint union of G with n copies of H_{2s}^k. Let the vertex of degree $k-1$ in the ith copy of H_{2s}^k be labelled v_i. We can form a k-regular graph G_s by connecting u_i to v_i for each i. Lemma A.6 tells us that there are minimum cardinality vertex covers (mcvc's) in the ith copy of H_{2s}^k both containing v_i and not containing v_i. Hence, any mcvc in G_s must be formed by taking a mcvc in G and taking mcvc's in each copy of H_{2s}^k. If G has N mcvc's and they are

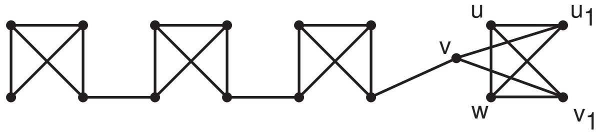

Figure 9: The graph H_2^3 in Reduction 9b.

of size c, then G_s has $N(x_s + y_s)^c(x_s)^{n-c}$ mcvc's, where x_s is the number of mcvc's in H_{2s}^k containing the vertex of degree $k-1$ and y_s is the number not containing the vertex of degree $k-1$. By Lemma A.6, $x_s = (k-1)^{s+1}/2$ and $y_s = (k-1)^s((k+1)s+4)/2$. Since x_s is easily computed, we can obtain, with a single oracle call, the number of mcvc's in G_s divided by x_s^n for any $s \geq 0$. This is simply the evaluation of the polynomial $f(x) = Nx^c$ at the value $z_s = 1 + y_s/x_s = 1 + ((k+1)s+4)/(k-1)$. Notice that $f(z_0)/f(z_1) = (z_0/z_1)^c$, so with just two oracle calls we can recover c and then N.

10. #3Δ-Planar Bipartite Minimum Cardinality Vertex Cover $\propto$ #4Δ-Planar Bipartite Vertex Cover

This is similar to 2. Let G be a planar bipartite graph of degree ≤ 3 on vertices $v_1, \ldots, v_n$. For each $s = 0, \ldots, n$, construct G_s as follows: Add nodes $v_{i,j}$ to G for $1 \leq i \leq n$, $1 \leq i \leq s$. Add edges to create disjoint chains $v_i - v_{i,1} - v_{i,2} - \cdots - v_{i,s}$ of $s+1$ nodes. Note that G_s is planar, bipartite, and of degree at most 4, since G is planar, bipartite and of degree at most 3. Now, for every vertex cover V' in G, we can obtain vertex covers in G_s by putting some of the new vertices into V' until each chain is covered. For each vertex v_i of G which is in V', we must cover the chain $v_{i,1} - \cdots - v_{i,s}$, and there are x_s ways to do this, where x_k is the number of vertex covers in a chain of k nodes. For each vertex v_i which is not in V', $v_{i,1}$ must be in any vertex cover of G_s extending V'. But we can cover the remaining chain $v_{i,2} - \cdots - v_{i,s}$ arbitrarily and there are x_{s-1} ways to do this. Thus, if A_k is the number of vertex covers in G of size exactly k, the number of vertex covers in G_s is exactly $\sum_{i=0}^n A_i(x_s)^i(x_{s-1})^{n-i}$. We can obtain these values for $s = 0, \ldots, n$ with $n+1$ oracle calls. By Lemma A.4, the x_s's are simply the Fibonacci numbers which we can compute in polynomial time. Dividing by $(x_{s-1})^n$, we obtain the evaluation of $f(x) = \sum_{i=0}^n A_i x^i$ at the points (x_s/x_{s-1}) for $s = 0, \ldots, n$. By Fact 4.1 and Lemma A.1, we can efficiently recover the coefficients of f. The first nonzero coefficient is the number of minimum cardinality vertex covers in G.

11a. #4-Regular Minimum Cardinality Vertex Cover $\propto$ #5-Regular Vertex Cover

Let G be any 4-regular graph on n nodes. We use the gadgets H_s^5 defined in Reduction 9b. As in that reduction, we can, for $0 \leq s \leq n$, form a 5-regular graph G_s by attaching n disjoint copies of H_s^5. Let A_i be the number of vertex covers of size i in G. Then the number of vertex covers in G_s is $\sum_{i=0}^n A_i(x_s+y_s)^i(x_s)^{n-i}$, where x_s is the number of vertex covers in H_s^5 containing its vertex of degree 4 and y_s is the number not containing its vertex of degree 4. By Lemma A.7, we can compute x_s and y_s in polynomial time and the sequence x_s/y_s never repeats. Thus, with $n+1$ oracle calls, we can evaluate the polynomial $f(x) = \sum_{i=0}^{n} A_i x^i$ at the $n+1$ distinct points $1 + y_s/x_s$, $s = 0, \ldots, n$. By Fact 4.1, we can recover the coefficients of f in polynomial time. The number of minimum cardinality vertex covers in G is the first nonzero coefficient.

11b. #$(k-2)$-Regular Minimum Cardinality Vertex Cover $\propto$
#k-Regular Vertex Cover
Let H be the complete graph on $k+1$ vertices with two edges incident to some vertex v removed. Consider the sequence of graphs $\{I_n^k : n \geq 0\}$ defined as follows: I_0^k is the complete graph on $k+1$ vertices with a single edge removed. I_{n+1}^k is formed by taking the disjoint union of I_n^k and a new copy of H and connecting the vertex of degree $k-2$ in H to the two vertices of degree $k-1$ in I_n^k. Let $(I_n^k)^+$ be the graph formed by adding to I_n^k a new vertex p and connecting p to the two vertices of degree $k-1$ in I_n^k.

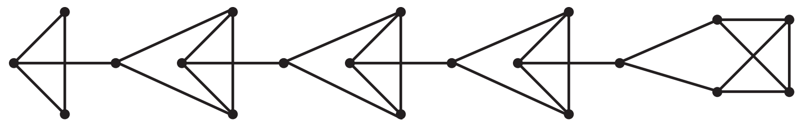

Figure 10: The graph I_4^3 in Reduction 11b.

Let G be any $(k-2)$-regular graph on n nodes $u_1, u_2, \ldots, u_n$. For any $0 \leq s \leq n$, consider the disjoint union of G with n copies of I_s^k. Let the vertices of degree $k-1$ in the ith copy of I_s^k be labelled v_i and w_i. We can form a k-regular graph G_s by connecting u_i to both v_i and w_i for each i. Let A_i be the number of vertex covers of size i in G. Then the number of vertex covers in G_s is $\sum_{i=0}^{n} A_i (x_s)^i (y_s)^{n-i}$, where x_s is the number of vertex covers in $(I_s^k)^+$ containing p and y_s is the number not containing p. By Lemma A.8, we can compute x_s and y_s in polynomial time and the sequence x_s/y_s never repeats. Thus, with $n+1$ oracle calls, we can evaluate the polynomial $f(x) = \sum_{i=0}^{n} A_i x^i$ at the $n+1$ distinct points x_s/y_s, $s = 0, \ldots, n$. By Fact 4.1, we can recover the coefficients of f in polynomial time. The number of minimum cardinality vertex covers in G is the first nonzero coefficient.

12. #4Δ-Planar Bipartite Vertex Cover $\propto$
#5Δ-Planar Bipartite Minimal Vertex Cover
This is similar to 5. Given a planar bipartite graph $G = (V, E)$ of degree ≤ 4, construct $G' = (V', E')$, where $V' = V \cup \{v' : v \in V\}$ and $E' = E \cup \{(v, v') : v \in V\}$. Note that every vertex cover W in G yields a unique minimal vertex cover $W' = W \cup \{v' : v \notin W\}$ in G' and every minimal vertex cover of G' can be obtained in this fashion. (Every minimal vertex cover contains exactly one of each pair v, v', for if both were contained, we could remove v'.) Observe that

G' is planar, bipartite, and of degree ≤ 5.

13. #5-Regular Vertex Cover $\propto$
#Regular Minimal Vertex Cover
Let G be any 5-regular graph on n nodes $u_1, u_2, \ldots, u_n$. For $r \geq 0$, let J_r be the complete bipartite graph on $(5+2r)+(5+2r)$ vertices with one edge removed. Consider the disjoint union of G with nr copies of J_r, where these copies are indexed by (i,j) with $1 \leq i \leq n, 0 \leq j \leq r$. Let G' be the graph formed by attaching each u_i to the two vertices of degree of $4+2r$ in each of $H_{i,1}, H_{i,2}, \ldots, H_{i,r}$. Notice that G' is a $(5+2r)$-regular graph. Let us consider the number of ways we can extend a vertex cover S of G of size s to a *minimal* vertex cover of G'. For each $u_i \in S$, there are 3 ways to cover each of the attached $H_{i,j}$'s so that no vertex can be removed without uncovering an edge. For each $u_i \notin S$, there are 2 ways to cover each of the $H_{i,j}$'s so that no vertex can be removed. Thus, if G has A_s vertex covers of size s, G' has $\sum_{i=0}^{n} A_s(3^r)^s(2^r)^{n-s}$ minimal vertex covers. Dividing by 2^{rn}, we get the evaluation of $f(x)=\sum_{i=0}^{n} A_s x^s$ at $(3/2)^r$. If we choose $r = \lceil \log_{3/2} n \rceil + 1$, we can, by Lemma 4.3, recover the coefficients of f in a single oracle call. The number of vertex covers in G is simply the sum of the coefficients. □

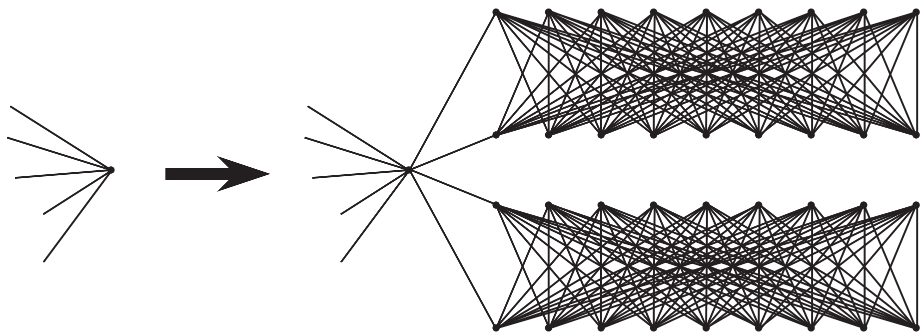

Figure 11: The transformation in Reduction 13 for $r = 2$.

Proof (of Corollary 1) These follow immediately from Theorem 1 and the discussion in Proposition 2.1. □

5 Proof of Theorem 2

In this section, we prove that it is $\mathcal{NP}$-hard to approximate the number of minimum cardinality vertex covers within ratio $2^{n^{1-\epsilon}}$. It has been observed that a direct parsimonious reduction from any $\mathcal{NP}$ language can be used to prove that approximately counting minimum cardinality vertex covers is hard in graphs whose degree is bounded by a constant [Val95]. (After modifying the nondeterministic Turing machine to have at least one trivial accepting computation.) However, it is unclear that the direct argument can achieve a degree bound as small as 3.

Our proof is based on the blow-up technique introduced in [JVV86] and later used in [Rot96, Sin93]. The intuition is: if we want to prove that it is hard to approximately count some class of objects (such as cycles) and some extremal subclass of these objects (such as Hamiltonian cycles) are $\mathcal{NP}$-hard to detect, then we should perform some blow-up procedure through which the extremal objects, if there are any, are multiplied many times more than the non-extremal ones. Thus, even a crude counting estimate in the modified instance will enable us to detect whether any of these extremal objects existed in the original problem instance. In the reductions of [JVV86, Rot96, Sin93], the blow-up procedure is applied to the original graph. However, we want to prove a result for bounded-degree graphs; it is difficult to perform any blow-up that preserves this property. Instead, we pass, via parsimonious reduction, to propositional formulae and perform the blow-up on formulae. We then reduce the number of occurrences of each variable to a constant using a standard technique, and return to graphs via another parsimonious reduction. We find that the condition that each variable occurs only a constant number of times translates to the resulting graph having bounded degree.

The approach described above is realized by the following sequence of reductions. We will prove that each of the following problems are $\mathcal{NP}$-hard in sequence. (3–7 being $\mathcal{NP}$-hard for every $\epsilon > 0$.):

1. $\frac{n}{2}$-Vertex Cover
   Input: An undirected graph G on n nodes.
   Output: Y if G has a vertex cover of size $\frac{n}{2}$, N otherwise.
2. $\frac{n}{2}$-Monotone 2Sat
   Input: A monotone 2-CNF formula F on n variables.
   Output: Y if F has a satisfying assignment with exactly $\frac{n}{2}$ variables true, N otherwise.
3. $3^{r^{1-\epsilon}}$-Approx #Guaranteed Weighted 3Sat
   Input: A formula $F = C_1 \wedge C_2 \wedge \cdots \wedge C_r$, where $C_i = (c_{i,1} \vee c_{i,2})$ or $C_i = (c_{i,1} \vee c_{i,2} \vee c_{i,3})$, on m variables, and an assignment β which satisfies F.
   Output: An approximation to
   $|\{(\alpha,t) : t \in \{1,2,3\}^r, \alpha \text{ an assignment for } F \text{ such that } \alpha \text{ makes } c_{i,t_i} \text{ true for each } i\}|$
   within ratio $3^{r^{1-\epsilon}}$.
4. $3^{(\frac{2m+3r}{18})^{1-\epsilon}}$-Approx #$2\nu$-Guaranteed Weighted 3Sat
   Input: As in 3, but with each literal occurring at most twice.
   Output: An approximation to the quantity in 3 within ratio $3^{(\frac{2m+3r}{18})^{1-\epsilon}}$.
5. $3^{(\frac{n}{18})^{1-\epsilon}}$-Approx #3Δ-Minimum Cardinality Vertex Cover
   Input: An undirected graph G of degree ≤ 3 on n nodes.
   Output: An approximation to the number of minimum cardinality vertex covers in G within ratio $3^{(\frac{n}{18})^{1-\epsilon}}$.
6. $2^{n^{1-\epsilon}}$-Approx #3Δ-Minimum Cardinality Vertex Cover

Input: An undirected graph G of degree ≤ 3 on n nodes.

Output: An approximation to the number of minimum cardinality vertex covers in G within ratio $2^{n^{1-\epsilon}}$.

7. $2^{n^{1-\epsilon}}$-Approx #3Δ-Maximum Cardinality Independent Set

Input: An undirected graph G of degree ≤ 3 on n nodes.

Output: An approximation to the number of maximum cardinality independent sets in G within ratio $2^{n^{1-\epsilon}}$.

Proof (of Theorem 2)

1. 3Sat $\propto$ $\frac{n}{2}$-Vertex Cover

The standard reduction from 3Sat to Vertex Cover takes a formula F in 3-CNF with m variables and r clauses and constructs a graph G on $2m + 3r$ vertices such that G has a vertex cover of size $m + 2r$ if and only if F is satisfiable. [GJ79] If we add r isolated vertices to G, then G has $n = 2m + 4r$ vertices and has a vertex cover of size $n/2$ iff F is satisfiable.

2. $\frac{n}{2}$-Vertex Cover $\propto$ $\frac{n}{2}$-Monotone 2Sat

This is immediate, given the discussion in the proof of Proposition 2.1.

3. $\frac{n}{2}$-Monotone 2Sat $\propto$ $3^{r^{1-\epsilon}}$-Approx #Guaranteed Weighted 3Sat

Let F be the given monotone 2-CNF formula, with r clauses. Intuitively, we want to magnify the number of choices of t for assignments in which exactly half the variables are true. Notice that the number of pairs of variables which have different values is largest when exactly half the variables are true. This is the basis of our reduction. Say F has variables $x_1, x_2, \ldots, x_n$. For each pair i, j with $i \neq j$, introduce a new variable z_{ij}, and define F_{ij} as follows.

$$\begin{aligned} F_{ij} \;=\;& (x_i \vee x_j \vee z_{ij}) \wedge (\neg x_i \vee \neg x_j \vee z_{ij}) \\ & \wedge \underbrace{(\neg z_{ij} \vee \neg z_{ij} \vee z_{ij}) \wedge (\neg z_{ij} \vee \neg z_{ij} \vee z_{ij}) \wedge \cdots \wedge (\neg z_{ij} \vee \neg z_{ij} \vee z_{ij})}_{s \ times} \end{aligned}$$

Now we define G as follows:

$$G = F \wedge \left(\bigwedge_{i \neq j} F_{ij} \right)$$

Notice that the assignment β which sets all variables to true can be used as the satisfying assignment in the input. Consider any assignment α that satisfies F. Let us consider the number of ways we can extend α to a satisfying assignment of G and choose $t \in \{1, 2, 3\}^{r+n(n-1)(s+2)}$ so that the required relationship holds. For each pair of variables x_i, x_j which take on different values, z_{ij} can either be true or false. When z_{ij} is true, there are 4 ways to choose the components of t corresponding to the first two clauses of F_{ij} (in each of those, two literals are true.) But there is only one way to choose the components of t

corresponding to the final s clauses in F_{ij}. When z_{ij} is chosen to be false, then there is only one way to choose the components of t for the first 2 clauses in F_{ij}, but there are 2^s ways to do so for the final s clauses. Thus, for each pair of variables taking different values under α, there are $4+2^s$ choices for z_{ij} and the relevant components of t.

For each pair of variables x_i, x_j which take on the same value under α, z_{ij} must be true to satisfy F_{ij}. Then there are 3 choices of the component of t corresponding one of the first two clauses of F_{ij}, but only one choice for the other clauses of F_{ij}. Thus, for each pair of variables taking on the same value under α, there are 3 choices for z_{ij} and the relevant components of t.

Let $m = \frac{n}{2}$. Suppose F has a satisfying assignment α in which exactly m variables are true. Then there are at least m^2 pairs of variables that take on different values under α. So there are at least $N_{min} = (4+2^s)^{m^2}$ choices of (β, t) to count for G.

Now suppose that no satisfying assignment of F has exactly m variables true. Then for each assignment α that satisfies F, there are at most m^2-1 pairs of variables taking on different values under and at most n^2 pairs of variables taking on the same value under α. Additionally, there are at most 2^r choices of the components of t for the original r clauses in F, so there are at most $(2^r)(4+2^s)^{m^2-1}(3^{n^2}) = N_{max}$ choices of (β, t) to count for G.

Given $0 < \epsilon < 1$, choose

$$s > \left[(\log_2 3)\left(n^2 + 2(r+3n(n-1))^{1-\epsilon}\right) + r\right]^{1/\epsilon}$$

Then,

$$\begin{aligned} s &> s^{1-\epsilon}\left[(\log_2 3)\left(n^2 + 2(r+3n(n-1))^{1-\epsilon}\right) + r\right] \\ &\geq (\log_2 3)\left(n^2 + 2(r+n(n-1)(s+2))^{1-\epsilon}\right) + r \\ 4+2^s &> \left(3^{n^2+2(r+n(n-1)(s+2))^{1-\epsilon}}\right)2^r \end{aligned}$$

$$\frac{N_{min}}{N_{max}} = \frac{4+2^s}{2^r 3^{n^2}} > \left(3^{(r+n(n-1)(s+2))^{1-\epsilon}}\right)^2 = \left(3^{z^{1-\epsilon}}\right)^2$$

where z is the number of clauses in G. Say γ is an approximation to the number N of (β, t) that work for G within ratio $\delta = 3^{z^{1-\epsilon}}$. If F has a satisfying assignment with $\frac{n}{2}$ variables true, then $N \geq N_{min} \Rightarrow \gamma \geq N_{min}/\delta > N_{max}\delta$. If F doesn't have a satisfying assignment with $\frac{n}{2}$ variables true, then $N \leq N_{max} \Rightarrow \gamma \leq N_{max}\delta$. Hence, by comparing γ with $N_{max}\delta$, we can decide if N has a satisfying assignment of size $\frac{n}{2}$.

4. $3^{r^{1-\epsilon}}$-Approx #Guaranteed Weighted 3Sat $\propto$
$3^{(\frac{2m+3r}{18})^{1-\epsilon}}$-Approx #$2\nu$-Guaranteed Weighted 3Sat

Given F on variable set X, consider each variable $x \in X$ in turn. If x occurs $k > 2$ times in F, replace the ith occurrence with a new variable x_i and conjoin the clauses $(x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \cdots \wedge (x_n \vee \neg x_1)$ onto F. This corresponds to requiring that $x_1 \leftarrow x_2 \leftarrow \cdots \leftarrow x_n \leftarrow x_1$, which is to say that all the x_i's must take on the same value in any satisfying assignment. Doing this for each of the original variables in F, we obtain a formula F', in which each *literal* appears at most twice. Now, every satisfying assignment to F yields a satisfying assignment to F' by assigning $x_1, \ldots, x_k$ the value that x got, and conversely. We need to note that the number of choices of t does not change. This is clear, because the number of ways to choose t_i does not change if the ith clause was originally in F, and in each of the new clauses, exactly one literal is true in every satisfying assignment, so there is only one way to choose t_i. Observe, additionally, that we can use the assignment β which sets all variables from X to true and all variables from Y to false as the required assignment in the input. Note that if the original formula F has r clauses, then F' has at most $3r$ variables and $4r$ clauses. Thus, if F' has m variables and s clauses, an approximation within ratio $3^{(\frac{2m+3s}{18})^{1-\epsilon}}$ for F yields an approximation within ratio $3^{r^{1-\epsilon}}$ for F.

5. $3^{(\frac{2m+3r}{18})^{1-\epsilon}}$-Approx 2ν-#Guaranteed Weighted 3Sat $\propto$ $3^{(\frac{n}{18})^{1-\epsilon}}$-Approx 3Δ-#Minimum Cardinality Vertex Cover

This is similar to Reduction 8 in the proof of Theorem 1. Given F, we construct a graph with $n + r_1$ dumbbells and r_2 triangles, where n is the number of variables in F, r_1 is the number of clauses of size 2, and r_2 is the number of clauses of size 3. Specifically, for each variable x, we have two nodes, x and $\neg x$, which are connected by an edge. For each clause $(c_{i,1} \vee c_{i,2})$, we have a dumbbell and we connect the endpoints to the nodes corresponding to the literals $c_{i,1}$ and $c_{i,2}$. For each clause $(c_{i,1} \vee c_{i,2} \vee c_{i,3})$, we have a triangle and connect the vertices to the nodes corresponding to $c_{i,1}$, $c_{i,2}$, and $c_{i,3}$.

Formally, if F is over variables $\{x_1, \ldots, x_n\}$, let $G = (V, E)$, where

$$\begin{aligned}
V &= \{x_i, \neg x_i : 1 \le i \le n\} \cup \{(i,j) : |C_i| \ge j, 1 \le i \le r, 1 \le j \le 3\} \\
E &= \{\langle x_i, \neg x_i \rangle : q \le i \le n\} \\
&\cup \{\langle (i,1), (i,2) \rangle : 1 \le i \le r\} \cup \{\langle (i,1), (i,3) \rangle, \langle (i,2,), (i,3) \rangle : |C_i| = 3, 1 \le i \le r\} \\
&\cup \{\langle c_{i,j}, (i,j) \rangle : |C_i| \ge j, 1 \le i \le r, 1 \le j \le 3\}
\end{aligned}$$

Observe that every vertex cover in G is of size at least $n + r_1 + 2r_2$ We will show that there is exactly one such cover for every choice of (α, t) for F. Given (α, t) satisfying the appropriate conditions, let

$$V' = \{x : \alpha(x) = 1\} \cup \{\neg x : \alpha(x) = 0\} \cup \{(i,j) : t_i \ne j, |C_i| \ge j\}$$

V' is a vertex cover of size $n + r_1 + 2r_2$. It is clear that the first two types of edges are covered. To see that the edges of the last type are covered, just note that if $c_{i,j}$ is not in the cover, then $c_{i,j}$ is not true, so $t_i \ne j$, which means that (i,j) is in the cover. There is at least one cover of this size, since F is

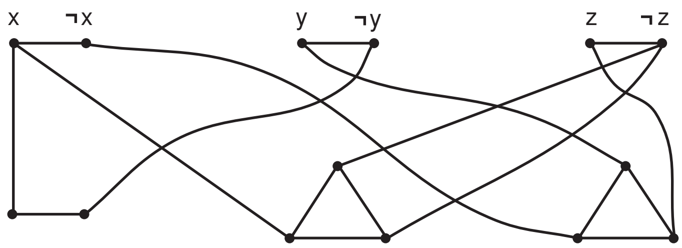

Figure 12: The graph produced for $F = (x \vee \neg y) \wedge (x \vee \neg z \vee \neg z) \wedge (\neg x \vee y \vee z)$ in Reduction 5.

satisfiable. Additionally, every vertex cover V' of size $n + r_1 + 2r_2$ must contain exactly one of each pair $x_i, \neg x_i$ and must be missing exactly one vertex (i, j) for each j. Thus, if we set $\alpha(x_i) = 1$ iff $x_i \in V'$ and $t_i = j$ iff $(i, j) \notin V'$, we obtain a correct pair (α, t). (Note that if $t_i = j$, then $(i, j) \notin V'$, so $c_{i,j} \in V'$, so $\alpha(c_{i,j}) = 1$, as needed.)

Finally, observe that each 'clause' node has degree ≤ 3, whereas each 'literal' node has degree one greater than the number of times it appears in F. Hence, the graph has degree at most 3, and this is a parsimonious reduction. The number of nodes n in G is at most $2m + 3r$, where m is the number of variables in F and r is the number of clauses, so an approximation within ratio $3^{(\frac{n}{18})^{1-\epsilon}}$ for G gives an approximation within ratio $3^{(\frac{2m+3r}{18})^{1-\epsilon}}$ for F.

6. $3^{(\frac{n}{18})^{1-\epsilon/2}}$-Approx #3Δ-Minimum Cardinality Vertex Cover $\propto$ $2^{n^{1-\epsilon}}$-Approx #3Δ-Minimum Cardinality Vertex Cover

Let $\epsilon > 0$ be fixed. Then for all n greater than some sufficiently large N, $n^{1-\epsilon} \leq \left(\frac{\log_2 3}{18^{1-\epsilon/2}}\right) n^{1-\epsilon/2}$, *i.e.* $2^{n^{1-\epsilon}} \leq 3^{(\frac{n}{18})^{1-\epsilon/2}}$. Hence, given an algorithm that approximates the number of minimum cardinality vertex covers within ratio $2^{n^{1-\epsilon}}$, we can obtain one that approximates the same quantity within ratio $3^{(\frac{n}{18})^{1-\epsilon/2}}$, by keeping a table of the number of minimum cardinality vertex covers in the (finitely many) graphs with fewer than N nodes.

7. $2^{n^{1-\epsilon}}$-Approx #3Δ-Minimum Cardinality Vertex Cover $\propto$ $2^{n^{1-\epsilon}}$-Approx #3Δ-Maximum Cardinality Independent Set

Immediate from Proposition 2.1. □

6 Conclusion

The study of counting and its computational complexity is both interesting and important. However, our understanding of the computational aspects of counting is far from complete. In particular, very little is known about counting, both exact and approximate, in restricted cases. In this paper, we have a answered a few questions about counting in restricted cases, but there are still many open problems. We believe that the tools developed here are likely to prove useful in obtaining restricted-case results for other counting problems.

Even regarding just the problems studied here, several unanswered questions stand out. For one, we have shown that a number of problems are hard in bounded-degree bipartite graphs and constant-degree regular graphs, but we do not know what happens if these conditions are imposed simultaneously. Do these problems remain hard in bipartite k-regular graphs, or even just bipartite regular graphs? In addition, we know that all the problems become tractable in degree 2, but some of our results only show hardness for degree 4 or higher. It seems likely that all of the problems remain $\#\mathcal{P}$-complete in degree 3, but this is still unknown.

Perhaps the most useful application of negative results such as ours is to guide the search for positive results. Our results mean that it would be significant if polynomial-time approximate-counting algorithms for structures such as vertex covers could be found even in very restricted cases. Sinclair and Jerrum's Markov chain techniques [SJ89, Sin93] have yielded polynomial-time approximation algorithms for a number of counting problems [Kan94]; perhaps they can be applied to the problems here. Some progress has already been made in this direction. There are polynomial-time Markov chain algorithms for approximately counting the number of matchings in general graphs and the number of perfect matchings in dense graphs or graphs with large factors [JS89, DL92]. Luby and Vigoda [LV97] have recently devised a Markov chain which approximates the number of independent sets in graphs of degree at most four [LV97]. These results, and that of [LV97] in particular, are even more surprising in light of Theorem 1.

Despite the successes of the Markov chain technique, it is still possible that the approximate versions of some of the problems studied here are also hard. Theorem 2 demonstrates that it is possible to get a restricted-case result for at least one approximate-counting problem. This might prove useful as a starting point for additional such results. Luby and Vigoda [LV97] have recently shown that it is $\mathcal{NP}$-hard to approximately count independent sets in graphs of degree at most k for some large constant k. It would be interesting to close the gap between their positive result for degree 4 and this negative result.

As evidenced by the open problems mentioned here, the complexity of counting is a rich and fascinating area in which there is still much work to be done. Perhaps the techniques and results introduced in this paper will help answer some of the many unanswered questions about the complexity of counting.

A Technical Lemmas

In order for our interpolation reductions to work, it is essential that the interpolation points are distinct. The simplest case of this occurred in Reductions 2 and 10 of Theorem 1, in which we interpolated at consecutive ratios of Fibonacci numbers.

Lemma A.1 *Let f_n denote the nth Fibonacci number. That is, $f_0 = 1$, $f_1 = 1$, and $f_{n+2} = f_{n+1} + f_n$ for all $n \geq 0$. Let $r_n = f_{n+1}/f_n$. Then $r_i \neq r_j$ for any $i \neq j$.*

Proof It is a well-known fact that $r_n \to \delta$ as $n \to \infty$, where $\delta = (1 + \sqrt{5})/2$. From the Fibonacci recurrence, we see that $r_{n+1} = 1 + \frac{1}{r_n}$. Hence, if $r_i = r_j$ for some $i < j$, the sequence $\{r_n\}$ would have to be eventually periodic with period $j - i$. As a nonconstant periodic sequence never converges, it would have to be eventually constant at δ. By noting that $r_n = (r_{n+1} - 1)^{-1}$ and that δ is a fixed point of this recurrence, we can conclude that the sequence would have to be constant at δ starting with r_0. But $r_0 = 1$. □

When our interpolation points are not as familiar as Fibonacci numbers, we need a more general tool for determining whether they repeat. The following two lemmas provide such a tool.

Lemma A.2 *Let a, b, c, d, be rational numbers and let α and β be nonzero complex numbers. Let the sequence z_n be defined by*

$$z_n = \frac{a\alpha^n + b\beta^n}{c\alpha^n + d\beta^n}$$

Then the sequence $\{z_n\}$ repeats iff $AD - BC = 0$ or α/β is a root of unity.

Proof Cross-multiplying, we see that $z_n = z_m$ iff $(ad - bc)(\alpha^n\beta^m - \beta^m\alpha^n) = 0$ iff $ad - bc = 0$ or $(\alpha/\beta)^{n-m} = 0$. □

Lemma A.3 *Let A, B, C, D, x_0, and y_0 be rational numbers. Define the sequences $\{x_n\}$ and $\{y_n\}$ recursively by $x_{n+1} = Ax_n + By_n$ and $y_{n+1} = Cx_n + Dy_n$. Then the sequence $\{z_n = x_n/y_n\}$ never repeats as long as all of the following conditions hold:*

$$AD - BC \neq 0 \tag{1}$$
$$D^2 - 2AD + A^2 + 4BC \neq 0 \tag{2}$$
$$D + A \neq 0 \tag{3}$$
$$D^2 + A^2 + 2BC \neq 0 \tag{4}$$
$$D^2 + AD + A^2 + BC \neq 0 \tag{5}$$
$$D^2 - AD + A^2 + 3BC \neq 0 \tag{6}$$
$$By_0^2 - Cx_0^2 - (A - D)x_0y_0 \neq 0 \tag{7}$$

Proof Let α and β be the eigenvalues of the matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$. It is well-known that, as long as α and β are distinct, the general solution to the 2×2 system of linear recurrences describing x_n and y_n is given by $x_n = a\alpha^n + b\beta^n$

and $y_n = c\alpha^n + b\beta^n$. So, by the previous lemma, as long as $\alpha \neq \beta$ and neither α nor β is zero, $\{z_n\}$ can repeat only if $ad - bc = 0$ or α/β is a root of unity.

If α/β is a root of unity, it must be one of degree 1 or 2 over $\mathbb{Q}$, as $\alpha/\beta \in \mathbb{Q}(\alpha, \beta) = \mathbb{Q}(\alpha)$, which is a field extension of degree ≤ 2 over $\mathbb{Q}$. The degree of a primitive nth root of unity over $\mathbb{Q}$ is $\phi(n)$ (see, e.g., [IR90, Sec. 13.2, Thm. 1]), where ϕ is the Euler totient function. By the formula $\phi(\prod p_i^{\alpha_i}) = \prod p_i^{\alpha_i - 1}(p_i - 1)$ for distinct primes p_i, one sees that only n for which $\phi(n) \leq 2$ are $1, 2, 3, 4$, and 6. The irreducible polynomials over $\mathbb{Q}$ for the corresponding primitive roots of unity are $x - 1$, $x + 1$, $x^2 + x + 1$, $x^2 + 1$, and $x^2 - x + 1$. So to check that α/β is not a root of unity, we need only check that α/β does not satisfy any of these polynomials. Using the quadratic formula, we can express α and β in terms of A, B, C, and D. The first 6 conditions in the lemma come from substituting these expressions into the polynomials that test whether (1) α or β is zero, (2) $\alpha = \beta$, and (3)–(6) α/β is a 2nd, 3rd, 4th, or 6th root of unity.

As long as $\alpha \neq \beta$ and neither are zero, we can solve for a, b, c, and d in terms of A, B, C, D, and the initial conditions x_0, y_0. Condition (7) amounts to testing whether $ad - bc = 0$ (given that $D^2 - 2AD + A^2 + 4BC \neq 0$, which is tested by condition 2). □

Note that Lemma A.1 follows from Lemma A.2 with $A = B = C = x_0 = y_0 = 1$ and $D = 0$. The remainder of our lemmas prove facts about sequences that arise in our reductions.

Lemma A.4 *Let C_n be the graph consisting of a chain of n nodes. Then the number of vertex covers in C_n is f_{n+1} and the number of matchings in C_n is f_n, where f_k is the kth Fibonacci number, as defined in Lemma A.1.*

Proof Let x_n be the number of vertex covers in a chain of n vertices. Given a chain $C_n = v_1 - v_2 - \cdots - v_n$ with $n \geq 2$, the number of vertex covers in C_n containing v_1 is x_{n-1} and the number not containing v_1 is x_{n-2}. Thus, $x_n = x_{n-1} + x_{n-2}$, which is the Fibonacci recurrence. Also note that $x_0 = 1$ and $x_1 = 2$. To obtain the result for matchings, observe that the line-graph of C_n is C_{n-1} and apply Lemma 2.2. □

Lemma A.5 *For $k \geq 4$, let L_n^k be the graph defined in Reduction 4 of the proof of Theorem 1. Let v and w be the vertices in L_n^k of degree $k-1$. Let x_s be the number of matchings in L_n^k in which both v and w are matched, let y_s be the number in which v is matched but w is not, and let z_s be the number in which neither v nor w is matched. Then we can compute x_s, y_s, and z_s in time polynomial in s and the sequence $\{w_s = (y_s + z_s)/(x_s + 2y_s + z_s)\}$ never repeats.*

Proof For all m, let M_m denote the number of matchings in the complete graph on m vertices. Observe that $M_{m+1} = M_m + mM_{m-1}$. By inspection, we can verify that the sequences x_s and y_s satisfy the following recurrences:

$$x_{s+1} = (k-1)M_{k-1}x_s + (M_k + (k-2)M_{k-1})y_s$$

$$y_{s+1} = M_k x_s + (M_k + M_{k-1}) y_s$$

It is easy to see that y_s and z_s satisfy the same recurrence relations, *i.e.*

$$\begin{aligned} y_{s+1} &= (k-1)M_{k-1}y_s + (M_k + (k-2)M_{k-1})z_s \\ z_{s+1} &= M_k y_s + (M_k + M_{k-1})z_s \end{aligned}$$

The initial conditions are $x_0 = kM_{k-1} - M_k$, $y_0 = M_k - M_{k-1}$, and $z_0 = M_{k-1}$. We can compute x_s, y_s, and z_s in polynomial time using these recurrences. Because we have three sequences here, we cannot apply Lemma A.3 directly. However, the proof here is nearly identical to the one of Lemma A.3, so we do not include the details that are worked out there. Since the two pairs of sequences satisfy the same recurrence relations, closed forms for these sequences will be of the form $x_s = a\alpha^s + b\beta^s, y_s = c\alpha^s + d\beta^s, z_s = e\alpha^s + f\beta^s$. So

$$w_s = \frac{(c+e)\alpha^s + (d+f)\beta^s}{(a+2c+e)\alpha^s + (b+2d+f)\beta^s}$$

This sequence will not repeat as long as α/β is not a root of unity and $(c+e)(b+2d+f) - (d+f)(a+2c+e) \neq 0$. The conditions for α/β to not be a root of unity are the same as conditions (1)–(6) of Lemma A.3. In our case, these conditions become, after simplification:

$$\begin{aligned} -M_k^2 - M_k M_{k-1} + (k-1)M_{k-1}^2 &\neq 0 & (1) \\ 5M_k^2 + (2k-4)M_k M_{k-1} + (k^2-4k+4)M_{k-1}^2 &\neq 0 & (2) \\ M_k + kM_{k-1} &\neq 0 & (3) \\ 3M_k^2 + (2k-2)M_k M_{k-1} + (k^2-2k+2)M_{k-1}^2 &\neq 0 & (4) \\ 2M_k^2 + (2k-1)M_k M_{k-1} + (k^2-k+1)M_{k-1}^2 &\neq 0 & (5) \\ 4M_k^2 + (2k-3)M_k M_{k-1} + (k^2-3k+3)M_{k-1}^2 &\neq 0 & (6) \end{aligned}$$

For $k \geq 0$, the left-hand sides of (2)–(6) are each sums of positive quantities and hence cannot be zero. Condition (1) is a bit harder to handle. Dividing by M_{k-1}^2 and applying the quadratic formula, we see that the condition can be reformulated as:

$$\frac{M_k}{M_{k-1}} \neq \frac{1 \pm \sqrt{4k-3}}{2}$$

Moser and Wyman [MW55] have shown the following:[7]

$$\frac{1+\sqrt{4k-3}}{2} \leq \frac{M_k}{M_{k-1}} \leq \frac{1+\sqrt{4k+1}}{2}$$

[7] Moser and Wyman discuss the number of solutions to $x^2 = 1$ in the symmetric group on k elements. It is easy to see that this quantity is exactly M_k.

The proof of this fact is by straightforward induction, applying the the recurrence $M_{k+1} = M_k + kM_{k-1}$. The same proof actually shows that *strict* inequality holds (on both sides) for all $k \geq 4$, as long as we use $k = 4$ as our base case. Thus condition (1) is also satisfied.

The only condition left to check is $(c+e)(b+2d+f)-(d+f)(a+2c+e) \neq 0$. In our situation, this reduces to

$$-2M_k^3 + (6-2k)M_{k-1}M_k^2 + (4k-6)M_{k-1}^2 M_k + (2k^2-6k+4)M_{k-1}^2 \neq 0$$

Dividing by M_{k-1}^3, we obtain a cubic polynomial in M_k/M_{k-1} which vanishes iff

$$\frac{M_k}{M_{k-1}} \in \left\{-k+2, \frac{1 \pm \sqrt{4k-3}}{2}\right\}$$

The strengthened Moser-Wyman result shows that this cannot hold for any $k \geq 4$. □

Lemma A.6 *Fix k to be an odd integer ≥ 3. Let H_n^k be the graph defined in reduction 9b of the proof of Theorem 1. Let v be the unique vertex in H_n^k of degree $k-1$. Then the number of minimum cardinality vertex covers in H_{2m}^k containing v is $(k-1)^{m+1}/2$ and the number not containing v is $(k-1)^m((k+1)m+4)/2$.*

Proof First, let H be the complete graph on $k+1$ vertices with one edge removed. We now prove the lemma by induction on m.
$m = 0$: It is easily verified that the size of the minimum cardinality vertex cover (mcvc) in H_0 is k, that there are $(k-1)/2$ such covers containing v, and that there are 2 such covers not containing v.
Induction step: Let v' be the vertex of degree $k-1$ in $H_{2(m+1)}$ and let v be the vertex of degree $k-1$ in H_{2m}. Now observe that, the smallest vertex cover in H is of size $k-1$ and the only such cover omits both the vertices of degree $k-1$. The two copies of H added to H_{2m} to form $H_{2(m+1)}$ cannot both simultaneously be covered by covers of size $k-1$, for this would leave the edge between them uncovered. Hence, the smallest possible cover for $H_{2(m+1)}$ could only come from taking a mcvc on H_{2m}, a cover of size $k-1$ on one of the added copies of H and a cover of size k on the other copy of H. It is now easy to treat the problem in cases: For each mcvc of H_{2m} that contains v, there are $k-1$ mcvc's in $H_{2(m+1)}$ containing v' and $k+1$ mcvc's not containing v'. For each mcvc of H_{2m} not containing v , there are 0 mcvc's containing v' and $k-1$ mcvc's not containing v'. By induction hypothesis, this gives a total of $(k-1)((k-1)^{m+1}/2) = (k-1)^{m+2}/2$ mcvc's containing v' and $(k+1)((k-1)^{m+1}/2)+(k-1)[(k-1)^m((k+1)m+4)/2] = (k-1)^{m+1}((k+1)(m+1)+4)/2$ not containing v'. □

Lemma A.7 *Fix k to be an odd integer ≥ 3. Let H_n^k be the graph defined in Reduction 9b of the proof of Theorem 1. Let v be the unique vertex in H_n^k of degree $k-1$. Define x_s to be the number of vertex covers in H_s^k containing v and y_s to be the number not containing v. Then x_s and y_s can be computed in time polynomial in s and the sequence $\{x_s/y_s\}$ never repeats.*

Proof By inspection, we can verify that the sequences x_s and y_s satisfy the following recurrences:

$$\begin{aligned} x_{s+1} &= (k+1)x_s + ky_s \\ y_{s+1} &= 2x_s + y_s \end{aligned}$$

The initial conditions are $x_0 = (3k+3)/2$ and $y_0 = 3$. We can compute x_s and y_s in time polynomial in s by simply using these recurrences. To show that x_s/y_s never repeats, we apply Lemma A.3. In our situation, the conditions of that theorem become:

$$\begin{aligned} -k+1 &\neq 0 && (1)\\ k^2+8k &\neq 0 && (2)\\ k+2 &\neq 0 && (3)\\ k^2+6k+2 &\neq 0 && (4)\\ k^2+5k+3 &\neq 0 && (5)\\ k^2+7k+1 &\neq 0 && (6)\\ (9k-9)/2 &\neq 0 && (7) \end{aligned}$$

It is clear that all of the above hold for all odd integers $k \geq 3$. □

Lemma A.8 *Fix k to be an integer ≥ 3. Let $(I_n^k)^+$ be the graph defined in Reduction 11b of the proof of Theorem 1. Let p be the vertex in $(I_n^k)^+$ of degree 2. Define x_s to be the number of vertex covers in I_s^k containing v and let y_s be the number not containing v. Then x_s and y_s can be computed in time polynomial in s and the sequence $\{x_s/y_s\}$ never repeats.*

Proof By inspection, we can verify that the sequences x_s and y_s satisfy the following recurrences:

$$\begin{aligned} x_{s+1} &= (k+1)x_s + 3y_s \\ y_{s+1} &= (k-1)x_s + y_s \end{aligned}$$

The initial conditions are $x_0 = k+3$ and $y_0 = k$. We can compute x_s and y_s in time polynomial in s by simply using these recurrences. To show that x_s/y_s never repeats, we apply Lemma A.3. In our situation, the conditions of that theorem become:

$$\begin{aligned} -2k+4 &\neq 0 && (1)\\ k^2+12k-12 &\neq 0 && (2)\\ k+2 &\neq 0 && (3)\\ k^2+8k-4 &\neq 0 && (4) \end{aligned}$$

$$\begin{aligned} k^2 + 6k &\neq 0 && (5) \\ k^2 + 10k - 8 &\neq 0 && (6) \\ k^2 - 3k + 9 &\neq 0 && (7) \\ & && (8) \end{aligned}$$

It is easily verified that all of the above hold for all integers $k \geq 3$. $\square$

Acknowledgements

I am indebted to a few people for helping me complete this work: My undergraduate advisor, Leslie Valiant, who sparked my interest in complexity theory and provided advice and inspiration throughout; Dan Roth, who suggested these problems and guided me through the initial stages of research; and Jennifer Sun, who spent many hours reading and critiquing the original version of this work.

References

[BD97] Russ Bubley and Martin Dyer. Graph orientations with no sink and an approximation for a hard case of #SAT. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 248–257, New Orleans, Louisiana, 5–7 January 1997.

[BW90] Graham Brightwell and Peter Winkler. Counting linear extensions is #P-complete. Technical Report 90–49, DIMACS, July 1990.

[DF89] M. E. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *Society for Industrial and Applied Mathematics Journal on Computing*, 18:205–226, 1989.

[DL92] Paul Dagum and Michael Luby. Approximating the permanent of graphs with large factors. *Theoretical Computer Science*, 102:283–305, 1992.

[Fis61] Michael E. Fisher. Statistical mechanics of dimers on a plane lattice. *Physical Review, Ser. 2*, 124:1664–1672, 1961.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[Gol93] Leslie Ann Goldberg. Automating polya theory: The computational complexity of the cycle index polynomial. *Information and Computation*, 105:268–288, 1993.

[Har67] F. Harary, editor. *Graph Theory and Theoretical Physics.* Academic Press, 1967.

[IR90] Kenneth Ireland and Michael Rosen. *A Classical Introduction to Modern Number Theory*. Graduate Texts in Mathematics 84. Springer, 2 edition, 1990.

[Jer87] Mark Jerrum. Two-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics*, 48(1/2):121–134, 1987. Erratum in 59(3/4):1087–1088, 1990.

[Jer90] Mark Jerrum. Erratum: Two-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics*, 59(3/4):1087–1088, 1990.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *Society for Industrial and Applied Mathematics Journal on Computing*, 18(6):1149–1178, December 1989.

[JS93] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *Society for Industrial and Applied Mathematics Journal on Computing*, 22(5):1087–1116, October 1993.

[JVV86] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[JVW90] F. Jaeger, D. L. Vertigan, and D. J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108:35–53, 1990.

[Kan94] Ravi Kannan. Markov chains and polynomial time algorithms. In *Annual Symposium on the Foundations of Computer Science*, pages 656–669. Institute of Electrical and Electronics Engineers, 1994.

[Kas63] P.W. Kasteleyn. Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4:287–293, 1963.

[Lin86] Nathan Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic and Discrete Methods*, 7:331–335, April 1986.

[LV97] Michael Luby and Eric Vigoda. Approximately counting up to four (extended abstract). In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*. Association for Computing Machinery, Inc., February 1997.

[MW55] Leo Moser and Max Wyman. On solutions of $x^d = 1$ in symmetric groups. *Canadian Journal of Mathematics*, 7:159–168, 1955.

[MW92] Milena Mihail and Peter Winkler. On the number of Eulerian orientations of a graph. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–145, Orlando, Florida, 27–29 January 1992.

[OTTW96] Mitsunori Ogihara, Thomas Thierauf, Seinosuke Toda, and Osamu Watanabe. On the closure properties of $\#P$ in the context of $PF \circ \#P*$. *Journal of Computer and System Sciences*, 53:171–179, October 1996.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[PB83] J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *Society for Industrial and Applied Mathematics Journal on Computing*, 12(4):777–788, 1983.

[Rot96] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, April 1996.

[Sin93] Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov chain Approach*. Progress in Theoretical Computer Science. Birkhäuser, 1993.

[SJ89] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation, and rapidly mixing markov chains. *Information and Computation*, 82:93–133, 1989.

[TF61] H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics — an exact result. *Philosophical Magazine, Ser. 8*, 6:1061–1063, 1961.

[Vad95] Salil P. Vadhan. The complexity of counting. Undergraduate thesis, Harvard University, 1995. Available from http://www-math.mit.edu/~salil.

[Val79a] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[Val79b] Leslie G. Valiant. The complexity of enumeration and reliability problems. *Society for Industrial and Applied Mathematics Journal on Computing*, 8(3):410–421, August 1979.

[Val95] Leslie G. Valiant, March 1995. Private communication.