

Simple deterministic approximation algorithms for counting matchings ^{*}

Mohsen Bayati [†] David Gamarnik [‡] Dmitriy Katz [§] Chandra Nair [¶] Prasad Tetali ^{||}

Abstract

We construct a deterministic fully polynomial time approximation scheme (FPTAS) for computing the total number of matchings in a bounded degree graph. Additionally, for an arbitrary graph, we construct a deterministic algorithm for computing approximately the number of matchings within running time $\exp(O(\sqrt{n} \log^2 n))$, where n is the number of vertices. Finally, we construct an FPTAS for the problem of computing the number of independent sets in certain subclasses of claw-free graphs.

Our approach is based on the *correlation decay* technique originating in statistical physics. Previously this approach was successfully used for approximately counting the number of independent sets and colorings in some classes of graphs [BG06],[Wei06],[GK07a]. Thus our results extend the small, but growing class of $\#P$ -complete problems for which there is now a deterministic FPTAS.

1 Introduction

The focus of the paper is the problem of computing the total number of (full and partial) matchings in a given graph, and the problem of computing the number of independent sets in certain classes of claw-free graphs (for the definition of a claw-free graph and other related definitions see Section 5). These problems, along with many other combinatorial counting problems falls into the class of $\#P$ -complete problems, and thus, modulo a basic complexity theoretic conjecture, cannot be solved (exactly) in polynomial time. Ever since the introduction of the $\#P$ complexity class by Valiant [Val79], the algorithmic focus on these $\#P$ -complete problems shifted to finding efficient approximating solutions. Here the strongest and the most general method comes from the Markov chain Monte Carlo (MCMC) approach [JVV86],[JS97, J03], [S93, JS97], which is based on the equivalence between the counting problem and a related sampling problem [JVV86]. This approach provided one with *fully polynomial randomized approximation schemes* (FPRAS) to some $\#P$ -complete counting problems.

The first non-trivial instance of a deterministic approximation scheme for $\#P$ -complete counting problem is due to Weitz [Wei06]; where he demonstrated the equivalence between computing a particular marginal probability (corresponding to a Gibbs distribution) on a graph to that on a certain computation tree for any binary spin system. Following this observation, he established a *spatial correlation decay* in the associated tree to approximately compute the marginals from a suitable neighborhood of the vertex. Indeed, our present work follows similar lines, where we use a known recursion[God81] on matchings to

^{*}Preliminary version of this paper will appear in Proceedings of 39th ACM Symposium on Theory of Computing, San Diego, CA

[†]Microsoft Research, One Microsoft Way, Redmond, WA 98052; e-mail: mohsenb@microsoft.com

[‡]Operations Research Center and Sloan School of Management, MIT, Cambridge, MA 02139; e-mail: gamarnik@mit.edu

[§]Operations Research Center, MIT, Cambridge, MA 02139; e-mail: dimdim@mit.edu

[¶]Department of Information Engineering, Chinese University of Hong Kong, ; e-mail: chandra@cuhk.edu.hk

^{||}School of Mathematics and School of Mathematics, Georgia Tech, Atlanta, GA 30332-0160; e-mail: tetali@math.gatech.edu

transform the recursive equations for marginal probability on graphs to one on trees. Then we establish a sufficiently fast spatial correlation decay for this recursion so as to enable us to efficiently approximate the marginal probability at a vertex for bounded degree graphs.

While existence of spatial correlation decay for the Gibbs distributions corresponding to matchings has been known in the statistical physics literature [HL72, vdB98], we establish correlation decay property on a computation tree since the latter leads to polynomial time approximation algorithm, while the former does not (see Section 6). Specifically, we obtain deterministic FPTAS for the problem of counting matchings in bounded degree graphs, which is our main result. One should note that the correlation decay on a computation tree implies spatial correlation decay. Our proof of the correlation decay property is similar to the one found in [KK98]. Thus, while the main conclusion *could have been deduced* by combining several earlier results and techniques, we establish it in a self-contained and a relatively simple way.

Finally the structural similarity of the recursive equations for counting matchings to the recursive equations for counting independent sets, allows one (in a certain class of claw free graphs) to obtain an FPTAS for counting independent sets as well.

1.1 Background

Recently an alternative approach for constructing approximate counting schemes was developed, leading to *deterministic* approximation schemes for counting problems. The approach is based on establishing *correlation decay* property on a certain *computation tree* associated with the underlying counting problem. The concept of *correlation decay* (also sometimes called *spatial* correlation decay) originates in statistical physics [Dob70] and is related to the notion of uniqueness of Gibbs measures in infinite lattices, hence the interest of statistical physicists in this subject. The approach of using correlation decay property for constructing efficient algorithms for counting was introduced in Weitz [Wei06]; where he observed that by constructing a certain computation tree, which has a self-avoiding tree structure and establishing the correlation decay property on this tree (rather than the original graph), one is able sometimes to solve the counting problem approximately. The correlation decay approach was also recently used by Bandyopadhyay and Gamarnik [BG06], for the problems of counting the number of independent sets and colorings of a graph in regular graphs with large girth. The development in [Wei06] was particularly impressive as, unlike in [BG06], no assumptions on the girth of a graph are made and the approach led to the first known *deterministic* fully polynomial time approximation scheme (FPTAS) for a natural problem in the #P-complete class, namely the problem of counting the number of independent sets in graphs with degree ≤ 5 . Recently, the correlation decay on a computation tree approach was also put to use in Gamarnik and Katz [GK07a] for the problem of counting the number of proper colorings of a graph. For completeness, we reproduce here a section from [GK07a] which explains the crucial difference between having the correlation decay property in the underlying graph and on a computation tree. Recent related works in this directions also include Jung and Shah [JS] and Nair and Tetali [NT].

The main goal of this paper is to use the correlation decay approach for constructing a fully polynomial deterministic approximation scheme for counting the total number of (partial and full) matchings of a graph. Thanks to Vadhan's work [V01], the problem of counting matchings in regular graphs of bounded degree (to be precise, degree \geq five) is also known to be #P-complete. In fact we solve a more general problem - the one of computing the *partition function* corresponding to matchings with a given activity level λ . The result is obtained by establishing the correlation decay property on the computation tree *for every* $\lambda > 0$ and *every* degree Δ (contrast with $\Delta \leq 5$ condition for counting independent sets [Wei06]). Interestingly, the analysis becomes far easier, when compared to the counterparts in

independent sets and colorings. The proof of the correlation decay property is done differently than in [Wei06] and uses mean value theorem to establish a required contraction. A similar approach was used in [KK98] and [GK07a], but in the present context of counting matchings it is particularly simple. Using a two-level recursion analysis, we show that the rate of the correlation decay is $\approx 1 - O(\frac{1}{\sqrt{\Delta}})$, where Δ is the maximum degree of the graph. As a corollary we construct a deterministic FPTAS for computing the number of matchings in any graph with $\Delta = O(1)$. For the case of arbitrary graphs (with no restriction on the maximum degree) we construct a deterministic approximation scheme which runs in time $\exp(O(\sqrt{n} \log^2 n))$, where n the number of vertices in the graph.

Our approach turns out to extend fairly directly to a different problem: counting the number of independent sets in constant degree claw-free graphs which are guaranteed to contain at least one simplicial clique (for definition see Section 5). It turns out that many types of claw-free graphs are guaranteed to contain a simplicial clique, line graphs being one of them. Note that independent sets in the *line graph* $\mathcal{L}(\mathbb{G})$, of a graph \mathbb{G} correspond to matchings in the underlying graph \mathbb{G} , thus our matching result applies directly. In fact this observation plus a recent result of Chudnovskaya and Seymour [CS07], which establishes that independence polynomial of a claw free graph has only real roots, served as a motivation for our result. We show that once a simplicial clique is guaranteed to exist in a given graph \mathbb{G} , a correlation decay property on the Gibbs measure corresponding to independent sets holds, and we construct an FPTAS in a way almost identical to its matching counterpart.

The problem of computing the number of matchings of graphs with large girth was addressed recently by Bayati and Nair [BN06] in the context of Belief Propagation algorithms and the validity of the cavity method. The use of tree like recursions, similar to the one in this paper, for computing the matching polynomials can also be found in the work of [God81]. The fact that the Gibbs distribution corresponding to matchings exhibits a spatial correlation decay was already established by Heilmann and Lieb [HL72] by looking at complex roots of the partition function and later by van den Berg [vdB98] using more probabilistic/combinatorial arguments. We also note that the work of Kahn and Kim [KK98] is pertinent here. Using Godsil's (self-avoiding) tree construction, they show that in any Δ -regular graph, the probability that any fixed vertex is *not* in a (uniformly chosen) random matching is asymptotically $1/\sqrt{\Delta}$; moreover, as we realized since doing the present work, their proof establishing this fact (see Claim (2.6) in their paper) also uses (as we do below in Theorem 3.2) a two level recursion and partial derivatives to show convergence to the above probability estimate.

It should be noted that, while an FPRAS for counting matchings is known to exist thanks to the MCMC method, constructing a deterministic counterpart was an open problem, prior to this work, and in general constructing non-randomized approximate counting algorithms is a very challenging task. For example, while an FPRAS for computing permanent of a matrix is known [JSV04], the best known deterministic approximation scheme only gives factor e^n approximation [LSW00]. In fact, the present paper led to a recent improvement of this result [GK07b].

The connection between the correlation decay property and the mixing rates of the Markov chain corresponding to the counting problems has been investigated recently [DSVW04],[GMP05],[BKMP01],[MS06]. It is known that if the underlying graph satisfies a certain sub-exponential growth condition, the spatial correlation decay implies rapid mixing (see e.g., [DSVW04], [GMP05]). The converse, however, does not hold in general, as shown by Berger et al. [BKMP01].

The present work, along with [BG06],[BN06],[Wei06],[GK07a],[GK07b],[NT] reinforces this connection, as well as, broadly speaking, contributes to the exciting and emerging connection between theoretical computer science, probability theory and statistical physics.

The rest of the paper has the following structure. The definitions and the main result Theorem 2.1 are presented in Section 2. The correlation decay analysis is the subject of Section 3. The approximate counting algorithm and the complexity analysis is presented in Section 4. The extensions to the problem

of counting independent sets in claw-free graph is done in Section 5. The discussion on the difference between spatial correlation decay and correlation decay on a computation tree, borrowed from [GK07a] is given in Section 6. Some concluding remarks and further open questions are presented in Section 7.

2 Definitions, preliminaries and the main result

We consider a simple labeled undirected graph \mathbb{G} with n vertices. The vertex set is denoted by $V = \{v_1, \dots, v_n\}$. E denotes the edge set. $N(v, \mathbb{G}) \subset V$ denotes the set of neighbors of v . We will also use $N(v)$, when the underlying graph \mathbb{G} is unambiguous. The degree of the graph is $\Delta \triangleq \max_v |N(v, \mathbb{G})|$. We abuse notations by letting $\mathbb{G} \setminus \{v\}$ denote a subgraph of \mathbb{G} induced by nodes $V \setminus \{v\}$. A matching is a subset $M \subseteq E$ such that no two edges in M share a vertex. We denote by $\mathcal{M} = \mathcal{M}(\mathbb{G})$ the set of all matchings of \mathbb{G} .

Given a fixed parameter $\lambda > 0$, called the *activity*, a natural (Gibbs) probability distribution on the set \mathcal{M} of matchings is defined as:

$$\mathbb{P}_{\mathbb{G}}(M) = \frac{\lambda^{|M|}}{Z(\mathbb{G})},$$

where the normalizing constant $Z(\mathbb{G})^*$ is called the *partition function* corresponding to λ , and is expressed as

$$Z(\mathbb{G}) = \sum_{M \in \mathcal{M}} \lambda^{|M|}.$$

When $\lambda = 1$, $Z(\mathbb{G})$ counts the number of matchings $|\mathcal{M}|$, both complete and partial. Our goal is constructing an algorithm which computes $Z(\mathbb{G})$ approximately. The instance size of the problem is $O(\max(|V|, |E|, |\log \lambda|))$.

Definition 2.1. *An approximation algorithm \mathcal{A} is a Fully Polynomial Time Approximation Scheme (FPTAS) for computing $Z(\mathbb{G})$ if, given an arbitrary $\epsilon > 0$, it produces a value \hat{Z} satisfying*

$$\exp(-\epsilon) \leq \frac{\hat{Z}}{Z(\mathbb{G})} \leq \exp(\epsilon),$$

in time which is polynomial in $1/\epsilon$ and size instance.

We now state our main result. Here and throughout \log denotes natural logarithm.

Theorem 2.1. *There exists a deterministic algorithm which provides FPTAS for computing $Z(\mathbb{G})$ corresponding to a constant activity $\lambda > 0$ for any graph, whose degree is bounded by a constant Δ . The above algorithm has complexity $O\left(\left(\frac{n}{\epsilon}\right)^{\kappa \log \Delta + 1}\right)$, where $\kappa = -2 / \log\left(1 - \frac{2}{\sqrt{1+\lambda\Delta}+1}\right)$.*

Thus, while the running time of the algorithm depends polynomially on $1/\epsilon$ (hence *Fully Polynomial approximation scheme*) the degree of the polynomial depends on both Δ and λ .

Additionally, we will show that the algorithm of Theorem 2.1 has subexponential complexity when the graph is general (non-constant degree).

Corollary 2.2. *The complexity of the algorithm of Theorem 2.1 is $\exp(O(\sqrt{n} \log^2 n))$ for general graphs and constant $\lambda > 0$.*

*For ease of notation we suppress the dependence of $Z(\mathbb{G})$ on λ .

We now introduce an identity which appears in various forms in the context of Markov chain sampling method as well. The identity is also the essence of the cavity method in statistical physics. It shows that the problem of computing the partition function can be reduced to the problem of computing certain marginal probabilities. Problems admitting such reduction are called self reducible.

Denote by \mathbf{M} a random matching selected according to the Gibbs measure $\mathbb{P}_{\mathbb{G}}$. Abusing notation slightly, we write $v \in \mathbf{M}$ to say that matching \mathbf{M} contains an edge incident to v .

Proposition 2.3. *Let $\mathbb{G}_k = \mathbb{G} \setminus \{v_1, \dots, v_{k-1}\}$ with $\mathbb{G}_1 = \mathbb{G}$. Then*

$$Z(\mathbb{G}) = 1 / \prod_{1 \leq k < n} \mathbb{P}_{\mathbb{G}_k}(v_k \notin \mathbf{M}).$$

Proof. Observe that for every graph \mathbb{G} and vertex $v \in \mathbb{G}$

$$\mathbb{P}_{\mathbb{G}_k}(v_k \notin \mathbf{M}) = \frac{Z(\mathbb{G}_k \setminus \{v_k\})}{Z(\mathbb{G}_k)} = \frac{Z(\mathbb{G}_{k+1})}{Z(\mathbb{G}_k)}. \quad (1)$$

Applying this identity recursively to v_k, \mathbb{G}_k , and using the convention that the partition function for the graph with no edges, $Z(\mathbb{G}_n)$, equals 1, we obtain the result. \square

The following corollary is a straightforward application of Proposition 2.3 and therefore we shall focus our attention on constructing an algorithm for computing an approximation of $\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M})$.

Corollary 2.4. *Given any $\epsilon > 0$, if there exists a fully polynomial time algorithm \mathcal{A} , which on input (\mathbb{G}_k, v_k) , $1 \leq k \leq n - 1$, computes a value $\hat{p}(v_k)$ satisfying*

$$\exp(-\frac{\epsilon}{n}) \leq \frac{\hat{p}(v_k)}{\mathbb{P}_{\mathbb{G}_k}(v_k \notin \mathbf{M})} \leq \exp(\frac{\epsilon}{n}), \quad (2)$$

for any $\mathbb{G}_k \subseteq \mathbb{G}$ having $n - k + 1$ vertices, then one immediately obtains a fully polynomial time approximation algorithm for $Z(\mathbb{G})$.

Thus in the next section we will interest ourselves in the construction of an FPTAS for computing $\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M})$ for a graph \mathbb{G} with m vertices, within an approximation factor $\exp(\frac{\epsilon}{n})$, for some fixed $n \geq m$.

3 Basic recursion and correlation decay analysis

The following recursion is similar to the one obtained by Godsil in [God81] and Heilmann and Lieb [HL72]. It was also used by Kahn and Kim [KK98] for the analysis of random matchings on regular large degree graphs.

Proposition 3.1. *The following holds for every vertex v :*

$$\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M}) = \frac{1}{1 + \lambda \sum_{u \in N(v, \mathbb{G})} \mathbb{P}_{\mathbb{G} \setminus \{v\}}(u \notin \mathbf{M})}. \quad (3)$$

Proof. We have

$$Z(\mathbb{G}) = \sum_{M \in \mathcal{M}(\mathbb{G}): v \notin M} \lambda^{|M|} + \sum_{u \in N(v, \mathbb{G})} \sum_{M \in \mathcal{M}(\mathbb{G}): (v, u) \in M} \lambda^{|M|}. \quad (4)$$

Observe that the set of all matchings M such that $v \notin M$ is the set of all matchings in $\mathbb{G} \setminus \{v\}$. Also for every matching M containing (v, u) , $M \setminus \{(v, u)\}$ induces a matching in the graph $\mathbb{G} \setminus \{v, u\}$. Conversely, for every matching M in $\mathbb{G} \setminus \{v, u\}$, $M \cup \{(v, u)\}$ creates a matching in \mathbb{G} containing the edge (v, u) . Thus we can rewrite (4) as

$$Z(\mathbb{G}) = Z(\mathbb{G} \setminus \{v\}) + \sum_{u \in N(v, \mathbb{G})} \lambda Z(\mathbb{G} \setminus \{v, u\}).$$

Dividing both parts by $Z(\mathbb{G} \setminus \{v\})$ and using the identity (1) we obtain the result. \square

Definition 3.1. For every induced subgraph \mathbb{H} of the graph \mathbb{G} every vertex $v \in \mathbb{H}$ and every $t \in \mathbb{Z}_+$ let $\Phi_{\mathbb{H}}(v, t)$ be defined inductively as follows:

$$\Phi_{\mathbb{H}}(v, t) = \begin{cases} \Phi_{\mathbb{H}}(v, 0) = 1, & \text{for all } \mathbb{H}, v; \\ \left(1 + \lambda \sum_{u \in N(v, \mathbb{H})} \Phi_{\mathbb{H} \setminus \{v\}}(u, t-1)\right)^{-1}, & t \geq 1. \end{cases} \quad (5)$$

While we have introduced the values $\Phi_{\mathbb{H}}(v, t)$ for potentially exponentially many subgraphs of \mathbb{G} , it is only a small family of subgraphs of \mathbb{G} for which the value of Φ will be relevant to us. The quantity $\Phi_{\mathbb{H}}(v, t)$ will serve as an approximation of $\mathbb{P}_{\mathbb{H}}(v \notin \mathbf{M})$. The essence of this approximation is described in the following result which is the basis of our algorithm.

Theorem 3.2 (Correlation Decay). *The following holds for every vertex v and every positive even value t :*

$$\left| \log \mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M}) - \log \Phi_{\mathbb{G}}(v, t) \right| \leq \left(1 - \frac{2}{(\sqrt{1 + \lambda \Delta} + 1)}\right)^{t/2} \log(1 + \lambda \Delta).$$

Proof. Fix a vertex $v \in \mathbb{G}$, and let $N(v, \mathbb{G}) = \{u_1, \dots, u_m\}$, $N(u_i, \mathbb{G} \setminus \{v\}) = \{w_1^{(i)}, \dots, w_{m_i}^{(i)}\}$. We introduce the following shorthand notations, with x 's representing the true probabilities (of certain vertices not being in random matchings) and y 's representing the corresponding approximations:

$$x = \log \mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M}), \quad x_i = \log \mathbb{P}_{\mathbb{G} \setminus \{v\}}(u_i \notin \mathbf{M}), \quad x_j^{(i)} = \log \mathbb{P}_{\mathbb{G} \setminus \{v, u_i\}}(w_j^{(i)} \notin \mathbf{M}), \quad (6)$$

$$y = \log \Phi_{\mathbb{G}}(v, t), \quad y_i = \log \Phi_{\mathbb{G} \setminus \{v\}}(u_i, t-1), \quad y_j^{(i)} = \log \Phi_{\mathbb{G} \setminus \{v, u_i\}}(w_j^{(i)}, t-2), \quad (7)$$

for $i = 1, \dots, m$, $j = 1, \dots, m_i$.

Let $M = \sum_{i=1}^m m_i$, $\vec{z} = (z_1^{(1)}, \dots, z_{m_1}^{(1)}, \dots, z_1^{(m)}, \dots, z_{m_m}^{(m)})$. Let $f : [0, 1]^M \rightarrow [0, 1]$ be given as

$$f(\vec{z}) = \log \left(1 + \lambda \sum_{i=1}^m \frac{1}{1 + \lambda \sum_{j=1}^{m_i} e^{z_j^{(i)}}}\right).$$

Then we can rewrite (3) and (5) as $x = -f(\vec{x})$, $y = -f(\vec{y})$. We consider $g(\alpha) = f(\alpha \vec{x} + (1 - \alpha) \vec{y})$ as a function of one-dimensional parameter $\alpha \in [0, 1]$ and fixed vectors \vec{x}, \vec{y} . Applying the mean value theorem, there exists $\alpha \in [0, 1]$ such that for $\vec{z}_\alpha = \alpha \vec{x} + (1 - \alpha) \vec{y}$,

$$|x - y| = |\nabla f(\vec{z}_\alpha) \cdot (\vec{x} - \vec{y})| \stackrel{(a)}{\leq} \|\nabla f(\vec{z}_\alpha)\|_{L_1} \|\vec{x} - \vec{y}\|_{L_\infty}, \quad (8)$$

where (a) follows from $|\sum a_i b_i| \leq (\sum |a_i|) \max |b_i| = \|a\|_{L_1} \|b\|_{L_\infty}$. It is easy to see that

$$\|\nabla f(\vec{z})\|_{L_1} = \frac{1}{1 + \lambda \sum_{i=1}^m \frac{1}{1 + \lambda \sum_{j=1}^{m_i} e^{z_j^{(i)}}}} \sum_{i=1}^m \lambda \left(\frac{1}{1 + \lambda \sum_{j=1}^{m_i} e^{z_j^{(i)}}} \right)^2 \lambda \sum_{j=1}^{m_i} e^{z_j^{(i)}}.$$

Lemma 3.3. For every \vec{z} ,

$$\|\nabla f(\vec{z})\|_{L_1} \leq 1 - \frac{2}{(\sqrt{1 + \lambda\Delta} + 1)}.$$

Proof. Define $A_i = 1 + \lambda \sum_{j=1}^{m_i} e^{z_j^{(i)}}$. The L_1 -norm can be re-written, in terms of A_i as

$$\|\nabla f(\vec{z})\|_{L_1} = \frac{1}{1 + \lambda \sum_{i=1}^m \frac{1}{A_i}} \sum_{i=1}^m \frac{\lambda(A_i - 1)}{A_i^2} = 1 - \frac{1 + \lambda \sum_{i=1}^m \frac{1}{A_i^2}}{1 + \lambda \sum_{i=1}^m \frac{1}{A_i}}.$$

It is not difficult to see that the expression $\frac{1 + \lambda \sum_{i=1}^m \frac{1}{A_i^2}}{1 + \lambda \sum_{i=1}^m \frac{1}{A_i}}$ is minimized, for $0 \leq 1/A_i \leq \infty$, when $1/A_i = \frac{\sqrt{1 + \lambda m} - 1}{\lambda m}$. To show this, first observe by taking partial derivatives that the minimum occurs when all of A_i are equal. Then the solution for optimal A_i reduces to a quadratic equation. Substituting for the minimum value, one obtains

$$\|\nabla f(\vec{z})\|_{L_1} = 1 - \frac{1 + \lambda \sum_{i=1}^m \frac{1}{A_i^2}}{1 + \lambda \sum_{i=1}^m \frac{1}{A_i}} \leq 1 - \frac{2}{(\sqrt{1 + \lambda m} + 1)} \leq 1 - \frac{2}{(\sqrt{1 + \lambda\Delta} + 1)}.$$

□

Applying Lemma 3.3 to (8) we obtain

$$\begin{aligned} & \left| \log \mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M}) - \log \Phi_{\mathbb{G}}(v, t) \right| \\ & \leq \left(1 - \frac{2}{(\sqrt{1 + \lambda\Delta} + 1)} \right) \max_{i,j} \left| \log \mathbb{P}_{\mathbb{G} \setminus \{v, u_i\}}(w_j^{(i)} \notin \mathbf{M}) - \log \Phi_{\mathbb{G} \setminus \{v, u_i\}}(w_j^{(i)}, t - 2) \right|. \end{aligned}$$

Iterating this bound $t/2$ times we obtain that $\left| \log \mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M}) - \log \Phi_{\mathbb{G}}(v, t) \right|$ is at most $(1 - \frac{2}{(\sqrt{1 + \lambda\Delta} + 1)})^{t/2}$ times $\max_{\mathbb{H}, u} \left| \log \mathbb{P}_{\mathbb{H}}(u \notin \mathbf{M}) - \log \Phi_{\mathbb{H}}(u, 0) \right|$, where the maximum is over all subgraph/vertex pairs ($\mathbb{H} \subset \mathbb{G}, u \in \mathbb{H}$). Observe from (3) that

$$\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M}) \geq \frac{1}{1 + \lambda\Delta}.$$

Applying this bound and the fact $\Phi_{\mathbb{H}}(u, 0) = 1$, $\max_{\mathbb{H}, u} \left| \log \mathbb{P}_{\mathbb{H}}(u \notin \mathbf{M}) - \log \Phi_{\mathbb{H}}(u, 0) \right|$ is at most $\log(1 + \lambda\Delta)$. □

Remark : In the proof below we analyzed two steps of the recursions (3) and (5). This leads to a correlation decay rate $\approx (1 - \frac{1}{\sqrt{\lambda\Delta}})$. We could instead use a one-step analysis, but this would give us a decay rate only $\approx (1 - \frac{1}{\lambda\Delta})$. While this would not make a big difference in the case $\lambda, \Delta = O(1)$, it does make a difference in the general case, since Δ could be as large as $n - 1$. The case $\lambda = O(n^{\frac{1}{3}})$ was used in [GK07b] for constructing an approximation algorithm for computing a permanent. It also seems that 3-step analysis would not buy us better correlation decay as the rate $\approx (1 - \frac{1}{\sqrt{\lambda\Delta}})$ is tight, as can be checked on a regular tree. This observation is also implicit in [KK98].

4 Algorithm

Our algorithm is based on computing the values $\Phi_{\mathbb{G}}(v, t)$. In our analysis of algorithm complexity, we assume that each arithmetic operation takes one unit of time. This can be done since arithmetic operations introduce at most a polynomial time overhead in the computation.

Lemma 4.1. *The values $\Phi_{\mathbb{G}}(v, t)$ can be computed in time $O(\Delta^t)$. In particular when $t = O(\log n)$ and $\Delta = O(1)$, $\Phi_{\mathbb{G}}(v, t)$ can be computed in polynomial time.*

Proof. The proof follows immediately from recursion (5). □

Based on this lemma, we propose the following algorithm for estimating the partition function $Z(\mathbb{G})$.

Algorithm CountMATCHINGS

```

INPUT: A graph/activity pair  $(\mathbb{G}, \lambda)$  and a positive integer  $t$ .
BEGIN
Set  $Z = 1, \mathbb{H} = \mathbb{G}$ .
While  $\mathbb{H} \neq \emptyset$ , find an arbitrary node  $v \in \mathbb{H}$ . Compute  $\Phi_{\mathbb{H}}(v, t)$ .
Set  $Z = \frac{Z}{\Phi_{\mathbb{H}}(v, t)}$ ,  $\mathbb{H} = \mathbb{H} \setminus \{v\}$ .
END
OUTPUT:  $Z$ .

```

As a final step we show that Φ can be used to approximate the marginal probabilities $\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M})$ with polynomial accuracy.

Lemma 4.2. *Let $\delta = -\log\left(1 - \frac{2}{\sqrt{1+\lambda\Delta}+1}\right)$. If $t = 2\lceil(\log n + \log \log(1 + \lambda\Delta) - \log \epsilon)/\delta\rceil$, then*

$$e^{-\frac{\epsilon}{n}} \leq \frac{\Phi_{\mathbb{G}}(v, t)}{\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M})} \leq e^{\frac{\epsilon}{n}}.$$

Proof. Theorem 3.2 implies

$$\left| \log \frac{\Phi_{\mathbb{G}}(v, t)}{\mathbb{P}_{\mathbb{G}}(v \notin \mathbf{M})} \right| \leq \left(1 - \frac{2}{\sqrt{1+\lambda\Delta}+1}\right)^{(\log n + \log \log(1+\lambda\Delta) - \log \epsilon)/\delta} \log(1 + \lambda\Delta) = \frac{\epsilon}{n}.$$

The result then follows. □

Proof of Theorem 2.1. First we assume the case of bounded degree graph and constant activity: $\lambda, \Delta = O(1)$. The bound on t given by Lemma 4.2 becomes in this case $t = O(\log n)$. The algorithm providing FPTAS is CountMATCHINGS, with input \mathbb{G}, λ , and $t = O(\log n)$ as in Lemma 4.2. We can combine Lemma 4.1, Lemma 4.2 and Corollary 2.4 and observe that the complexity of the algorithm is bounded by $O(n\Delta^t)$ where $t = 2\lceil(\log n + \log \log(1 + \lambda\Delta) - \log \epsilon)/\delta\rceil$. This gives the desired FPTAS with the complexity bounded as stated in the theorem.

In the general case we have for δ defined in Lemma 4.2 that $\delta^{-1} = O(\sqrt{\lambda\Delta})$. Due to our assumption $\lambda = O(n)$, this gives $t = O(\frac{1}{\delta} \log \frac{n}{\epsilon}) = O(\sqrt{\lambda\Delta} \log \frac{n}{\epsilon})$. Thus again applying Lemma 4.1, the complexity of the algorithm CountMATCHINGS is

$$O(n\Delta^t) = O(n^t) = \exp(O(\sqrt{\lambda\Delta} \log^2 n)),$$

where $\Delta = O(n)$ is used. (Here we ignore the explicit dependence on ϵ but it is easy to see that the complexity depends polynomially on $\frac{1}{\epsilon}$.) The special case corresponding to counting matchings $\lambda = 1$ leads to an upper bound $\exp(O(\sqrt{n} \log^2 n))$. \square

5 Claw-free graphs and counting independent sets

In this section we show how the results of the previous sections extend directly to the problem of computing the number of independent sets in certain subclasses of claw-free graphs. A set $I \subset V$ is an independent set (also known as a stable set) if for every $u, v \in I$, we have $(u, v) \notin E$. A set $K \subset V$ is a clique if $(u, v) \in E$ for every $u, v \in K$. Let $cl(\mathbb{G})$ denote the size of the largest clique in \mathbb{G} . Given $\lambda > 0$, define the partition function (also known as the independent set polynomial) to be $Z_I(\mathbb{G}) = \sum_I \lambda^{|I|}$, where the summation is over all independent sets of \mathbb{G} . A Gibbs measure on the set of independent sets is then introduced as

$$\mathbb{P}_{\mathbb{G}}(I) = \frac{\lambda^{|I|}}{Z_I(\mathbb{G})}.$$

We denote by I a random independent set selected according to this measure.

A claw is a graph on four nodes v, v_1, v_2, v_3 with precisely the edges (v, v_i) , $i = 1, 2, 3$. A graph is claw-free if it does not contain an induced claw. The line graph $\mathcal{L}(\mathbb{G})$ of a graph \mathbb{G} is a graph whose vertices are the edges of \mathbb{G} and for $e, e' \in E(\mathbb{G})$, there is an edge between e and e' in $\mathcal{L}(\mathbb{G})$ if and only if e and e' share a vertex in \mathbb{G} . Thus every vertex in $\mathcal{L}(\mathbb{G})$, say corresponding to edge (i, j) in \mathbb{G} , is part of exactly two cliques, one composed of vertices in $\mathcal{L}(\mathbb{G})$ corresponding to the edges in \mathbb{G} that share vertex i , and the other composed of vertices in $\mathcal{L}(\mathbb{G})$ corresponding to the edges in \mathbb{G} that share vertex j . This immediately implies that $\mathcal{L}(\mathbb{G})$ is claw-free. Recently it was shown that the partition function of every claw-free graph has only real roots [CS07], answering an earlier question by Hamidoune [Ham90] and Stanley [Stan98]. A similar result for matchings is known to be true and is a subject of a classical paper by Heilmann and Lieb [HL72], which has been used to show the absence of phase transition for the matching model for every $\lambda > 0$. It stands to reason that perhaps a similar correlation decay (in an appropriate sense) holds for the Gibbs distribution of independent sets in claw-free graphs; and consequently, perhaps a fast algorithm for counting independent sets in claw-free graphs is possible. The goal of this section is a step in this direction – while we are not yet able to prove such a result for every claw-free graph, we do obtain the result for claw-free graphs which contain a so-called *simplicial* clique (see definition below).

Following the work of Chudnovsky and Seymour [CS07], let a clique K in a graph \mathbb{G} be called *simplicial* if for every $v \in K$, $N(v) \setminus K$ is a clique. Then the main result of this section is the following.

Theorem 5.1. *Let \mathbb{G} be a connected, claw-free graph containing a simplicial clique. Then there exists a deterministic FPTAS for computing $Z_I(\mathbb{G})$ for \mathbb{G} with constant $cl(\mathbb{G})$ and for every constant λ . The complexity of this algorithm is $O(\left(\frac{n}{\epsilon}\right)^{\kappa \log cl(\mathbb{G}) + 1})$, where κ is the same as in Theorem 2.1 with $cl(\mathbb{G})$ replacing Δ .*

Several observations are in order. If Δ is a constant then $cl(\mathbb{G})$ is a constant as well, and therefore the result applies to the case of bounded degree graphs as well.

Note that a clawfree graph need not have a simplicial clique at all! The graph of the icosahedron is an example, and larger families of graphs can be constructed using small examples such as the icosahedron. On the other hand, there are certainly interesting classes of graphs which do contain a simplicial clique – circular interval graphs is one such example family. More importantly, it can be

shown (see Lemma 5.2 below) that once a claw-free graph has a simplicial clique, then any induced subgraph of it will also contain a simplicial clique, thus making a recursive approach possible. For additional (wealth of) information on clawfree graphs and related aspects, we recommend the reader to consult [CS08].

The rest of the section is devoted to proving Theorem 5.1. We begin with the following useful lemma provided to us by Maria Chudnovsky.

Lemma 5.2. *Let \mathbb{G} be a connected, clawfree graph with a simplicial clique, and let \mathbb{H} be an induced subgraph of \mathbb{G} . Then \mathbb{H} has a simplicial clique.*

Proof. Let \mathbb{G} and \mathbb{H} be as in the statement of the lemma. We may assume that \mathbb{H} is connected, since K is a simplicial clique of \mathbb{H} if and only if K is a simplicial clique of some *component* of \mathbb{H} . For $i = 1, 2, \dots$, let V_i be the set of vertices of \mathbb{G} at distance i from \mathbb{H} . Since \mathbb{G} is connected, there exists an integer k so that, $V(\mathbb{G}) = V(\mathbb{H}) \cup \bigcup_{i=1}^k V_i$. By deleting vertices from non-empty V_i with maximal i , we can get from \mathbb{G} to \mathbb{H} by a series of connected, induced subgraphs of \mathbb{G} , deleting one vertex at a time.

Thus, to prove the lemma, it suffices to prove that if \mathbb{G} is a connected, clawfree graph with a simplicial clique, and $v \in V(\mathbb{G})$ is such that $\mathbb{G} \setminus v$ is connected, then $\mathbb{G} \setminus v$ has a simplicial clique. Towards this, let K be a simplicial clique in \mathbb{G} , and let v be an arbitrary vertex of \mathbb{G} . If $K \neq \{v\}$, then $K \setminus v$ is a simplicial clique in $\mathbb{G} \setminus v$. So suppose that $K = \{v\}$. Let $N(v)$ be the set of neighbors v . Then $N(v)$ is a clique. In fact, we claim that $N(v)$ is a simplicial clique in $\mathbb{G} \setminus v$. To prove by contradiction, suppose some vertex w in $N(v)$ has two non-adjacent neighbors x, y in $V(\mathbb{G}) \setminus (N(v) \cup \{v\})$. Then the induced graph on $\{w, x, y, v\}$ is a claw in \mathbb{G} , completing the proof. \square

Observe that the proof of the above lemma suggests an efficient algorithm to find a simplicial clique, given that the graph contains one. However, such a clique is only a simplicial clique in an appropriate subgraph, and not necessarily in the original graph. Thus it is not clear how to efficiently find a simplicial clique in an arbitrary claw-free graph which contains such a clique. Thus we need to make the assumption that $cl(\mathbb{G})$ is bounded, for our approximation algorithm below to be an FPTAS.

In the following, we make additional use of a few simple identities and observations established in [CS07]. The proof of the following lemma is immediate and is depicted in Figure 1 through an example.

Lemma 5.3. *Let $K \subset V(\mathbb{G})$ be a clique. Then*

$$Z_I(\mathbb{G}) = Z_I(\mathbb{G} \setminus K) + \lambda \sum_{v \in K} Z_I(\mathbb{G} \setminus (\{v\} \cup N(v))) \quad (9)$$

The following lemma is proved as Statement 2.4 in [CS07].

Lemma 5.4. *If K is a simplicial clique, then for every $v \in K$, $N(v) \setminus K$ is a simplicial clique in $\mathbb{G}(V \setminus K)$.*

Suppose we have an algorithm for identifying a simplicial clique K and computing approximately $\mathbb{P}_{\mathbb{G}}(K \cap \mathbf{I} = \emptyset)$. This algorithm can be translated into an approximation algorithm for computing $Z_I(\mathbb{G})$ via an observation

$$\mathbb{P}_{\mathbb{G}}(K \cap \mathbf{I} = \emptyset) = \frac{Z_I(\mathbb{G} \setminus K)}{Z_I(\mathbb{G})},$$

and establishing an analogue of Proposition 2.3 (obvious details are omitted). Thus we focus on constructing an algorithm for computing $\mathbb{P}_{\mathbb{G}}(K \cap \mathbf{I} = \emptyset)$. The following identity is the direct analogue of (3).

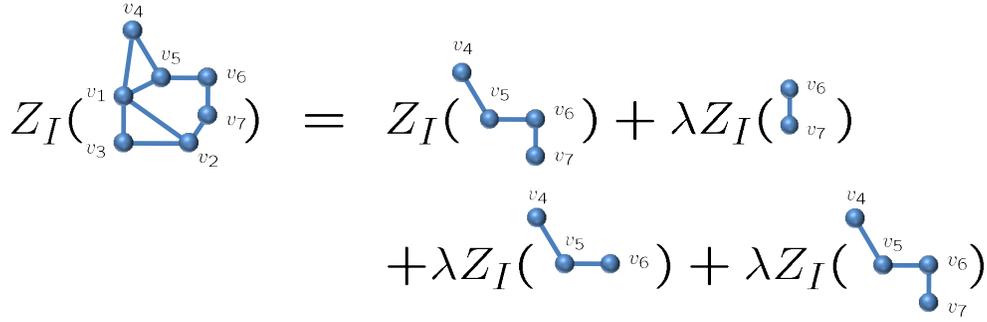


Figure 1: Here is an example of equation (9) for a graph with 7 vertices where the clique K consists of vertices $\{v_1, v_2, v_3\}$.

Proposition 5.5. *Suppose K is a simplicial clique. Then*

$$\mathbb{P}_{\mathbb{G}}(K \cap \mathbf{I} = \emptyset) = \frac{1}{1 + \lambda \sum_{v \in K} \mathbb{P}_{\mathbb{G} \setminus K}((N(v) \setminus K) \cap \mathbf{I} = \emptyset)}. \quad (10)$$

Proof. Divide both parts of (9) by $Z_I(\mathbb{G} \setminus K)$ and observe that $Z_I(\mathbb{G})/Z_I(\mathbb{G} \setminus K) = (\mathbb{P}_{\mathbb{G}}(K \cap \mathbf{I} = \emptyset))^{-1}$ and $Z_I(\mathbb{G} \setminus (\{v\} \cup N(v)))/Z_I(\mathbb{G} \setminus K) = \mathbb{P}_{\mathbb{G} \setminus K}((N(v) \setminus K) \cap \mathbf{I} = \emptyset)$. \square

We observe that the recursion in Proposition 5.5 has again the form $x = 1/(1 + \lambda \sum_k x_k)$, namely precisely the same form as the corresponding recursion for matchings. The analysis of correlation decay follows verbatim, and we conclude that we can compute approximately the values $\mathbb{P}_{\mathbb{G}}(K \cap \mathbf{I} = \emptyset)$ with inverse polynomial accuracy, *provided that we have identified at least one simplicial clique K* . Indeed, once we have a simplicial clique K , the resulting subproblems for computing $\mathbb{P}_{\mathbb{G} \setminus K}((N(v) \setminus K) \cap \mathbf{I} = \emptyset)$ are formulated for cliques $N(v) \setminus K$ in $\mathbb{G} \setminus K$ which are simplicial cliques as well, according to Lemma 5.4. Observe also that at least one of the simplicial cliques $N(v) \setminus K$ is non-empty, otherwise K would be isolated from the rest of \mathbb{G} , contradicting the fact that \mathbb{G} is connected.

We conclude that we have an algorithm for computing $Z_I(\mathbb{G})$, provided that we can identify at least one simplicial clique K to get the algorithm rolling, and the latter is indeed guaranteed by (the proof of) Lemma 5.2. The number of terms in the sum in the recursion (10) is upper bounded $cl(\mathbb{G})$. The analysis of the complexity of our algorithm is from this point identical to the one done for counting matchings, with $cl(\mathbb{G})$ replacing Δ . Observe also, that once the existence of a simplicial clique in a graph \mathbb{G} is guaranteed, it can be found in polynomial time, provided again that $cl(\mathbb{G})$ is constant. Thus, if $cl(\mathbb{G})$ is constant, our algorithm is FPTAS. Unfortunately, we cannot establish an analogue Corollary 2.2, since in this case the search of a simplicial clique can take an exponential amount of time in the worst case.

This completes the proof of Theorem 5.1.

6 Comparison of the correlation decay on a computation tree and the spatial correlation decay property

As we have mentioned above, the (spatial) correlation decay is known to hold for the Gibbs distribution associated with matchings [HL72],[vdB98] in the following sense: for every node v the marginal probability $\mathbb{P}(v \in \mathbf{M})$ is asymptotically independent from similar probabilities associated with nodes on a boundary of the depth- d neighborhood $B(v, d)$ of v in the underlying graph \mathbb{G} . It is natural to try to use this result directly as a method for computing approximately the marginal probabilities $\mathbb{P}(v \in \mathbf{M})$,

for example by computing $\mathbb{P}_{B(v,d)}(v \in \mathbf{M})$ instead, say using brute force computation. Unfortunately, this does not lead to a polynomial time algorithm. In order to obtain ϵ -approximation of the partition function, we need order $O(\epsilon/n)$ approximation of the marginal probabilities, which means the depth d of the neighborhood $B(v,d)$ needs to be at least $O(\log n)$. Here n is the number of nodes. But the resulting cardinality of $B(v,d)$, even for the case of constant degree graphs is $O(\Delta^{\log n}) = n^{O(1)}$ - polynomial in n and the brute-force computation effort would be exponential in n . Notice that even if the underlying graph has a polynomial expansion $|B(v,d)| \leq d^r$, for some power $r \geq 1$, the brute-force computation would still be $O(\exp(\log^r n))$ which is super-polynomial. This is where having correlation decay on computation tree as opposed to spatial decay of correlation is useful.

7 Conclusions

We have constructed a deterministic algorithm for counting approximately the number of matchings of a given graph. The algorithm runs in polynomial time for the class of bounded degree graphs, and in subexponential time $\exp(O(\sqrt{n} \log^2 n))$ for the class of all graphs, where n is the number of nodes. We have extended our result to the problem of counting independent sets in certain classes of claw-free graphs.

A natural open question is whether there is an FPTAS for counting matchings (or other counting problems) in graphs of unbounded degrees. There seem to be some fundamental limitations of the approach proposed in this paper – the correlation decay rate corresponding to the case of matchings seems to be of order $1 - O(\frac{1}{\sqrt{\Delta}})$, and thus we speculate that the improvement should come along some combinatorial, (rather than statistical physics), arguments. In general, it is of interest to see to what extent the correlation decay approach can be used for solving approximately other counting problems for which the MCMC method has been successful. This line of investigation might also bring us a step closer to understanding the extent to which randomized algorithms are more powerful than deterministic algorithms.

Acknowledgements

Bayati, Nair and Tetali acknowledge the support of the Theory Group at Microsoft Research, where part of this work was carried out.

References

- [BG06] A. Bandyopadhyay and D. Gamarnik, *Counting without sampling. New algorithms for enumeration problems using statistical physics.*, Proceedings of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2006.
- [BKMP01] N. Berger, C. Kenyon, E. Mossel, and Y. Peres, *Glauber dynamics on trees and hyperbolic graphs*, Proc. 42nd IEEE Symposium on Foundations of Computer Science (2001).
- [BN06] M. Bayati and C. Nair, *A rigorous proof of cavity method for counting matchings*, Annual Allerton Conference on Communication, Control and Computing, 2006.
- [CS07] M. Chudnovsky and P. Seymour, *The roots of the independence polynomial of a clawfree graph*, J. Combin. Theory Ser. B **97** (2007), no. 3, 350–357.
- [CS08] M. Chudnovsky and P. Seymour, *Claw-free Graphs V. Global Structure*, Preprint.
- [Dob70] R. L. Dobrushin, *Prescribing a system of random variables by the help of conditional distributions*, Theory of Probability and its Applications **15** (1970), 469–497.

- [DSVW04] M. Dyer, A. Sinclair, E. Vigoda, and D. Weitz, *Mixing in time and space for lattice spin systems: a combinatorial view*, Random Struct. & Alg. **24** (2004), 461–479.
- [GK07a] D. Gamarnik and D. Katz, *Correlation decay and deterministic FPTAS for counting list-colorings of a graph*, Proceedings of 18th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007.
- [GK07b] ———, *A deterministic approximation algorithm for computing a permanent of a 0,1 matrix*, Preprint on <http://arxiv.org/abs/math.CO/0702039> (2007).
- [GMP05] L. A. Goldberg, R. Martin, and M. Paterson, *Strong spatial mixing with fewer colors for lattice graphs*, SIAM J. Comput. **35** (2005), no. 2, 486–517.
- [God81] C. D. Godsil, *Matchings and walks in graphs*, J. Graph Th. **5** (1981), 285–297.
- [HL72] O.J.Heilman and E.H.Lieb, *Theory of monomer-dimer systems*, Comm. Math. Phys. **25** (1972), 190–232.
- [Ham90] Y. O. Hamidoune, *On the numbers of independent k -sets in a claw free graph*, J. Combin. Theory Ser. B **50** (1990), no. 2, 241–244.
- [J03] M. Jerrum, *Counting, sampling and integrating: algorithms and complexity*, Lectures in Mathematics, ETH-Zurich, Birkhauser Verlag (2003).
- [JS97] M. Jerrum and A. Sinclair, *The Markov chain Monte Carlo method: an approach to approximate counting and integration*, Approximation algorithms for NP-hard problems (D. Hochbaum, ed.), PWS Publishing Company, Boston, MA, 1997.
- [JS] K. Jung and D. Shah, *Inference in Binary Pair-wise Markov Random Field through Self-Avoiding Walk*, Preprint on <http://arxiv.org/abs/cs.AI/0610111v2>.
- [JSV04] M. Jerrum, A. Sinclair, and E. Vigoda, *A polynomial-time approximation algorithms for permanent of a matrix with non-negative entries*, Journal of the Association for Computing Machinery **51** (2004), no. 4, 671–697.
- [JVV86] M. Jerrum, L. Valiant, and V. Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoret. Comput. Sci. **43** (1986), no. 2-3, 169188.
- [KK98] J. Kahn and J. H. Kim, *Random matchings in regular graphs*, Combinatorica **8** (1998), 201–226.
- [LSW00] N. Linial, A. Samorodnitsky, and A. Wigderson, *A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents*, Combinatorica **20** (2000), no. 4, 545–568.
- [MS06] A. Montanari and G. Semerjian, *Rigorous inequalities between length and time scales in glassy systems*, Preprint in arXiv.org (2006).
- [NT] C. Nair and P. Tetali, *The correlation decay (CD) tree and strong spatial mixing in multi-spin systems*, Preprint on <http://front.math.ucdavis.edu/math.PR/0701494>.
- [S93] A. Sinclair, *Algorithms for random generation and counting: a Markov chain approach*, Birkhauser Progress in Theoretical Computer Science Series, Birkhauser Verlag (1993).
- [Stan98] R. P. Stanley, *Graph colorings and related symmetric functions: ideas and applications*, Discrete Mathematics **193** (1998), 267-286.
- [V01] S. Vadhan, *The complexity of counting in sparse, regular, and planar graphs*, SIAM Journal on Computing **31** (2001), no. 2, pp. 398–427.
- [Val79] L. G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science, **8** (1979), 189–201.
- [vdB98] J. van den Berg, *On the absence of phase transition in the monomer-dimer model*, CWI reports, PNA-R9813, ISSN 1386-3711 (1998).
- [Wei06] D. Weitz, *Counting independent sets up to the tree threshold*, Proc. 38th Ann. Symposium on the Theory of Computing (2006).