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

Salil P. Vadhan*

September 20, 2000

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, we prove 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. As corollaries, we obtain results about counting cliques in restricted classes of graphs, and counting satisfying assignments to restricted classes of monotone 2-CNF formulae. To achieve these results, a new interpolation-based reduction technique which preserves properties such as constant degree is introduced.

1 Introduction

From the time that Valiant [Val79a, Val79b] introduced the class $\#\mathcal{P}$ of counting problems and gave a complexity-theoretic explanation for the apparent difficulty of enumeration, counting has held an important place in theoretical computer science. Although many researchers have continued Valiant's work by adding to the list of $\#\mathcal{P}$-complete problems,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. Counting problems also arise naturally in Artificial Intelligence research [Orp90, Pro90, Rot96]. As explained by Roth [Rot96], various methods used in reasoning, such as computing "degree of belief" and "Bayesian belief networks" are computationally equivalent to counting the number of satisfying assignments to a propositional formula. Thus, understanding the types of propositional formulae for which counting satisfying assignments is feasible tells us the extent to which these reasoning techniques might be useful. Graph-theoretic counting problems such as the ones we consider also appear often in statistical physics (cf., [Har67, JS93, Jer87, LV99]).

Perhaps the most significant deficiency in our understanding of counting is that, in many 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.

While researchers have managed to prove a number of restricted-case hardness results for counting problems, the techniques have been somewhat ad hoc, requiring new ideas for each problem. One reason for this is that many of the known reductions between counting problems employ 'blow-up' techniques, which

*Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138. E-mail: salil@deas.harvard.edu. Much of this work originally appeared in the author's undergraduate thesis [Vad95], done at Harvard University under the supervision of Leslie Valiant, and some of it was done while the author was at MIT, supported by a DOD NDSEG graduate fellowship.

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.

Our results. 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 4.1 that counting matchings, vertex covers, independent sets, and variants of these structures remains difficult in planar bipartite graphs of bounded degree and in regular graphs of constant degree. As immediate corollaries, we deduce hardness results for counting cliques and satisfying assignments in restricted classes of graphs and formulae. Our main reduction 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, the technique is quite general, and it can be applied to different problems in an almost mechanical manner.

A summary of our results, together with previous work, is given in Tables 1 and 2. Precise definitions of the problems we consider can be found in Sections 3 and 4 and a more detailed summary of related work is given in Section 2. We note that our results for vertex covers, independent sets, monotone 2-CNF satisfying assignments, and cliques are essentially restatements of each other via well-known equivalences (cf., Propositions 3.1 and 3.2), but we list them separately on Tables 1 and 2 for clarity and comparison to previous work.

Several of our results answer open problems explicitly stated in previous work: counting maximal matchings in unrestricted graphs [Val79b], counting satisfying assignments to monotone 2-CNF formulae in which every variable appears a bounded number of times [Rot96], and counting satisfying assignments to planar 2-CNF formulae [HMRS98].

When counting remains hard even in restricted cases, 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 Proposition 4.3, 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.

Techniques. Our main technique is best illustrated with an example: reducing #PERFECT MATCHINGS to #MATCHINGS while preserving the sparsity of the input graph. Suppose we are given an oracle which counts all matchings in a graph, and we want to use this oracle to count the number of perfect matchings in a graph G. Let $v_1, \ldots, v_n$ be the vertices of G. For $s = 0, \ldots, n$, consider the graph G_s obtained by adding disjoint chains $v_i - v_{i,1} - v_{i,2} - \cdots - v_{i,s}$ to each vertex v_i of G.

We will describe the number of matchings in G_s in terms of matchings in G. Consider any matching M in G, and let us count the number of ways M can be extended to a matching in G_s. For each vertex v_i of G which is matched by M, the edge $(v_i, v_{i,1})$ cannot be added to the matching, but we can choose an arbitrary matching for the chain $v_{i,1} - \cdots - v_{i,s}$. 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. Thus the number of ways to extend M to a matching in G_s is exactly $x_s^j x_{s+1}^{n-j}$, where j is the number of vertices matched by M and x_t denotes the number of matchings in chain of t nodes. Therefore, if we let A_j denote the number matchings in G in which exactly j nodes are matched, then G_s has exactly $\sum_{j=0}^{n} A_j x_s^j x_{s+1}^{n-j}$ matchings. We can obtain these values for $s = 0, \ldots, n$ with $n+1$ oracle calls. Dividing by x_{s+1}^n, we obtain the evaluation of the polynomial $f(x) = \sum_{j=0}^{n} A_j x^j$ at the points (x_s/x_{s+1}) for $s = 0, \ldots, n$. Now, we'd like to use polynomial interpolation to recover the coefficients of f, and in particular, the leading coefficient A_n which is the number of perfect matchings in G. We can do this provided we can compute the values x_s and the evaluation points x_s/x_{s+1} are distinct.

Luckily, it is not hard to get a handle on x_s, the number of matchings in a chain of s nodes. It turns out that x_s is simply the s'th Fibonacci number! One can use the Fibonacci recurrence to compute the values

Table 1: #$\mathcal{P}$-completeness Results for Counting Problems in Restricted Classes of Graphs

Problem	Polynomial-time Solvable	Previous Results	This Paper
#Perfect Matchings	• planar [Fis61, Kas63, TF61] • $\Delta \leq 2$	• k-regular bipartite, any $k \geq 3$ [DL92] • $(n-k)$-regular bipartite, any $k \geq 3$ [DL92]	
#Matchings	• $\Delta = 2$	• bipartite [Val79b] • planar [Jer87]	• bipartite, $\Delta = 4$ • planar bipartite, $\Delta = 6$ • k-regular, any $k \geq 5$
#Maximal Matchings	• $\Delta = 2$		• bipartite, $\Delta = 5$ • planar bipartite, $\Delta = 7$
#Vertex Covers #Independent Sets	• $\Delta = 2$ • $n - \delta$ constant	• bipartite [PB83]	• planar bipartite, $\Delta = 4$ • k-regular, any $k \geq 5$[1]
#Min Cardinality Vertex Covers #Max Cardinality Independent Sets	• $\Delta = 2$ • $n - \delta$ constant	• bipartite [PB83] • planar [HMRS98]	• planar bipartite, $\Delta = 3$ • k-regular, any $k \geq 4$
#Minimal Vertex Covers #Maximal Independent Sets	• $\Delta = 2$ • $n - \delta$ constant	• bipartite [PB83][2]	• planar bipartite, $\Delta = 5$ • regular
#Cliques, #Maximal Cliques #MaxCardinality Cliques[3]	• $\delta = n - 2$ • Δ constant • planar • bipartite	• unrestricted [Val79b][2]	• regular • $\delta = n - 5$
#Bipartite Cliques, #Maximal Bipartite Cliques #Max Cardinality Bipartite Cliques[3]	• $\delta = n - 2$ • Δ constant • planar	• bipartite [PB83][2]	• bipartite • $\delta = n - 5$

Remarks.
• Δ (resp., δ) denotes an upper (resp., lower) bound on the maximum (resp., minimum) vertex degree.
• k is always a constant.
• n denotes the number of vertices, except when referring explicitly to bipartite graphs, in which case it refers to the number of vertices on each side.

[1] Greenhill [Gre99] has improved this to $k = 3$.
[2] These results are not stated explicitly in [Val79b, PB83], but follow readily from the results and reductions in those papers.
[3] In general, any result for independent sets also holds for cliques *in the complement graph*, and conversely (cf., Proposition 3.1). An analogous relationship holds between independent sets in bipartite graphs and bipartite cliques in the bipartite complement graph. (cf., Proposition 3.2). For brevity, we do not enumerate all the results we obtain in this way.

Table 2: $\#\mathcal{P}$-completeness Results for Counting Satisfying Assignments in Restricted Classes of Formulae

Problem	Polynomial-time Solvable	Previous Results	This Paper
#SAT	• monotone 2-CNF, each variable appears at most twice [Rot96] • monotone 2-CNF, each variable fails to appear in a clause with only a constant number of other variables • acyclic monotone 2-CNF [Rot96] • acyclic Horn 2-CNF [Rot96] • Horn 2-CNF, each variable appears at most twice [Rot96]	• planar 3-CNF [HMRS98][1] • bipartite monotone 2-CNF [PB83] • Horn 2-CNF, each variable appears at most 3 times [Rot96] • CNF, each literal appears exactly once [BD97a] • monotone CNF, each variable appears at most twice [BD97a]	• planar bipartite monotone 2-CNF, each variable appears at most 4 times • monotone 2-CNF, each variable appears exactly k times, any $k \geq 5$[2]
#MINTERMS	• monotone 2-CNF, each variable appears at most twice • monotone 2-CNF, each variable fails to appear in a clause with only a constant number of other variables	• bipartite monotone 2-CNF [Val79b, PB83][3]	• planar bipartite monotone 2-CNF, each variable appears at most 5 times • monotone 2-CNF, each variable appears the same number of times
#MIN WEIGHT SAT	• monotone 2-CNF, each variable appears at most twice • monotone 2-CNF, each variable fails to appear in a clause with only a constant number of other variable	• bipartite monotone 2-CNF [Val79b, PB83][3] • planar monotone 2-CNF [HMRS98][3]	• planar bipartite monotone 2-CNF, each variable appears at most 3 times • monotone 2-CNF, each variable appears exactly k times, and $k \geq 4$

Remarks.

• Terms such as planar, acyclic, etc. refer to properties of the graph obtained from a CNF formula by having a vertex for each variable and connecting two vertices if they appear in the same clause.

• k is always a constant.

[1] In [HMRS98], the graph associated to a CNF formula is the bipartite graph on variables and clauses in which a clause is connected to the variables it contains. For 2-CNF, the planarity of this graph is equivalent to the planarity of the graphs we consider.

[2] Greenhill [Gre99] has improved this to $k = 3$.

[3] These results are not stated explicitly in [Val79b, PB83, HMRS98], but follow readily from the results and reductions in those papers.

x_s, and the well-known closed form for the Fibonacci numbers (cf., [Tuc95, Sec. 7.3]) can be used to show that their consecutive ratios x_s/x_{s+1} never repeat. (We formally prove all this in Lemma 6.3.)

In contrast to the previously known reduction from #PERFECT MATCHINGS to #MATCHINGS [Val79b], the above reduction only increases the maximum vertex degree by one. It also preserves other graph properties, such as bipartiteness and planarity. Moreover, this technique generalizes quite easily to other problems and graph properties (such as regularity). We used only a few properties of chains and matchings in the reduction:

1. The number of matchings in the graphs G_s is related to matchings in the original graph G via polynomial evaluation.
2. The evaluation points can be expressed in terms of the number of matchings in a chain of length s (and the number in which one end vertex is not matched).
3. The evaluation points can be computed efficiently.
4. The evaluation points do not repeat.

We will see by inspection that analogues of the first two conditions still hold when we replace matchings with a variety of other problems, and when chains are replaced with other gadgets. For the last two conditions, we will use gadgets possessing a repetitive structure, so that the evaluation points satisfy simple linear recurrence relations. These recurrences will certainly allow the evaluation points to be computed efficiently, but we are left with proving that they do not repeat. To this end, in Section 6 we prove the following lemma, which gives general conditions under which (ratios of) sequences defined by 2×2 linear recurrences do not repeat.

Lemma 6.2 *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:*

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

Greenhill [Gre99] has observed that when the coefficients A,B,C,D are all positive, Conditions 2–6 are guaranteed to hold, so only the first and last must be checked.

This lemma, with the approach outlined above, gives an almost mechanical way to find reductions that preserve properties such as sparsity and regularity. We will refer to this method as the *Fibonacci technique*, in reference to its simplest incarnation described above. The reductions we obtain from the Fibonacci technique have some interesting features not present in many previous interpolation-based reductions, such as those in [Val79b]. First, our interpolation points are typically rapidly converging sequences of rational numbers (e.g. the consecutive ratios of Fibonacci numbers converge to the golden ratio), whereas previous methods often interpolated at distinct integer points, which seems difficult to do without losing special properties of the original graph. Second, we do not know how to reduce the number of oracle calls in these reductions 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 operations used by Valiant blow up the degree.

Of course, it is not enough to have reductions that preserve properties such as sparsity and regularity; we need initial hardness results for restricted classes of graphs from which to reduce. For this, we rely heavily on results and techniques from previous work, such as [DL92, PB83, Jer87, Val79b, Rot96]. In particular, to obtain results about planar graphs, we use the Fibonacci technique to refine the reduction of Jerrum [Jer87] so that properties such as sparsity and bipartiteness are preserved, and also extend his approach to problems involving vertex covers.

2 Related Work

Our results for counting in sparse bipartite graphs first appeared in the author's undergraduate thesis [Vad95], and the first version of this paper [Vad97] added our results for planar and regular graphs. Some of the related work, namely that of Luby and Vigoda [LV97] and Bubley and Dyer [BD97a, BD97b], was done subsequent to [Vad95], but independently of [Vad97]. We describe those works, together with more recent developments, under the heading "Subsequent Work." Throughout this section, we only discuss works that address the same counting problems as us. The reader is referred to [Wel93, Pap94, Vad95, Jer95] for more general surveys on the complexity of counting, and [Sin93, Kan94, MR95, Vad96, JS97] for approximate counting.

Previous work. In his seminal paper [Val79a], Valiant introduced the class $\#\mathcal{P}$ of counting problems and proved that counting perfect matchings in bipartite graphs is $\#\mathcal{P}$-complete. In [Val79b], he showed that a number of other counting and reliability problems are $\#\mathcal{P}$-complete, including the unrestricted versions of most of the problems we study in this paper. Some of these problems, such as #Vertex Covers, were shown to remain hard in bipartite graphs by Provan and Ball [PB83], who also proved hardness results for reliability problems. Provan [Pro86] obtained restricted-case results (acyclic planar graphs of maximum degree 3) for some reliability problems, but not the problems that we investigate. Jerrum [Jer87] showed that counting matchings in planar graphs is $\#\mathcal{P}$-complete, in striking contrast to perfect matchings which can efficiently counted in planar graphs via algorithms due to Fisher, Kasteleyn, and Temperley [Fis61, Kas63, TF61]. Broder [Bro86] proved that counting perfect matchings in bipartite graphs of minimum vertex degree at least $n/2$ is $\#\mathcal{P}$-complete. Dagum and Luby [DL92] obtained even stronger results, showing that counting perfect matchings remains hard even in k-regular and $(n-k)$-regular bipartite graphs, for any constant $k \geq 3$.

From the start, the $\#\mathcal{P}$-completeness of counting satisfying assignments to a propositional formula was seen to follow immediately from parsimonious versions of Cook's reduction [Val79a]. Valiant [Val79b] proved that the problem remained just as hard in the dramatically restricted case of monotone 2-CNF, which was restricted further to bipartite monotone 2-CNF by Provan and Ball [PB83]. Roth [Rot96] showed that counting satisfying assignments is $\#\mathcal{P}$-complete even in 2-CNF Horn formulae in which each variable appears at most 3 times, along with giving a number of polynomial-time algorithms to count satisfying assignments in other restricted types of 2-CNF formulae. Hunt, Marathe, Radhakrishnan, and Stearns [HMRS98] showed that counting satisfying assignments to planar 3-CNF formulae is $\#\mathcal{P}$-complete, and gave a number of other counting problems that are hard in planar graphs, including counting minimum cardinality vertex covers.

The general theory of approximate counting was developed in work of Stockmeyer [Sto85], Karp and Luby [KLM89], and Jerrum, Valiant, and Vazirani [JVV86]. The first positive result for approximate counting that relates to our work is due to Karp and Luby [KLM89], who gave gave a polynomial-time algorithm for approximately counting satisfying assignments to a DNF formula. After that, most of the positive results on approximate counting have come via the theory of "rapidly mixing Markov chains." In the first dramatic application of this approach to approximating a $\#\mathcal{P}$-complete counting problem, Jerrum and Sinclair [JS89] analyzed a Markov chain proposed by Broder [Bro86] and thereby showed that it is possible to approximately count the number of perfect matchings in a graph of minimum vertex degree at least $n/2$ in polynomial time. They also gave an algorithm for approximately counting all matchings in an arbitrary graph. Their perfect matching algorithm was simplified by Dagum and Luby [DL92], who thereby obtained algorithms for approximately counting perfect matchings in αn-regular bipartite graphs for any constant α (and in fact a more general class of graphs).

For the counting problems we consider, the only previous inapproximability results involved approximately counting independent sets (equivalently, vertex covers or cliques) in general graphs [Sin93, Zuc96]. Specifically, Sinclair [Sin93] used the 'blow-up' technique introduced in [JVV86] to show that it is $\mathcal{NP}$-hard to approximate the number of independent sets in a graph, even up to an approximation factor of $2^{n^{1-\varepsilon}}$, for any constant $\varepsilon > 0$. This result directly translates to the $\mathcal{NP}$-hardness of approximately counting the number of satisfying assignments to a monotone 2-CNF formula to within a factor of $2^{n^{1-\varepsilon}}$ [Rot96] (cf., Proposition 3.1). Zuckerman [Zuc96], using techniques from the theory of probabilistically checkable proofs (PCP) [FGL+96, AS98, ALM+98], showed that it is hard to approximate arbitrarily iterated logarithms of the number of independent sets in a graph unless $\mathcal{NP}$ has slightly superpolynomial-time randomized algorithms.

Subsequent work. Since this work was done, there has been a substantial improvement in our understanding of counting independent sets (equivalently, vertex covers) in sparse graphs. With respect to exact counting, our results left a gap between the easy and hard cases; specifically, we showed that counting independent sets in graphs of maximal degree 4 is $\#\mathcal{P}$-complete, whereas the problem is polynomial-time solvable in graphs of maximal degree 2. For regular graphs, our hardness result only held for degrees ≥ 5. Greenhill [Gre99] has closed these gaps, showing that counting independent sets in 3-regular graphs is $\#\mathcal{P}$-complete. We note that her hardness result uses ours as a starting point, and also makes use of (generalizations of) our Fibonacci technique.

For approximately counting independent sets in sparse graphs, almost nothing was known at the time of this work. There were no polynomial-time approximation algorithms and no inapproximability results other than Sinclair's [Sin93] and Zuckerman's [Zuc96] results for unrestricted graphs and our result about counting maximum cardinality independent sets. Luby and Vigoda [LV97] have remedied this situation in both respects. First, using the Markov chain approach, they have given a polynomial-time algorithm to approximately count independent sets in graphs of maximal degree at most 4. (Extensions and improvements can be found in [DG97, LV99, RT98].) Second, combining a blow-up technique of [Sin93] with PCP-based inapproximability results [PY91, AS98, ALM+98], they proved that for some (large) constant Δ, approximately counting independent sets in graphs of maximal degree Δ is $\mathcal{NP}$-hard. Using a more sophisticated reduction and results in [Hås97], Dyer, Frieze, and Jerrum [DFJ99] reduce the degree for this inapproximability result to $\Delta = 25$, and give evidence that the Markov chain approach is unlikely to work for any $\Delta \geq 6$. These results suggest that the PCP Theorem, which has yielded many inapproximability results for optimization problems, may also be the right starting point for proving hardness of approximate counting.

Bubley and Dyer [BD97b] have considered the problem of counting independent sets of a given size s in a graph of maximal degree Δ. Our results on counting maximum cardinality independent sets imply that the exact (resp., approximate) version of this problem is $\#\mathcal{P}$-complete (resp., $\mathcal{NP}$-hard) even when $\Delta = 3$, if there is no restriction on s. They show that in fact the approximate counting problem can be solved in polynomial time for $s < n/2(\Delta+1)+1$, whereas the exact counting problem remains $\#\mathcal{P}$-complete under this restriction.

In another work, Bubley and Dyer [BD97a] have proven some new results about counting satisfying assignments in restricted classes of formulae. Instead of looking at 2-CNF formulae (as we do, and as happens when translating results about independent sets), they do not restrict the number of variables per clause, but only allow each variable to appear at most twice. They have shown that it is possible to efficiently approximate the number of satisfying assignments to such formulae. On the other hand, they show that exactly counting satisfying assignments remains $\#\mathcal{P}$-complete in CNF formula in which each literal appears exactly once, and monotone CNF formulae in which variable appears at most twice. They also relate these versions of #SAT to counting "sink-free orientations" in directed graphs.

3 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 $\#\mathcal{P}$. Below, we informally review some basic definitions. For a more detailed discussion of $\#\mathcal{P}$, the reader is referred to any of [Wel93, Vad95, Jer95, Pap94].

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, x should be thought of 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 standard Karp reductions (i.e., polynomial-time many-one 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, following [Val79a], we consider a problem Π to be as hard as a problem Γ iff Γ can be solved by a polynomial-time algorithm with an oracle for Π, and we denote this by $\Gamma \propto \Pi$. Such a reduction is known as a **Cook reduction** (or **polynomial-time Turing reduction**), 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 polynomial-time 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.

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 **(maximum) degree** of G is the maximum number of edges incident to any vertex, and the **minimum degree** of G is defined similarly. A **vertex covers** in G is a subset S of V such that every edge in E has at least one endpoint in S. An **independent sets** in G is a subset S of V such that no two vertices in S are connected by an edge in E. A **cliques** 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 intimately related objects. Their relationship is formalized by the following proposition:

Proposition 3.1 *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. □

We will also be examining the complexity of these problems in bipartite graphs. However, the study of cliques in bipartite graphs is not very interesting, as the only cliques are edges. So, for a bipartite graph $G = (V, E)$, we will instead look at **bipartite cliques**, which are subsets $S \subset V$ of vertices which can be partitioned $S = S_1 \cup S_2$ so that $S_1 \times S_2 \subset E$. To obtain an analogoue of Proposition 3.1, we say that bipartite graphs $G = (V, E)$ and $H = (V, F)$ are **bipartite complements** if V can be partitioned $V = V_1 \cup V_2$ such that $E \subset V_1 \times V_2$ and $F = V_1 \times V_2 \setminus E$. Note that a bipartite complement of a graph can be found in polynomial time, and is unique if the graph is connected. Proposition 3.1 has the following bipartite analogue.

Proposition 3.2 *Let bipartite graphs $G = (V, E)$ and $H = (V, F)$ be bipartite complements. Then $S \subset V$ is an independent set in G iff it is a bipartite clique in H.*

Proof By the definitions. □

The above propositions 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 primarily on the vertex cover problem.

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 3.1, 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 **matchings** in an undirected graph $G = (V, E)$ is a set $M \subset E$ of edges, no two of which share an endpoint. A **perfect matchings** 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 3.3 *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.*

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 3.1. □

One of the 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 3.1 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.

There are several reasons for studying the problems described above. 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 important problems in other areas. As discussed in the introduction, several problems in Artificial Intelligence are equivalent to counting satisfying assignments to a propositional formula [Orp90, Pro90, Rot96]. In addition, the counting problems we consider arise naturally in statistical physics; specifically, counting matchings amounts to counting arrangements of monomer-dimer systems, and counting independent sets is tantamount to computing the partition function in the hard-core model of a gas (cf., [Har67, Jer87, LV99]).

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.

4 Formal Statement of Results

Before stating our negative results, we explain the entries in the "Polynomial-time Solvable" columns of Tables 1 and 2 for which no reference is given. The tractability of problems such as counting matchings, vertex covers, and their variants in graphs of maximal degree at most 2 follows from the simple structure of these graphs: The connected components of such graphs are cycles and chains, and objects such as matchings and vertex covers multiply across connected components, so it suffices to count them in cycles and chains. It is not hard to write down closed forms or recurrences which can be used to count these structures in chains and cycles. This is done explicitly in [Vad95]. The polynomial-time solvability of counting cliques in graphs of maximal degree at most Δ follows from the fact that the largest clique in such a graph is of size at most

Δ. Thus, one can exhaustively check the $\leq n^{\Delta}$ possibilities in polynomial time. Similarly, a planar graph cannot contain any clique of size ≥ 5, nor any bipartite clique involving at least 3 vertices on both sides of the partition, as these structures are nonplanar (cf., [Tuc95, Sec. 1.4]). Finally, as noted earlier, the only cliques in a bipartite graph are singletons and pairs. All the other entries follow from the above observations via Propositions 3.1 and 3.2.

We now state our negative results. Below, the counting problems are named as follows: the beginning of the name indicates any restrictions on the input graph or formula, and the end of the name contains the structures to be counted. A prefix of $k\Delta$ indicates that the maximum degree of the graph is at most k. An example of a problem denoted this way is:

#7Δ-PLANAR BIPARTITE MAXIMAL MATCHINGS
Input: A planar bipartite graph G of degree ≤ 7.
Output: he number of maximal matchings in G.

Theorem 4.1 *The following problems are #$\mathcal{P}$-complete (except for Problem 8, which is #$\mathcal{P}$-hard*[1]*):*
1. #4Δ-BIPARTITE MATCHINGS
2. #6Δ-PLANAR BIPARTITE MATCHINGS
3. #k-REGULAR MATCHINGS*, any fixed* $k \geq 5$
4. #5Δ-BIPARTITE MAXIMAL MATCHINGS
5. #7Δ-PLANAR BIPARTITE MAXIMAL MATCHINGS
6. #PLANAR BIPARTITE VERTEX COVERS
7. #3Δ-PLANAR BIPARTITE MINIMUM CARDINALITY VERTEX COVERS
8. #k-REGULAR MINIMUM CARDINALITY VERTEX COVERS*, any fixed* $k \geq 4$
9. #4Δ-PLANAR BIPARTITE VERTEX COVERS
10. #k-REGULAR VERTEX COVERS*, any fixed* $k \geq 5$
11. #5Δ-PLANAR BIPARTITE MINIMAL VERTEX COVERS
12. #REGULAR MINIMAL VERTEX COVERS

In the following corollary, we use the same naming conventions as above, with some additional ones. When BIPARTITE CLIQUES appears, it refers to counting bipartite cliques in bipartite graphs. A prefix of $k\delta$ means that the minimum vertex degree is at least k. n refers to the number of vertices in the graph, except when the input is restricted to be bipartite, in which case n is the number of vertices on each side of the bipartition.

For problems involving satisfying assignments, 2SAT denotes the problem of counting the satisfying assignments of the input formula F, MINTERMS the problem of counting the minterms of F (i.e., the number of clauses in the smallest DNF formula equivalent to F), and MIN WEIGHT 2SAT the problem of counting satisfying assignments with the fewest variables set to true. A prefix of $k\Delta$ (resp., k-REGULAR) indicates that each variable appears at most (resp., exactly) k times. A *bipartite* 2-CNF formula is one in which the variables can be partitioned into two sets such that no clause contains two variables from the same set. For example:

#3Δ-PLANAR BIPARTITE MONOTONE MIN WEIGHT 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 3 times.
Output: The number of satisfying assignments to F with the fewest variables set to 'true'.

Corollary 4.2 *The following problems are #$\mathcal{P}$-complete (except for Problems 2, 9 and 17, which are #$\mathcal{P}$-hard):*
1. #3Δ-PLANAR BIPARTITE MAXIMUM CARDINALITY INDEPENDENT SETS

[1]Problem 8 is not likely to be in #$\mathcal{P}$, because 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].

In contrast, Problem 7 is in #$\mathcal{P}$ because in *bipartite graphs*, the size of the minimum cardinality vertex cover equals the size of the maximum cardinality matching. The latter quantity can be found using any of the standard maximum cardinality matching algorithms. See [Pap94, Problem 9.5.25].

2. #k-Regular Maximum Cardinality Independent Sets, *any fixed* $k \geq 4$
3. #4Δ-Planar Bipartite Independent Sets
4. #k-Regular Independent Sets, *any fixed* $k \geq 5$
5. #5Δ-Planar Bipartite Maximal Independent Sets
6. #Regular Maximal Independent Sets
7. #$(n-3)\delta$-Maximum Cardinality Cliques
8. #$(n-3)\delta$-Maximum Cardinality Bipartite Cliques
9. #$(n-k)$-Regular Maximum Cardinality Cliques, *any fixed* $k \geq 4$
10. #$(n-4)\delta$-Cliques
11. #$(n-4)\delta$-Bipartite Cliques
12. #$(n-k)$-Regular Cliques, *any fixed* $k \geq 5$
13. #$(n-5)\delta$-Maximal Cliques
14. #$(n-5)\delta$-Maximal Bipartite Cliques
15. #Regular Maximal Cliques
16. #3Δ-Planar Bipartite Monotone Min Weight 2Sat
17. #k-Regular Monotone Min Weight 2Sat, *any fixed* $k \geq 5$
18. #4Δ-Planar Bipartite Monotone 2Sat
19. #k-Regular Monotone 2Sat, *any fixed* $k \geq 5$
20. #5Δ-Planar Bipartite Minterms
21. #Regular Minterms

The following proposition contains our inapproximability result. The prefix of f-approx indicates the problem of solving the given counting problem within a multiplicative approximation factor of f. Though we have adopted Cook reductions as our notion of reducibility, the following results are actually proved via Karp reductions to 'gap promise problems' (cf., [BGS98]).

Proposition 4.3 *The following problems are $\mathcal{NP}$-hard for every $\epsilon > 0$:*
1. $2^{n^{1-\epsilon}}$-Approx #3Δ-Minimum Cardinality Vertex Covers
2. $2^{n^{1-\epsilon}}$-Approx #3Δ-Maximum Cardinality Independent Sets
3. $2^{n^{1-\epsilon}}$-Approx #$(n-3)\delta$-Maximum Cardinality Cliques
4. $2^{n^{1-\epsilon}}$-Approx #3Δ-Min Weight Monotone 2Sat

5 The Reductions

We state three facts about polynomial interpolation here that we will use repeatedly in our reductions. 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. (If K is a degree d extension, elements of K are represented by polynomials of degree $\leq d-1$ with rational coefficients, and arithmetic is done modulo some irreducible polynomial defining K.) 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. We use a **dense** representation of polynomials, which means that if some monomial $x_1^{d_1} \cdot x_2^{d_2} \cdots x_n^{d_n}$ has a nonzero coefficient in f, then the coefficients of all smaller monomials (i.e., any $x_1^{e_1} \cdots x_n^{e_n}$ such that $e_i \leq d_i$ for all i) must be included when computing $\|f\|$. (Zero coefficients count as one bit). For a rational function $q = f/g$, let $\|q\| = \|f\| + \|g\|$.

Fact 5.1 *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 5.2 *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 5.3 *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 5.1) 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. □

Proof (of Fact 5.2) Define, for each $0 \leq r \leq n$, $g_r(y) = \sum_{i,j} a_{ij} x_r^i y^j$. 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 5.1. 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 5.1, we can recover the coefficients of f. □

Proof (of Fact 5.3) 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 $\sum_{i=0}^{d-1} a_i \alpha^i \leq \sum_{i=0}^{d-1} (\alpha - 1)\alpha^i = \alpha^d - 1 < \alpha^d$, 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. □

Proof (of Theorem 4.1) 1. #4Δ-Bipartite Matchings
The reduction given in the Introduction reduces #Perfect Matchings to #Matchings while only increasing the degree by 1 and preserving bipartiteness. Dagum and Luby [DL92] have proven the former to be $\#\mathcal{P}$-complete in 3-regular bipartite graphs; our result follows. (The facts about matchings in chains and Fibonacci sequences needed for the reduction are given by Lemma 6.3.)

2. #6Δ-Planar Bipartite Matchings
The starting point for this proof is the work of Jerrum [Jer87] 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 1 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 $\boldsymbol{w(M)} = \prod_{v \notin C(M)} w(v)$, *i.e.* the product of the weights of all vertices *not* covered by M; for a perfect matching M, $w(M) = 1$. 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 bipartite planar graphs of bounded degree is hard.

2a. #3-Regular Perfect Matchings $\propto$ #5Δ-2λ-Planar Weighted Matchings
Jerrum [Jer87] 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.

2b. #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.) Straightforward calculations show the following:

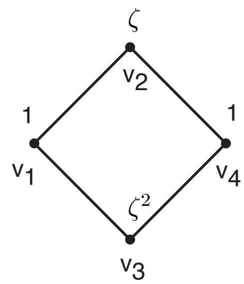

Figure 1: The graph H in Reduction 2b.

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

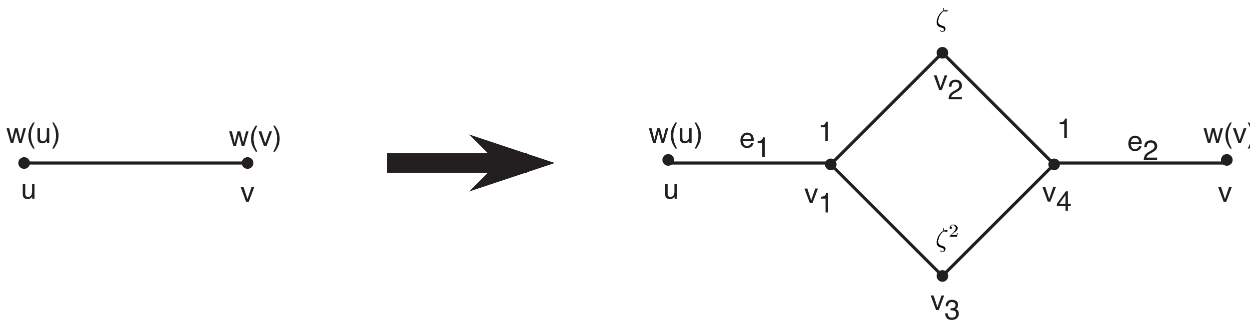

Figure 2: The transformation of Reduction 2b.

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 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 $k = 4$.

2c. #$k\mu$-Planar Bipartite Weighted Matchings $\propto$ #$(k-1)\mu$-Planar Bipartite Weighted Matchings

Jerrum [Jer87] gives a reduction which removes weights one by one, but his reduction blows up the degree. To replace his reduction, we use the Fibonacci technique.

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 $W(G_s) = (x_{s+1}/\alpha)^m \sum_{i=0}^m A_i (\alpha x_s / x_{s+1})^i$. As in original application of the Fibonacci technique, we can recover the coefficients A_i with $m+1$ oracle calls and polynomial interpolation, using Lemma 6.3 to verify that the evaluation points are distinct. Then it is easy to compute $W(G) = \sum_{i=0}^m A_i$, as desired.

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 #0μ-Planar Bipartite Weighted Matchings is exactly #6Δ-Planar Bipartite Matchings.

3. #k-Regular Matchings, any fixed $k \geq 5$

We reduce from #$(k-2)$-Regular Bipartite Perfect Matchings, which was shown to be #$\mathcal{P}$-complete by Dagum and Luby [DL92]. Let H be the complete graph on $k+1$ vertices with one edge removed. Consider the graph L_s^k formed by taking $s+1$ disjoint copies of H, labelled $H_0, \ldots, H_s$, and attaching y_i to x_{i+1} for each $0 \leq i < s$, where x_i and y_i are the two vertices in H_i of degree $k-1$. (See Figure 3.)

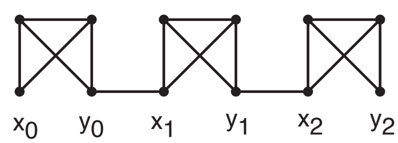

Figure 3: The graph L_2^3 in Reduction 3.

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 6.4, we can compute x_s, y_s, and z_s in polynomial time and the sequence $(x_s+4y_s+3z_s)/(x_s+2y_s+z_s) = 1+2(y_s+z_s)/(x_s+2y_s+z_s) =$ never repeats. So, with $n+1$ oracle calls and interpolation, we can recover A_0, the number of perfect matchings in G.

4. #4Δ-Bipartite Matchings $\propto$ #5Δ-Bipartite Maximal Matchings

The reduction we use is from [PB83], though they use it for vertex cover problems. 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.

5. #6Δ-Planar Bipartite Matchings $\propto$ #7Δ-Planar Bipartite Maximal Matchings.

Notice that Reduction 4 preserves planarity, so it applies here, too.

6. #Planar Bipartite Vertex Covers

This result will come via a sequence of reductions beginning with #Vertex Covers, whose #$\mathcal{P}$-completeness follows immediately from Lemma 3.3 and the #$\mathcal{P}$-completeness of #Matchings. In the spirit of Reduction 2, the intermediate problems involve a generalization of vertex covers which we call an **edge-weighted sets**. 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 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 EW(G) is simply the number of vertex covers in G, so we have indeed generalized #Vertex Covers. 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 Covers 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: EW(G).

We now reduce #Vertex Covers to this problem.

6a. #Vertex Covers $\propto$ #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 $G_{s,t} = (V_{s,t}, E_{s,t})$ 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 6.3. 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 6.3), 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 6.3 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 5.2, 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 $EW(G)$.

6b. #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.

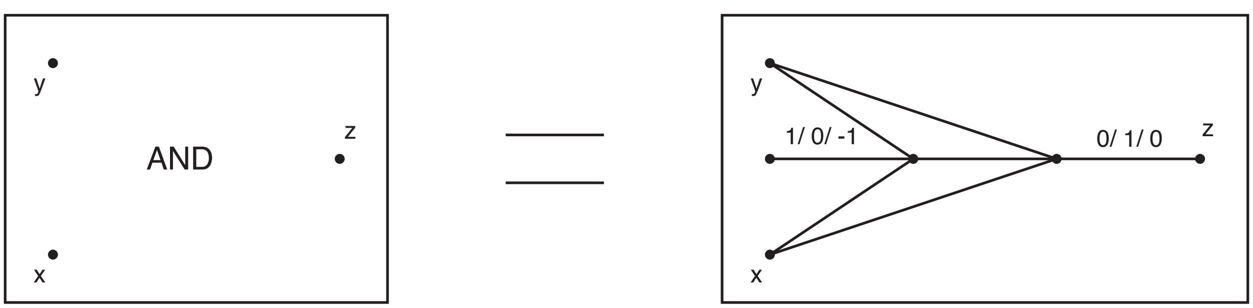

Figure 4: The AND gadget in Reduction 6b.

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 take on the same value and an edge labelled 1/0/0

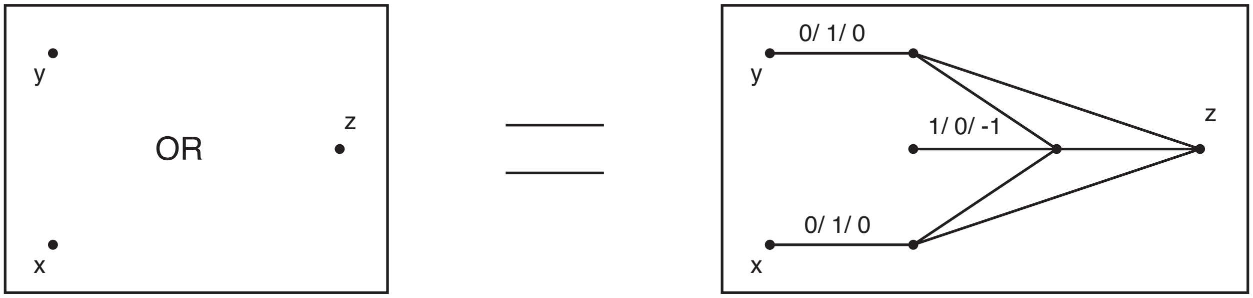

Figure 5: The OR gadget in Reduction 6b.

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.

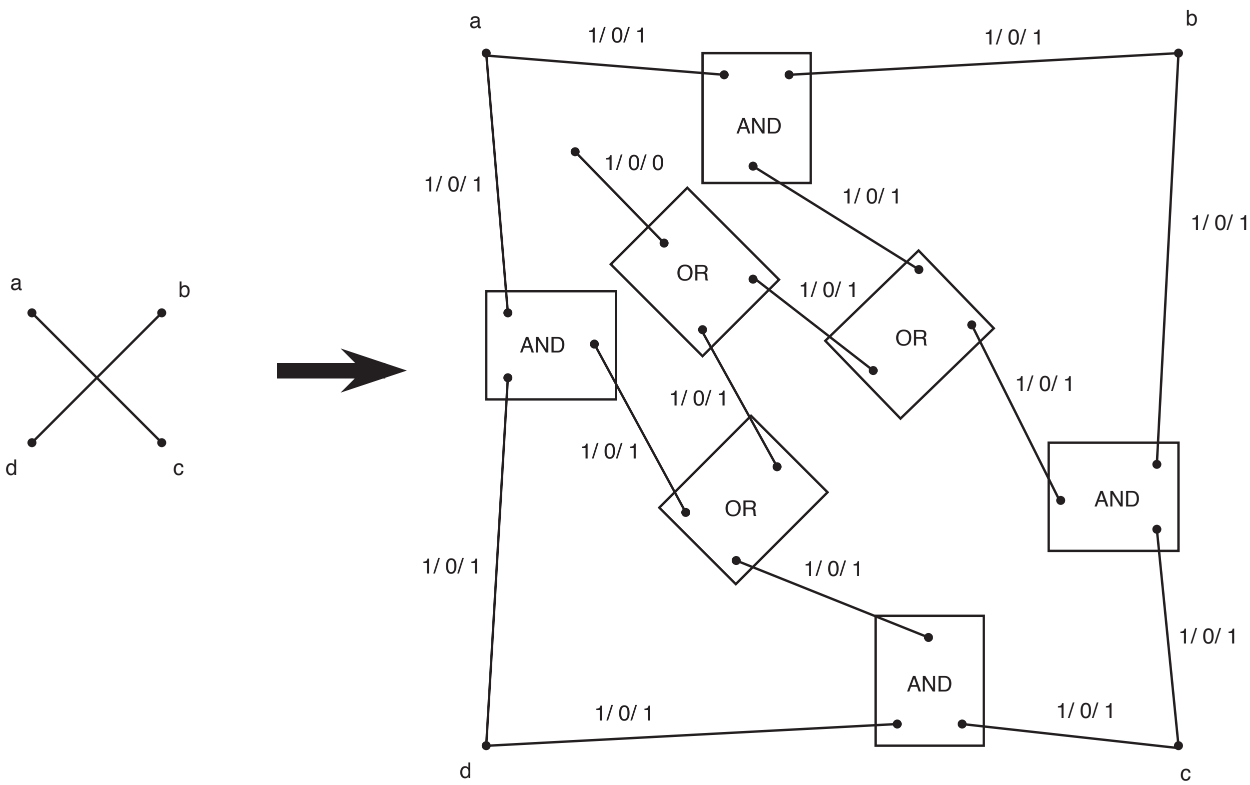

Figure 6: Replacing crossings in Reduction 6b.

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.

6c. $\#k\lambda$-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

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

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 $0\le i\le s, 0\le j\le t$.

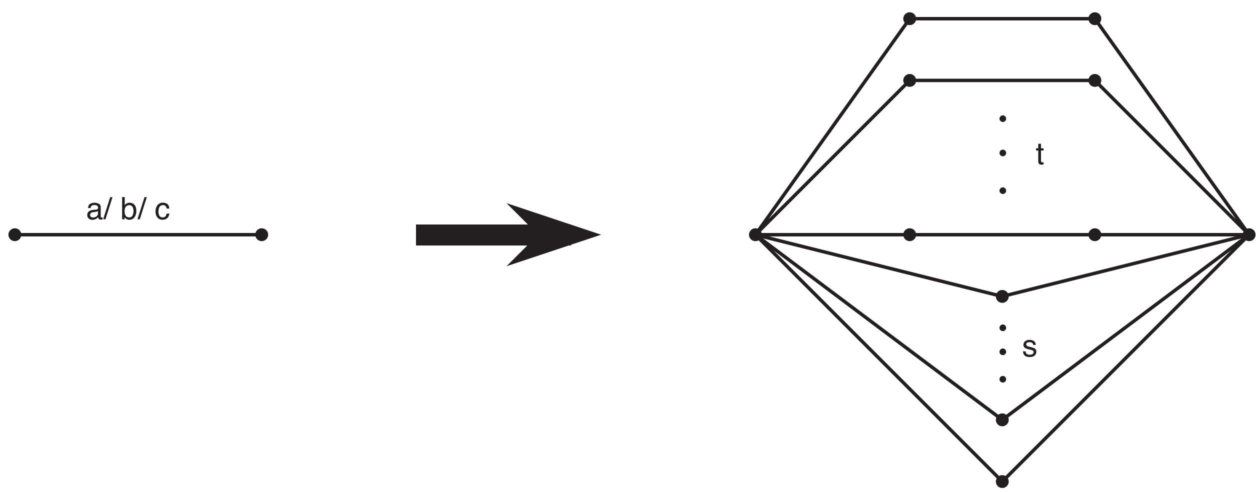

Figure 7: The transformation of Reduction 6c.

Let S be a subset of V. For each edge $e\in L$ doubly covered by S, there are 2^s3^t (weighted) ways to extend S using vertices of V_e. For each edge $e\in L$ 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

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

then

$$\mathrm{EW}(G_{s,t})=\sum_{i,j}A_{ij}(2^s3^t)^i(2^t)^j.$$

By Fact 5.2, if we can compute $\mathrm{EW}(G_{s,t})$ for $t=0,\dots,\ell$, $s=0,\dots,\ell$, we can recover the coefficients A_{ij} and thereby compute

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

Notice that $\#0\lambda$-PLANAR EDGE-WEIGHTED SUM is exactly PLANAR VERTEX COVERS.

6d. #PLANAR VERTEX COVERS $\propto$ #PLANAR BIPARTITE VERTEX COVERS
Observe that the reduction of Provan and Ball [PB83] from #VERTEX COVERS to #BIPARTITE VERTEX COVERS preserves planarity.

7. #PLANAR BIPARTITE VERTEX COVERS $\propto$ $\#3\Delta$-PLANAR BIPARTITE MINIMUM CARDINALITY VERTEX

Covers

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.

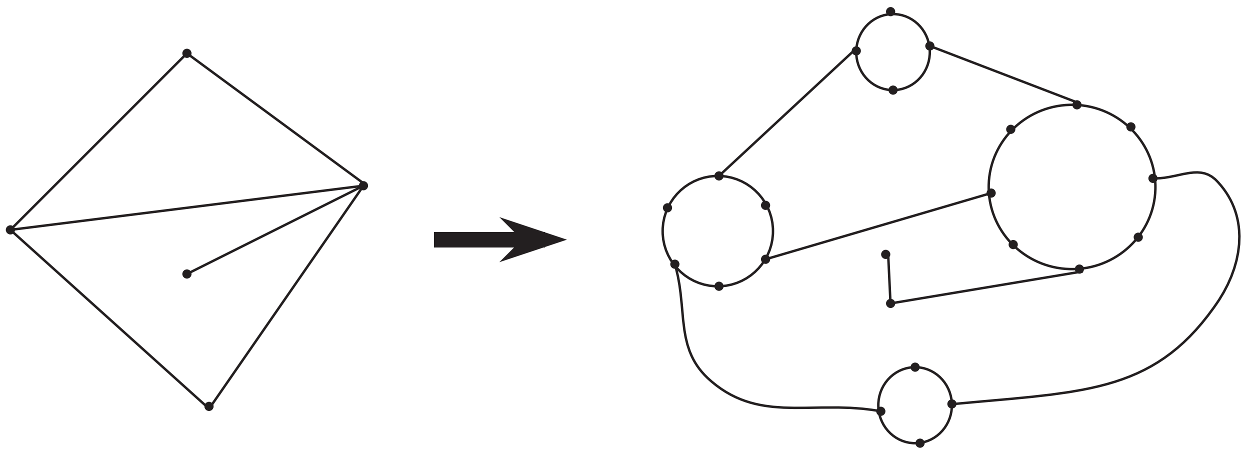

Figure 8: The tranformation of Reduction 7.

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.

8a. #k-Regular Bipartite Perfect Matchings $\propto$ #$(2k-2)$-Regular Minimum Cardinality Vertex Covers, any fixed $k \geq 3$

This follows immediately from Lemma 3.3, noting that the line-graph of a k-regular graph is a $(2k-2)$-regular graph.

8b. #$(k-1)$-Regular Minimum Cardinality Vertex Covers $\propto$ #k-Regular Minimum Cardinality Vertex Covers, 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^k 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 $k-1$. See Figure 9.

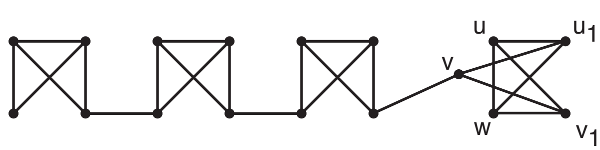

Figure 9: The graph H_3^3 in Reduction 8b.

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^k_{2s}. Let the vertex of degree $k-1$ in the ith copy of H^k_{2s} be labelled v_i. We can form a k-regular graph G_s by connecting u_i to v_i for each i. Lemma 6.5 tells us that there are minimum cardinality vertex covers (mcvc's) in the ith copy of H^k_{2s} 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^k_{2s}. If G has N mcvc's and they are 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^k_{2s} containing the vertex of degree $k-1$ and y_s is the number not containing the vertex of degree $k-1$. With a single oracle call, we obtain the evaluation of the polynomial $f(x) = Nx^c$ at the point $z_s = 1 + y_s/x_s$, which equals $1 + ((k+1)s + 4)/(k-1)$ by Lemma 6.5. 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. (In fact, Reduction 8a produces instances G in which we know c, so actually only one oracle call is necessary here.)

9. #3Δ-Planar Bipartite Minimum Cardinality Vertex Covers $\propto$ #4Δ-Planar Bipartite Vertex Covers
This is a standard application of the Fibonacci technique, almost identical to Reduction 1: Form G_s by attaching chains of length s to each vertex of the input graph G for $s = 0, \ldots, n$. The number of vertex covers in G_s is essentially the evaluation of a polynomial whose coefficients are the number of vertex covers in G of each size. By Lemma 6.3, these evaluation points are consecutive ratios of Fibonacci numbers, which do not repeat, so by interpolation we can recover the number of minimum cardinality vertex covers in G.

10a. #4-Regular Minimum Cardinality Vertex Covers $\propto$ #5-Regular Vertex Covers
This is another application of the Fibonacci technique. Let G be any 4-regular graph on n nodes. As in Reduction 8b, for $0 \leq s \leq n$, form a 5-regular graph G_s by attaching n disjoint copies of the gadget H^5_s defined in that reduction. We recover the number of minimum cardinality vertex covers in G by polynomial interpolation, using Lemma 6.6 to guarantee that the evaluation points are distinct.

10b. #$(k-2)$-Regular Minimum Cardinality Vertex Covers $\propto$ #k-Regular Vertex Covers
This yet another application of the Fibonacci technique, with slightly different gadgets. Let H be the complete graph on $k+1$ vertices with two edges incident to some vertex v removed. Consider the sequence of graphs $\{I^k_n : n \geq 0\}$ defined as follows: I^k_0 is the complete graph on $k+1$ vertices with a single edge removed. I^k_{n+1} is formed by taking the disjoint union of I^k_n 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^k_n. Let $(I^k_n)^+$ be the graph formed by adding to I^k_n a new vertex p and connecting p to the two vertices of degree $k-1$ in I^k_n.

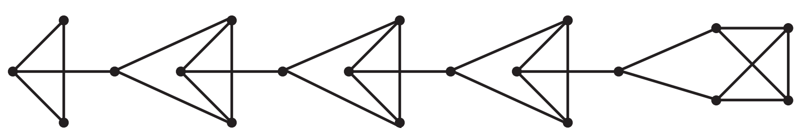

Figure 10: The graph I^3_4 in Reduction 10b.

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^k_s. Let the vertices of degree $k-1$ in the ith copy of I^k_s 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. By Lemma 6.7 and polynomial interpolation, the number of minimum cardinality vertex covers in G can be recovered from the number of vertex covers in G_s for $s = 0, \ldots, n$.

11. #4Δ-Planar Bipartite Vertex Covers $\propto$ #5Δ-Planar Bipartite Minimal Vertex Covers
This reduction is identical to Reduction 4.

12. #5-Regular Vertex Covers $\propto$ #Regular Minimal Vertex Covers
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 named $H_{i,j}$ for $1 \leq i \leq n, 1 \leq j \leq r$. 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. Notice that if G has A_i vertex covers of size i, G' has $\sum_{i=0}^{n} A_i (3^r)^i (2^r)^{n-i}$ minimal vertex covers. Dividing by 2^{rn}, we get the evaluation of $f(x) = \sum_{i=0}^{n} A_i x^i$ at $(3/2)^r$. If we choose $r = n$, we can, by Lemma 5.3, recover the coefficients of f in a single oracle call. The number of vertex covers in G is simply the sum of the coefficients. □

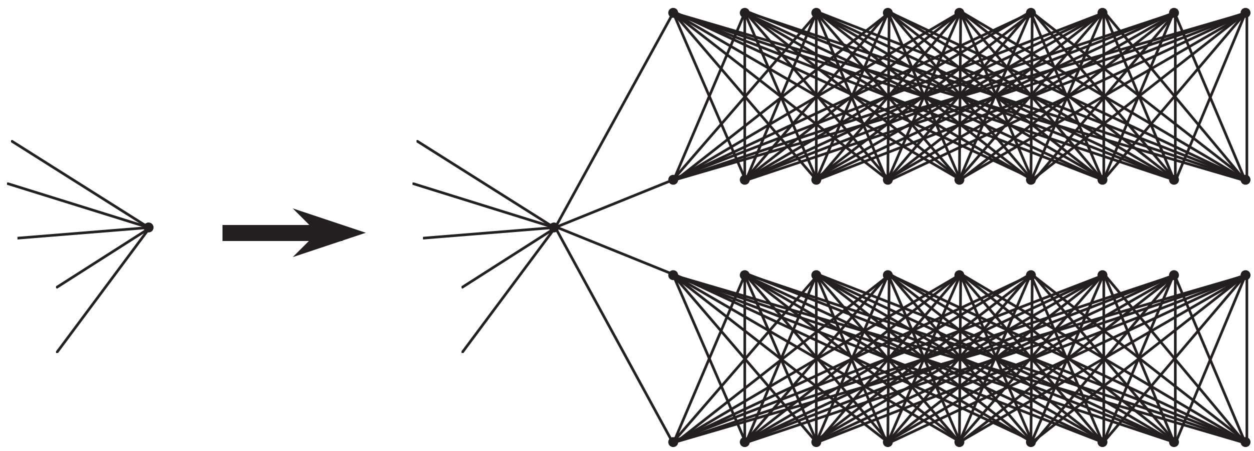

Figure 11: The transformation in Reduction 12 for $r = 2$.

Proof (of Corollary 4.2) These follow immediately from Theorem 4.1, Proposition 3.1, and Proposition 3.2. □

Proof (of Proposition 4.3) Here we only prove that the given problems are hard to approximate within a factor $2^{n^{1/2-\epsilon}}$. Proving inapproximability within $2^{n^{1-\epsilon}}$ is more involved, and details can be found in [Vad95] or [Vad97].

By Proposition 3.1, we may focus on vertex covers, and the other results follow. We reduce from $2^{n^{1-\epsilon}}$-Approx #Vertex Covers, which was shown to be $\mathcal{NP}$-hard by Sinclair [Sin93] (see also Roth [Rot96]). Note that, ignoring the planarity and bipartiteness conditions, Reduction 7 in the proof of Theorem 4.1 is a parsimonious reduction from #Vertex Covers to #3Δ-Minimum Cardinality Vertex Covers. That is, it transforms graphs G to graphs G' such that the number of minimum cardinality vertex covers in G' equals the number of vertex covers in G. Note that if G has n vertices, then the number of vertices in G' is $n' < 2n^2$, so an approximation within $2^{(n')^{1/2-\epsilon}}$ for G' gives an approximation within $2^{n^{1-\epsilon}}$ for G (for sufficiently large n). □

6 Proving that Sequences Don't Repeat

In this section, we develop general tools for proving that sequences defined by 2×2 linear recurrences do not repeat, and apply them to deduce that the interpolation points in our reductions are distinct.

Lemma 6.1 *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 $(ad-bc)(\alpha^m\beta^n - \beta^m\alpha^n) = 0$ iff $ad - bc = 0$ or $(\alpha/\beta)^{n-m} = 1$. □

Lemma 6.2 *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:*

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

Proof Let α and β be the eigenvalues of the matrix $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix} \right)$. By basic linear algebra, 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 $y_n = c\alpha^n + b\beta^n$, for some $a, b, c, d \in \mathbb{C}$. 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). □

As observed by Greenhill [Gre99], if A, B, C, and D are all positive (or if all are nonnegative and $A \neq D$), then only Conditions 1 and 7 must be checked in the above lemma. We now apply this lemma to the various sequences that arise in our reductions.

Lemma 6.3 *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 $n \geq 0$. The number of matchings (resp., vertex covers) in a chain of n vertices is F_n (resp., F_{n+1}). Moreover, $F_{n+1}/F_n \neq F_{m+1}/F_m$ for any $n \neq m$.*

Proof Let x_n be the number of matchings in a chain of n vertices. Given a chain $C_n = v_1 - v_2 - \cdots - v_n$ with $n \geq 1$, the number of matchings in C_n in which v_1 is matched is x_{n-2} and the number in which v_1 is unmatched is x_{n-1}. Thus, $x_n = x_{n-1} + x_{n-2}$, which is the Fibonacci recurrence. Also note that $x_0 = 1$ and $x_1 = 1$. To obtain the result for vertex covers, observe that C_n is the line-graph of C_{n+1} and apply Lemma 3.3. The "moreover" part of the lemma follows from Lemma 6.1 with $A = B = C = x_0 = y_0 = 1$ and $D = 0$. □

Lemma 6.4 *For $k \geq 4$, let L_s^k be the graph defined in Reduction 3 of the proof of Theorem 4.1. Let v and w be the vertices in L_s^k of degree $k - 1$. Let x_s be the number of matchings in L_s^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 x_s, y_s, and z_s can be computed 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:

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

It is easy to see that y_s and z_s satisfy the same recurrence relations:

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

The initial conditions are

$$\begin{aligned} z_0 &= M_{k-1} \\ y_0 &= M_k - z_0 = M_k - M_{k-1} \\ x_0 &= M_{k+1} - 2y_0 - 2x_0 = kM_{k-1} - M_k \end{aligned}$$

We can compute x_s, y_s, and z_s in polynomial time using the above recurrences. Because we have three sequences here, we cannot apply Lemma 6.2 directly. However, the proof here is nearly identical to the one of Lemma 6.2, 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 6.2; of these, Conditions 2–6 are automatically satisfied since the recurrence coefficients are all positive. After simplification, Condition 1 becomes:

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

Dividing by M_{k-1}^2 and applying the quadratic formula, we can reformulate this condition as:

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

Moser and Wyman [MW55] have shown the following:[2]

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

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 $(c+e)(b+2d+f) - (d+f)(a+2c+e) \neq 0$. Using the initial conditions to solve for these values, this reduces to

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

[2]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.

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 6.5 *Fix k to be an odd integer ≥ 3. Let H_n^k be the graph defined in Reduction 8b of the proof of Theorem 4.1. 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^k 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 $H_{2(m+1)}^k$ and let v be the vertex of degree $k-1$ in H_{2m}^k. 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}^k to form $H_{2(m+1)}^k$ 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 $H_{2(m+1)}^k$ could only come from taking a mcvc on H_{2m}^k, 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}^k that contains v, there are $k-1$ mcvc's in $H_{2(m+1)}^k$ containing v' and $k+1$ mcvc's not containing v'. For each mcvc of H_{2m}^k 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 6.6 *Fix k to be an odd integer ≥ 3. Let H_n^k be the graph defined in Reduction 8b of the proof of Theorem 4.1. 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 The sequences x_s and y_s satisfy the following recurrences:

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

with initial conditions $x_0 = (3k+3)/2$ and $y_0 = 3$. Conditions 1 and 7 of Lemma 6.2 are

$$\begin{aligned} -k+1 &\neq 0 \\ (9k-9)/2 &\neq 0 \end{aligned}$$

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

Lemma 6.7 *Fix k to be an integer ≥ 3. Let $(I_n^k)^+$ be the graph defined in Reduction 10b of the proof of Theorem 4.1. 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 The sequences x_s and y_s satisfy the following recurrences:

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

with initial conditions $x_0 = k + 3$ and $y_0 = k$. Conditions 1 and 7 of Lemma 6.2 are

$$\begin{aligned} -2k + 4 &\neq 0 \\ k^2 - 3k + 9 &\neq 0 \end{aligned}$$

It is easily verified that these hold for all integers $k \geq 3$. □

7 Conclusion

The study of counting and its computational complexity is both interesting and important. However, we only have a limited understanding of how the complexity of counting problems behaves in restricted cases. The results of this paper have improved the situation somewhat, 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. Recall that, subsequent to this work, Greenhill [Gre99] has closed this degree gap for counting independent sets, but other gaps still remain.

For approximate counting, the gaps between positive and negative results are even larger. For instance, Luby and Vigoda [LV97] have given a polynomial-time algorithm for approximately counting independent sets in graphs of degree 4, but the problem is only known to become $\mathcal{NP}$-hard at degree 25, as shown by Dyer, Frieze, and Jerrum [DFJ99]. An even larger gap in our knowledge is the long-standing open problem of approximately counting perfect matchings in general graphs. (Recall that this can be solved in polynomial time for dense graphs [JS89].) In the context of optimization problems, a substantial body of work has yielded numerous tight inapproximability results based on the the PCP Theorem (cf., [CK00]). There is a need to develop analogous general techniques for the inapproximability of counting problems, perhaps by designing PCP systems that are tailored for this purpose.

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.

I also thank Russ Bubley, Sergey Fomin, Michel Goemans, Catherine Greenhill, Vijay Vazirani, and the anonymous referees for numerous helpful suggestions and pointers.

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs and non-approximability — towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[Bro86] Andrei Z. Broder. How hard is to marry at random? (On the approximation of the permanent). In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 50–58, Berkeley, California, 28–30 May 1986. Erratum in Proceedings of 20th STOC.

[BD97a] 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.

[BD97b] Russ Bubley and Martin Dyer. Path coupling: a technique for proving rapid mixing in Markov chains. In *38th Annual Symposium on Foundations of Computer Science*, pages 223–231, Miami Beach, Florida, 20–22 October 1997. IEEE.

[CK00] Pierluigi Crescenzi and Viggo Kann. A compendium of NP optimization problems. `http://www.nada.kth.se/~viggo/wwwcompendium/wwwcompendium.html`, 2000.

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

[DFJ99] Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 210–217. IEEE, October 1999.

[DG97] Martin Dyer and Catherine Greenhill. On Markov chains for independent sets. To appear in *Journal of Algorithms*, December 1997.

[FGL$^+$96] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[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.

[Gre99] Catherine Greenhill. The complexity of counting colourings and independent sets in sparse graphs and hypergraphs. To appear in *Computational Complexity*, 1999.

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

[Hås97] Johan Håstad. Some optimal inapproximability results. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 1–10, El Paso, Texas, 4–6 May 1997.

[HMRS98] Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, and Richard E. Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27(4):1142–1167, August 1998.

[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.

[Jer95] Mark Jerrum. The computational complexity of counting. In *Proceedings of the International Congress of Mathematicians, Zr̆ich, 1994*, pages 1407–1416, Basel, 1995. Birkhäuser.

[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.

[JS97] Mark Jerrum and Alistair Sinclair. The Markov chain Monte Carlo method: An approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems*, pages 482–520. PWS Publishing Company, 1997.

[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.

[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.

[KLM89] Richard M. Karp, Michael Luby, and Neal Madras. Monte-Carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10(3):429–448, September 1989.

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

[LV97] Michael Luby and Eric Vigoda. Approximately counting up to four (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 682–687, El Paso, Texas, 4–6 May 1997.

[LV99] Michael Luby and Eric Vigoda. Fast convergence of the Glauber dynamics for sampling independent sets. *Random Structures & Algorithms*, 15(3–4):229–241, 1999.

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

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*, chapter 11: Approximate Counting, pages 306–334. Cambridge University Press, 1995.

[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.

[Orp90] Pekka Orponen. Dempster's rule of combination is #P-complete. *Artificial Intelligence*, 44(1–2):245–253, 1990.

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

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[Pro90] Gregory M. Provan. A logical-based analysis of Dempster-Shafer theory. *International Journal of Approximate Reasoning*, 4:451–498, 1990.

[Pro86] J. Scott Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM Journal on Computing*, 15(3):694–702, August 1986.

[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.

[RT98] Dana Randall and Prasad Tetali. Analyzing Glauber dynamics by comparison of Markov chains. In C.L. Lucchesi and A.V. Moura, editors, *Proceedings of LATIN '98: Third Latin American Symposium on Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 292–304. Springer, April 1998.

[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.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, November 1985.

[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.

[Tuc95] Alan Tucker. *Applied Combinatorics.* John Wiley & Sons, Inc., 3rd edition, 1995.

[Vad95] Salil P. Vadhan. The complexity of counting. Undergraduate thesis, Harvard University, 1995. Available from `http://deas.harvard.edu/~salil`.

[Vad96] Salil P. Vadhan. Rapidly mixing Markov chains and their applications. Essay, Churchill College, Cambridge University, May 1996. Available from `http://deas.harvard.edu/~salil`.

[Vad97] Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. Preliminary version, May 1997. Available from `http://deas.harvard.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.

[Wel93] Dominic J. A. Welsh. *Complexity: Knots, Colourings, and Counting.* Number 188 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1993.

[Zuc96] David Zuckerman. On unapproximable versions of NP-complete problems. *SIAM Journal on Computing*, 25(6):1293–1304, 1996.