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.

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],[GK07].

[*]Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510; e-mail: bayati@stanford.edu

[†]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

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

[¶]School of Mathematics and College of Computing, Georgia Tech, Atlanta, GA 30332-0160; e-mail: tetali@math.gatech.edu

1 Introduction

The focus of the paper is the problem of computing the total number of (full and partial) matchings of a given graph. This problem, along with many other combinatorial counting problems falls into the class of #P complexity class, 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 focus shifted to trying to find an approximating solution. Here by far the strongest and the most general method comes from the Monte Carlo Markov chain (MCMC) approach. Based on the equivalence between the counting problem and a related sampling problem [JVV86], this approach focuses on rapidly mixing Markov chains to obtain appropriate random samples. Many randomized approximation schemes for various counting problems were derived in this way – see e.g., Vigoda's survey [Vig00] for an extensive summary. Among other results, a fully polynomial *randomized* approximation scheme (FPRAS) for computing the total number of matchings of a given graph was provided by Jerrum and Sinclair [JS97]. Our main thrust in this work is in providing *deterministic* algorithms for counting the number of matchings; this we are able to do efficiently (in fully polynomial time) for the class of bounded degree graphs, and in sub-exponential time for general graphs (see details below).

Recently an alternative approach for constructing approximate counting schemes was developed, leading to *deterministic* approximation schemes for counting problems. The approach is based on the concept of *correlation decay* originating in statistical physics [Dob70] for estimating certain marginal probability distributions (See Proposition 2.2 below). The correlation decay concept is related to the concept of uniqueness of Gibbs measures in infinite lattices, hence the interest of statistical physicists in this subject. The idea of using correlation decay directly for counting was introduced in Weitz [Wei06] and Bandyopadhyay-Gamarnik [BG06], for the problems of counting the number of independent sets and colorings of a graph. 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 problem in the #P class, namely the problem of counting the number of independent sets in graphs with degree ≤ 5. Weitz's approach was based on a clever construction of a certain tree of self-avoiding walks and reducing the graph problem to that on a tree. Very recently, the correlation decay approach was also put to use in Gamarnik and Katz [GK07] for the problem of counting the number of proper colorings of a graph. In this work the step of constructing a self-avoiding walk was bypassed, by instead creating a certain *computation tree* in a dynamic programming type recursion and establishing correlation decay on the corresponding computation tree. As observed in [GK07], the advantage of establishing correlation decay on a computation tree has been highlighted in [TJ02] in the context of belief propagation algorithms and the Dobrushin uniqueness criterion.

In this paper we use the correlation decay approach for constructing a deterministic approximation scheme for counting the number of matchings of a graph. We also use the idea of proving correlation decay property on a computation tree. Surprisingly, the analysis becomes far more transparent, when compared to the counterparts in independent sets and colorings. In particular, we establish that the correlation decay property corresponding to counting matchings holds for an arbitrary graph, in an appropriate sense. We show that the rate of the correlation decay (appropriately defined) is $\approx 1-O(\frac{1}{\sqrt{\Delta}})$, where Δ is the maximum degree of the graph. As a result 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. 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 key new idea is the use of correlation decay to reduce the computation time using the recursions.

It should be noted that, while an FPRAS for counting matchings is known to exist, constructing deterministic counterparts is a far more challenging task. For example, while an FPRAS for computing the permanent (say, of a 0-1 matrix) is known [JSV04], the best known deterministic approximation scheme only gives factor e^n approximation [LSW00].

The connection between the correlation decay property and counting was realized within the field of rapidly mixing Markov chains as well. Broadly speaking one expects that the lack of long-range dependence (correlation decay) implies rapid mixing and thus the existence of randomized approximation scheme for counting/sampling problems. Specifically, when the graph satisfies a certain kind of sub-exponential growth condition the correlation decay does imply rapid mixing (see e.g., [DSVW02], [GMP05]). The converse, however, does not hold in general, as shown by Berger et al. [BKMP01], but does hold in some weaker sense, as shown recently by Montanari and Semerjian [MS06].

The present work, along with [BG06], [Wei06],[GK07],[BN06] reinforces this connection, as well as, broadly speaking, contributing to the exciting and emerging connection between theoretical computer science and statistical physics.

The rest of the paper has the following structure. Various definitions and the main result are presented in Section 2. The correlation decay analysis is the subject of Section 3. The approximate counting algorithm is presented in Section 4. The extension to the case of general graphs is discussed in Section 5. Some concluding remarks and further open questions are presented in Section 6.

2 Definitions, preliminaries and the main result

We consider a simple undirected graph $\mathbb{G}$ with n vertices, $V = \{v_1, v_2, \ldots, v_n\}$. Let E, Δ denote respectively the set of edges and the maximum degree of the graph. $N(v, \mathbb{G}) \subset V$ denotes the set of neighbors of v and $\Delta(v) = |N(v, \mathbb{G})|$ denotes the degree of the vertex v. The degree of the graph is $\Delta \triangleq \max_v \Delta(v)$. Let $\mathbb{G}_k = \mathbb{G} \setminus \{v_1, \ldots, v_{k-1}\}$ with $\mathbb{G}_0 = \mathbb{G}$. The order of vertices plays no particular role in our analysis, but we assume that a particular order is fixed.

A matching is a subset $M \subset E$ 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 $\mathbb{G}$. For every $k \leq |V|/2$ let $M(k)$ be the number of size k matchings in $\mathbb{G}$.

Given a fixed parameter $\lambda > 1$, called the *activity*, a natural (Gibbs) probability distribution on the set $\mathcal{M}$ of matchings may be defined using:

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

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

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

Denote by $\boldsymbol{M}$ a random matching selected according to the Gibbs measure $\mathbb{P}_{\mathbb{G}}$, which clearly depends on the given graph $\mathbb{G}$. The principle goal of this paper is obtaining an approximation of $\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})$ for every vertex v in polynomial time, for the class of bounded degree graphs $\mathbb{G}$. This quantity simply represents the probability that a random matching selected according to the Gibbs measure does *not* contain an edge adjacent to v.

Our goal is constructing an algorithm which computes $Z(\lambda, \mathbb{G})$ approximately. Let $|G|$ denote the size of the problem instance. Namely, $|G| = O(\max(|V|, |E|, \log \lambda))$. We will exclusively consider the case of constant λ. Then $|G| = O(n^2)$ for dense graphs and $|G| = O(n)$ for bounded degree graphs.

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

$$1 - \delta \leq \frac{\hat{Z}}{Z(\lambda, \mathbb{G})} \leq 1 + \delta,$$

in time which is polynomial in $n, \frac{1}{\delta}$.

We now state our main result.

Theorem 2.1. *There exist a deterministic algorithm which provides an FPTAS for computing $Z(\lambda, \mathbb{G})$ for an arbitrary graph/activity pair $(\mathbb{G}, \lambda)$ when Δ and λ are constants.*

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

We begin by establishing the following identity. This identity appears in various forms in the context of Markov chain sampling method as well. The identity is the essence of the cavity method of statistical physics.

Proposition 2.2. *The following identity holds*

$$Z(\lambda, \mathbb{G}) = \prod_{1 \leq k \leq |V|} \mathbb{P}^{-1}_{\mathbb{G}_k}(v_k \notin \boldsymbol{M}).$$

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

$$\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M}) = \frac{Z(\lambda, G \setminus \{v\})}{Z(\lambda, G)}. \tag{1}$$

Applying this identity recursively to $v_k, \mathbb{G}_k$, and noting that the partition function for the graph with no edges equals 1, we obtain the result. □

The above recursion for the partition function (matching polynomial) is the same as the one obtained by Godsil in [God81]. The following corollary is a straightforward application of Proposition 2.2 and therefore we shall focus our attention on constructing an algorithm for computing an approximation of $\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})$.

Corollary 2.3. *Given any $\epsilon > 0$, if there exists a fully polynomial time algorithm $\mathcal{A}$, which on input $(\mathbb{G}, v)$, computes a value $\hat{p}(v)$ satisfying*

$$\Big| \frac{\hat{p}(v) - \mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})}{\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})} \Big| \leq \frac{\epsilon}{n^2}, \tag{2}$$

then one immediately obtains a fully polynomial time approximation algorithm for $Z(\lambda, G)$.

3 Correlation decay analysis

A salient feature of the matching problem is that it allows a very simple recursive expression for the value $\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})$.

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

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

In particular,

$$\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M}) \geq \frac{1}{1+\lambda\Delta}. \tag{4}$$

Proof. With a slight abuse of notation we will write $v \in M$ if one of the edges incident to v belongs to the matching M. We have

$$Z(\lambda, \mathbb{G}) = \sum_{M \in \mathcal{M}(\mathbb{G}): v \notin M} \lambda^{|M|} + \sum_{u \in N(v,\mathbb{G})} \sum_{M \in \mathcal{M}(\mathbb{G}):(v,u)\in M} \lambda^{|M|}. \tag{5}$$

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 $\mathbb{G}\setminus\{v,u\}$ not containing u, $M \cup \{(v,u)\}$ creates a matching in $\mathbb{G}$ containing the edge (v,u). Thus we can rewrite (5) as

$$Z(\lambda, \mathbb{G}) = Z(\lambda, \mathbb{G}\setminus\{v\}) + \sum_{u \in N(v,\mathbb{G})} \lambda Z(\lambda, \mathbb{G}\setminus\{v,u\}). \tag{6}$$

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

For every subgraph $\hat{\mathbb{G}}$ of the graph $\mathbb{G}$ every vertex $v \in \hat{\mathbb{G}}$ and every $t \in \mathbb{Z}_+$ we introduce a quantity $\Phi_{\hat{\mathbb{G}}}(v,t)$ inductively as follows.

1. $\Phi_{\hat{\mathbb{G}}}(v,0) = 1$ for all $\hat{\mathbb{G}}, v$.
2. For every $t \geq 1$,

$$\Phi_{\hat{\mathbb{G}}}(v,t+1) = \frac{1}{1+\lambda \sum_{u \in N(v,\mathbb{G})} \Phi_{\hat{\mathbb{G}}\setminus\{v\}}(u,t)}. \tag{7}$$

The following is an immediate consequence of the definition of Φ: for every $t \geq 0$,

$$\frac{1}{1+\lambda\Delta} \leq \Phi_{\hat{\mathbb{G}}}(v,t) \leq 1. \tag{8}$$

While we have introduced the values $\Phi_{\hat{\mathbb{G}}}(v,t)$ 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 $\Phi_{\hat{\mathbb{G}}}(v,t)$ will serve as an approximation of $\mathbb{P}_{\hat{\mathbb{G}}}(v \notin \boldsymbol{M})$. The essence of this approximation is described in the following result.

Theorem 3.2 (Correlation Decay)**.** *The following holds for every vertex v and for time instance t, an even integer:*

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

Proof. Fix a vertex $v \in \mathbb{G}$, and let $N(v,\mathbb{G}) = \{u_1,\ldots,u_m\}$, $N(u_i, \mathbb{G}\setminus\{v\}) = \{w_1^{(i)},...,w_{m_i}^{(i)}\}$. 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:

$$x = \log \mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M}),\ x_i = \log \mathbb{P}_{\mathbb{G}\setminus\{v\}}(u_i \notin \boldsymbol{M}),\ x_j^{(i)} = \log \mathbb{P}_{\mathbb{G}\setminus\{v,u_i\}}(w_j^{(i)} \notin \boldsymbol{M}), \tag{9}$$

$$y = \log \Phi_{\mathbb{G}}(v,t),\ y_i = \log \Phi_{\mathbb{G}\setminus\{v\}}(u_i, t-1),\ y_j^{(i)} = \log \Phi_{\mathbb{G}\setminus\{v,u_i\}}(w_j^{(i)}, t-2), \tag{10}$$

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

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

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

Then we can rewrite (3) and (7) as

$$x = -f(\vec{x}),$$
$$y = -f(\vec{y}).$$

Applying the mean value theorem, there exists $\alpha \in [0,1]$ such that for $\vec{z}_\alpha = \alpha\vec{x} + (1-\alpha)\vec{y}$,

$$|x-y| = |\nabla f(\vec{z}_\alpha)'(\vec{x}-\vec{y})| \overset{(a)}{\leq} |\nabla f(\vec{z}_\alpha)|_{L_1}\ \|\vec{x}-\vec{y}\|_{L_\infty}, \tag{11}$$

where (a) follows from Hölder's inequality. It is easy to see that

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

Define $A_i = 1 + \lambda\sum_{j=1}^{m_i} e^{z_j^{(i)}}$. The $\mathbb{L}_1$-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 that the minimum occurs for some $0 < \frac{1}{A_i} < 1$ and obtain the saddle point equations by differentiating the logarithm of $\frac{1+\lambda\sum_{i=1}^m \frac{1}{A_i^2}}{1+\lambda\sum_{i=1}^m \frac{1}{A_i}}$.) 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 this to (11) we obtain

$$\Big|\log \mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M}) - \log \Phi_{\mathbb{G}}(v,t)\Big| \\ \leq \Big(1 - \frac{2}{(\sqrt{1+\lambda\Delta}+1)}\Big) \max_{i,j} \Big|\log \mathbb{P}_{\mathbb{G}\setminus\{v,u_i\}}(w_j^{(i)} \notin \boldsymbol{M}) - \log \Phi_{\mathbb{G}\setminus\{v,u_i\}}(w_j^{(i)}, t-2)\Big|.$$

Applying the same bound to the right-hand side, we obtain that $\Big|\log \mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M}) - \log \Phi_{\mathbb{G}}(v,t)\Big|$ is at most $(1 - \frac{2}{(\sqrt{1+\lambda\Delta}+1)})^{t/2}$ times $\max_{\hat{G},u} \Big|\log \mathbb{P}_{\hat{\mathbb{G}}}(u \notin \boldsymbol{M}) - \log \Phi_{\hat{\mathbb{G}}}(u,0)\Big|$, where the maximum is over all subgraph/vertex pairs $(\hat{\mathbb{G}} \subset \mathbb{G}, u \in \hat{\mathbb{G}})$. Applying (4) and (8), $\max_{\hat{G},u} \Big|\log \mathbb{P}_{\hat{\mathbb{G}}}(u \notin \boldsymbol{M}) - \log \Phi_{\hat{\mathbb{G}}}(u,0)\Big|$ is at most $\log(1+\lambda\Delta)$. □

4 Algorithm

Our algorithm is based on computing the values $\Phi_{\mathbb{G}}(v,t)$, for $t = O(\log n)$.

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

Proof. The first part follows immediately from recursion (7). The second part follows from the fact that $\Delta = O(1)$, a constant independent of n. □

Now based on this lemma, we propose the following algorithm for computing the partition function $Z(\lambda, \mathbb{G})$.

Algorithm CountMATCHINGS

```
INPUT: A graph/activity pair (𝔾, λ) and a positive integer t.
BEGIN
Set Ẑ = 1, 𝔾̂ = 𝔾.
While Ĝ ≠ ∅, find an arbitrary node v ∈ Ĝ.  Compute Φ_𝔾̂(v,t).
Set Ẑ = Φ_𝔾̂^{-1}(v,t) Ẑ,   Ĝ = Ĝ \ {v}.
END
OUTPUT: Ẑ.
```

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

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

$$1 - \epsilon\frac{\log(1+\lambda\Delta)}{n^2} \leq \frac{\Phi_{\mathbb{G}}(v,t)}{\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})} \leq 1 + \epsilon(1+\lambda\Delta)\frac{\log(1+\lambda\Delta)}{n^2}\,.$$

Proof. Theorem 3.2 implies

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

When $x \in [0, \beta]$, the following holds

$$1 - x \leq e^{-x} \leq e^{x} \leq 1 + e^{\beta} x \,,$$

and therefore one has

$$1 - \epsilon \frac{\log(1+\lambda\Delta)}{n^2} \leq \frac{\Phi_{\mathbb{G}}(v,t)}{\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})} \leq 1 + \epsilon(1+\lambda\Delta)\frac{\log(1+\lambda\Delta)}{n^2} \,.$$

□

Proof of Theorem 2.1. The algorithm providing FPTAS is CountMATCHINGS, with input $\mathbb{G}, \lambda$, and $t = O(\log n)$ as in Lemma 4.2. The proof is then obtained by combining Lemma 4.1, Lemma 4.2 and Corollary 2.3. □

5 Extensions to general graphs

The above technique can be used to approximate the partition function of arbitrary (i.e., unbounded degree) simple graphs using a deterministic $\exp(O(\sqrt{n}\log^2 n))$ algorithm.

Replacing Δ by n, we see from Lemma 4.2 that when $t = -2\left\lfloor (2\log n - \log\epsilon)/\log\left(1 - \frac{2}{(\sqrt{1+\lambda n}+1)}\right)\right\rfloor$ one has

$$1 - \epsilon \frac{\log(1+\lambda n)}{n^2} \leq \frac{\Phi_{\mathbb{G}}(v,t)}{\mathbb{P}_{\mathbb{G}}(v \notin \boldsymbol{M})} \leq 1 + \epsilon(1+\lambda n)\frac{\log(1+\lambda n)}{n^2} \,. \tag{12}$$

Theorem 5.1. *There exists a deterministic algorithm for computing $Z(\lambda, \mathbb{G})$ for an arbitrary graph/activity pair $(\mathbb{G}, \lambda)$ in $\exp(O(\sqrt{n}\log^2 n))$ time.*

Proof. The proof follows from Equation 12, Lemma 4.1 and Proposition 2.2. □

6 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.

Naturally, the following questions is raised: is there a deterministic FPTAS for counting matchings for the class of all graphs? There are 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), type arguments. In general, it is of interest to see to what extent the correlation decay approach can be used for solving other type of counting problems, such as counting the number of spanning trees and forests, counting the number of feasible solutions of bin-packing problems, and a host of other problems where 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 (FOCS) (2001).

[BN06] M. Bayati and C. Nair, *A rigorous proof of the cavity method for counting matchings*, Annual Allerton Conference on Communication, Control and Computing, 2006.

[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.

[DSVW02] 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), pp.461–479.

[GK07] 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.

[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.

[Jer06] M. Jerrum, *Counting, sampling and integrating: algorithms and complexity. Lecture notes, Chapter 3.*, 2006.

[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.

[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.

[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).

[TJ02] S. Tatikonda and M.I. Jordan, *Loopy belief propagation and Gibbs measures*, In Uncertainty in Artificial Intelligence (UAI), D. Koller and A. Darwiche (Eds)., 2002.

[Val79] L. G. Valiant, *The complexity of computing the permanent*, Theoretical computer science **8** (1979), 189–201.

[Vig00] E. Vigoda, *Improved bounds for sampling colorings*, Journal of Mathematical Physics (2000).

[Wei06] D. Weitz, *Counting down the tree*, Proc. 38th Ann. Symposium on the Theory of Computing (STOC) (2006).