Correlation decay and deterministic FPTAS for counting list-colorings of a graph

David Gamarnik [*] Dmitriy Katz [†]

May 16, 2006

Abstract

We propose a deterministic algorithm for approximately counting the number of list colorings of a graph. Under the assumption that the graph is triangle free, the size of every list is at least $\alpha\Delta$, where α is an arbitrary constant bigger than $\alpha^{**} = 2.8432\ldots$, the solution of $\alpha e^{-\frac{1}{\alpha}} = 2$, and Δ is the maximum degree of the graph, we obtain the following results. For the case when the size of the each list is a large constant, we show the existence of a *deterministic* FPTAS for computing the total number of list colorings. The same deterministic algorithm has complexity $2^{O(\log^2 n)}$, without any assumptions on the sizes of the lists, where n is the size of the instance.

Our results are not based on the most powerful existing counting technique – rapidly mixing Markov chain method. Rather we build upon concepts from statistical physics, in particular, the decay of correlation phenomena and its implication for the uniqueness of Gibbs measures in infinite graphs. This approach was proposed in two recent papers [BG06] and [Wei05]. The principle insight of the present work is that the correlation decay property can be established with respect to certain *computation tree*, as opposed to the conventional correlation decay property which is typically established with respect to graph theoretic neighborhoods of a given node. This allows truncation of computation at a logarithmic depth in order to obtain polynomial accuracy in polynomial time. While the analysis conducted in this paper is limited to the problem of counting list colorings, the proposed algorithm can be extended to an arbitrary constraint satisfaction problem in a straightforward way.

1 Introduction

This paper is devoted to the problem of computing the total number of list colorings of a graph. The setting is as follows. Each node of a given graph is associated with a list of colors. An assignment of nodes to colors is called list coloring if every node is assigned to some color from its list and no two nodes sharing an edge are assigned to the same color. When all the lists are identical, the problem reduces to the problem of coloring of a graph. The problem of determining whether a list coloring exists is NP-hard, but provided that the size of each list is stictly larger than the degree for each node, a simple greedy algorithm produces a coloring. We are concerned with the corresponding counting problem – compute the total number of list colorings of a given graph/list pair. This problem is known to be $\#P$ hard even for the restricted problem of counting the colorings, and the focus is on the approximation algorithms. The existing approximation schemes have primarily been developed for the coloring problem and are mostly based on the rapidly mixing Markov chain technique, also known as Glauber dynamics

[*]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

approach. It was established by Jerrum [Jer95] that the Glauber dynamics corresponding to graphs where the ratio of the number of colors to degree satisfies $q/\Delta \geq 2$, mixes rapidly. This leads to a randomized approximation algorithm for enumerating the number of colorings. The 2-barrier was first broken by Vigoda [Vig00], who lowered the ratio requirement to 11/6. Many further significant improvements were obtained subsequently. The state of the art is summarized in [FV06]. For a while the improvement over 11/6 ratio came at a cost of lower bound $\Omega(\log n)$ on the maximum degree, where n is the number of nodes. Recently, this requirement was lifted by Dyer et al. [DFHV04].

In this paper we focus on a different approach to the counting list colorings problem. Our setting is a list coloring problem. We require that the size of every list is at least $\alpha\Delta + \beta$, where $\alpha > \alpha^{**}$ - the unique solution to $\alpha e^{-\frac{1}{\alpha}} = 2$, and β is a large constant. Our girth restriction is $g \geq 4$, namely, the graph is triangle-free. We obtain the following results. First, assuming that the size of each list is at most a constant, we construct a *deterministic* Fully Polynomial Time Approximation Scheme (FPTAS) for the problem of computing the total number of list colorings of a given graph/list pair. Second, for arbitrary graph/list pair (no assumptions on the list sizes) we construct an approximation algorithm with complexity $2^{O(\log^2 n)}$. Namely, our algorithm is super-polynomial but still significantly quicker than exponential time.

Although our regime $\alpha > 2.8432\ldots$ is weaker than $q/\Delta > 2$, for which the Markov chain is known to mix rapidly, our method has advantages over the Markov chains based method. The most notable is the deterministic nature of the algorithm, which eliminates the sampling error. Also, to the best of our knowledge, we propose the first algorithm for list colorings of a graph in a sufficiently general setting.

Our approach is based on establishing a certain correlation decay property which has been considered in many settings [BW04], including the coloring problem [SS97], [GMP05], [BW02],[Jon02] and has been recently a subject of interest. In particular, the correlation decay has been established in [GMP05] for coloring triangle-free graphs under the assumption that $\alpha > \alpha^* = 1.763\ldots$, the unique solution of $\alpha e^{-\frac{1}{\alpha}} = 1$. (Some mild additional assumptions were adopted). The principal motivation for establishing correlation decay property comes from statistical physics, in particular the connection between the correlation decay and the uniqueness of the associated Gibbs measure (uniform measure in our setting) on infinite versions of the graph, typically lattices. Recently, however, a new approach linking correlation decay to counting algorithms was proposed in Bandyopadhyay and Gamarnik [BG06] and Weitz [Wei05]. The idea is to use correlation decay property instead of Markov sampling for computing marginals of the Gibbs (uniform) distribution. This leads to a deterministic approach since the marginals are computed using a dynamic programming like scheme (also known as Belief Propagation (BP) algorithm [YFW00]). This approach typically needs a locally-tree like structure (large girth) [Sha05] in order to be successful. The large girth assumption was explicitly assume in [BG06], where the problems of computing the number of independent sets and colorings in some special structured (regular) graphs was considered. Weitz [Wei05] cleverly by-passes the large girth assumption by using a certain self-avoiding tree construction thus essentially reducing the problem to a problem on a tree with careful boundary conditions implied by independent sets. This idea was used recently by Jung and Shah [JS06] to introduce a version of a BP algorithm which works on a non-locally-tree like graphs, where appropriate correlation decay can be established. This approach works for binary type problems (independent sets, matchings, Ising model) but does not apparently extend to multi-valued problems.

In this paper we propose a general deterministic approximate counting algorithm which can be used for arbitrary multi-valued counting problem, although we analyze the approach only for the case of counting list colorings. We also by-pass the large girth assumption by considering a certain *computation tree* corresponding to the Gibbs (uniform for the case of colorings) measure. Our principal insight is establishing correlation decay for the certain associated *computation tree* as opposed to the conventional correlation decay associated with the graph-theoretic structure of the graph. We provide a discussion explaining why it is crucial to establish the correlation decay in this way in order to obtain FPTAS. Contrast this with [GMP05] where correlation decay is established for the coloring problem but in the conventional graph-theoretic distance sense. The advantage of establishing correlation decay on a computation tree as opposed to the original graph has been highlighted also in [TJ02] in the context of BP algorithms and the Dobrushin's Uniqueness condition.

The remainder of the paper has the following structure. The model description and the main result are stated in Section 2. Some preliminary technical results are established in Section 3. The description of the algorithm and its complexity are subject of Section 4. The principal technical result is established in Section 5. The key result is Theorem 2, which establishes the correlation decay result on a computation tree arising in computing the marginals of the uniform distribution on the set of all list colorings. Section 6 provides a brief comparison between the correlation decay on a computation tree and the correlation decay in a conventional sense. Some conclusions and open problems are in Section 7.

2 Definitions and the main result

We consider a simple graph $\mathbb{G}$ with the node set $V = \{v_1, v_2, \ldots, v_{|V|}\}$. Our graph is assumed to be triangle-free. Namely the girth (the size of the smallest cycle) is at least $g \geq 4$. Let E, Δ denote respectively the set of edges and the maximum degree of the graph. We also let $\Delta(v)$ denote the degree of the node v. Each node v is associated with a list of colors $L(v) \subset \{1, 2, \ldots, q\} = \cup_{v\in V}|L(v)|$, where $\{1, 2, \ldots, q\}$ is the total universe of colors. We let $\boldsymbol{L} = (L(v), 1 \leq v \leq n)$ denote the vector of lists. We also let $\|\boldsymbol{L}\| = \max_v |L(v)|$ the size of the largest list. The list-coloring problem on $\mathbb{G}$ is formulated as follows: associate each node v with a color $c(v) \in L(v)$ such that no two nodes sharing an edge are associated with the same color. When all the lists are identical and contain q elements, the corresponding problem is the problem of coloring $\mathbb{G}$ using q colors. We let $|L(v)|$ denote the cardinality of $L(v)$. It is easy to see that if

$$|L(v)| \geq \Delta(v) + 1 \tag{1}$$

for every node v, then a simple greedy procedure produces a list-coloring. We adopt here a stronger assumption

$$|L(v)| \geq \alpha\Delta(v) + \beta, \tag{2}$$

where α is an arbitrary constant strictly larger than α^{**}, the unique solution of $\alpha^{**} = \exp(-\frac{1}{\alpha*}) \approx 2.8432\ldots$ and β is a large constant depending on α.

Let $Z(\mathbb{G}, \boldsymbol{L})$ denote the total number of possible list-colorings of a graph/list pair $(\mathbb{G}, \boldsymbol{L})$. The corresponding counting problem is to compute (approximately) $Z(\mathbb{G}, \boldsymbol{L})$. In statistical physics terminology, $Z(\mathbb{G}, \boldsymbol{L})$ is the partition function. We let $Z(\mathbb{G}, \boldsymbol{L}, \phi)$ denote the number of list colorings of $(\mathbb{G}, \boldsymbol{L})$ which satisfy some condition ϕ. For example $Z(\mathbb{G}, \boldsymbol{L}, c(v) = i, c(u) = j)$ is the number of list colorings such that the color of v is i and the color of u is j.

On the space of all list colorings of $\mathbb{G}$ we consider a uniform probability distribution, where each list coloring assumes weight $1/Z(\mathbb{G}, \boldsymbol{L})$. For every node/color pair $v \in V, i \in L(v)$, $\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = i)$ denotes the probability that node v is colored i with respect to this probability measure. The size of the instance corresponding to a graph/list pair$(\mathbb{G}, \boldsymbol{L})$ is defined to be $n = \max\{|V|, |E|, q\}$.

Definition 1. *An approximation algorithm $\mathcal{A}$ is defined to be a Fully Polynomial Time Approximation*

Scheme for a computing $Z(\mathbb{G}, \boldsymbol{L})$ if given arbitrary $\delta > 0$ it produces a value $\hat{Z}$ satisfying

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

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

We now state our main result.

Theorem 1. *There exist a deterministic algorithm which provides a FPTAS for computing $Z(\mathbb{G}, \boldsymbol{L})$ for arbitrary graph list pair $\mathbb{G}, \boldsymbol{L}$ satisfying (2), when the size of the largest list $\|\boldsymbol{L}\|$ is constant. The same algorithm has complexity $2^{O(\log^2 n)}$, without any restriction on $\|\boldsymbol{L}\|$, where n is the size of the instance.*

3 Preliminary technical results

3.1 Basic recursion

We begin by establishing a standard relationship between the partition function $Z(\mathbb{G}, \boldsymbol{L})$ and the marginals $\mathbb{P}_{\mathbb{G}, \boldsymbol{L}}(c(v) = i)$. The relation, also known as cavity method, is also the basis of the Glauber dynamics approach for computing partition functions.

Proposition 1. *Consider an arbitrary list coloring $i_1, \ldots, i_{|V|}$ of the graph $\mathbb{G}$ (which can be constructed using a simple greedy procedure). For every $k = 0, 1, \ldots, |V| - 1$ consider a graph list pair $\mathbb{G}_k, \boldsymbol{L}_k$, where $(\mathbb{G}_0, \boldsymbol{L}_0) = (\mathbb{G}, \boldsymbol{L})$, $\mathbb{G}_k = \mathbb{G} \setminus \{v_1, \ldots, v_k\}, k \geq 1$ and the list $\boldsymbol{L}_k$ is obtained by deleting from each list $L(v_l), l > k$ a color $i_r, r \leq k$ if $(v_l, v_r) \in E$. Then*

$$Z(\mathbb{G}, \boldsymbol{L}) = \prod_{0 \leq k \leq |V|-1} \mathbb{P}^{-1}_{\mathbb{G}_k, \boldsymbol{L}_k}(c(v_k) = i_k).$$

Proof. We have

$$\mathbb{P}_{\mathbb{G}, \boldsymbol{L}}(c(v_1) = i_1) = \frac{Z(\mathbb{G}, \boldsymbol{L}, c(v_1) = i_1)}{Z(\mathbb{G}, \boldsymbol{L})} = \frac{Z(\mathbb{G}_1, \boldsymbol{L}_1)}{Z(\mathbb{G}, \boldsymbol{L})},$$

from which we obtain

$$Z(\mathbb{G}, \boldsymbol{L}) = \mathbb{P}_{\mathbb{G}, \boldsymbol{L}}(c(v_1) = i_1)^{-1} Z(\mathbb{G}_1, \boldsymbol{L}_1).$$

Iterating further for $k \geq 2$ we obtain the result. □

Our algorithm is based on a recursive procedure which relates the number of list colorings of a given graph/list pair in terms of the number of list colorings of some reduced graph/list pairs.

Given a pair $(\mathbb{G}, \boldsymbol{L})$ and a node $v \in \mathbb{G}$, let $v_1, \ldots, v_m$ be the set of neighbors of v. For every pair $(k, i) \in \{1, \ldots, m\} \times L(v)$ we define a new pair $(\mathbb{G}_v, \boldsymbol{L}_{k,i})$ as follows. The set of nodes of $\mathbb{G}$ is $V_k = V \setminus \{v\}$ and $L_{k,i}(v_r) = L(v_r) \setminus \{i\}$ for $1 \leq r < k$, $L_{k,j}(u) = L(u)$ for all other u. Namely, we first delete node v from the graph. Then we delete color i from the lists corresponding to the nodes $v_r, r < k$, and leave all the other lists intact.

Lemma 1. *The graph/list pair $(\mathbb{G}_v, \boldsymbol{L}_{k,j})$ satisfies (2) for every $1 \leq k \leq m, j \in L(v)$, provided that $(\mathbb{G}, \boldsymbol{L})$ does.*

Proof. When we create graph $\mathbb{G}_v$ from $\mathbb{G}$ the list size of every remaining node either stays the same or is reduced by one. The second event can only happen for neighbors $v_1, \ldots, v_m$ of the deleted node v. When the list is reduced by one the degree is reduced by one as well. Since $\alpha > 1$, the assertion follows by observing that $|L(v_k)| \geq \alpha\Delta(v_k) + \beta$ implies $|L(v_k)| - 1 \geq \alpha(\Delta(v_k) - 1) + \beta$. $\square$

The basis of our algorithm is the following simple result.

Proposition 2. *Given a graph/list pair $(\mathbb{G}, \boldsymbol{L})$ and a node v, suppose $\Delta(v) = m > 0$. For every $i \in L(v)$*

$$\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = i) = \frac{\prod_{1\leq k\leq m}(1 - \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}_{k,i}}(c(v) = i))}{\sum_{j\in L(v)} \prod_{1\leq k\leq m}(1 - \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}_{k,j}}(c(v) = j))}. \tag{3}$$

The recursion as well as the proof is similar to the one used by Weitz in [Wei05], except we bypass the construction of a self-avoiding tree, considered in [Wei05].

Proof. Consider a graph/list $(\mathbb{G}_v, \boldsymbol{L})$ obtained simply by removing node v from $\mathbb{G}$, and leaving $\boldsymbol{L}$ intact for the remaining nodes. We have

$$\begin{aligned}
\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = i) &= \frac{\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = i)}{\sum_{j\in L(v)} \mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = j)} \\
&= \frac{Z(\mathbb{G}, \boldsymbol{L}, c(v) = i)Z^{-1}(\mathbb{G}, \boldsymbol{L})}{\sum_{j\in L(v)} Z(\mathbb{G}, \boldsymbol{L}, c(v) = j)Z^{-1}(\mathbb{G}, \boldsymbol{L})} \\
&= \frac{Z(\mathbb{G}_v, \boldsymbol{L}, c(v_k) \neq i,\ 1 \leq k \leq m)}{\sum_{j\in L(v)} Z(\mathbb{G}_v, \boldsymbol{L}, c(v_k) \neq j,\ 1 \leq k \leq m)} \\
&= \frac{\mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_k) \neq i,\ 1 \leq k \leq m)}{\sum_{j\in L(v)} \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_k) \neq j,\ 1 \leq k \leq m)}
\end{aligned}$$

Now, for every $j \in L(v)$

$$\mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_k) \neq j,\ 1 \leq k \leq m) = \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_1) \neq j) \prod_{2\leq k\leq m} \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_k) \neq j | c(v_r) \neq j,\ 1 \leq r < k)$$

We observe that $\boldsymbol{L}_{1,j} = \boldsymbol{L}$ for every j (no colors are removed due to the vacuous condition $r < 1$), and $\mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_k) \neq j | c(v_r) \neq j,\ 1 \leq r < k) = \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}_{k,j}}(c(v_k) \neq j)$. Namely

$$\mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}}(c(v_k) \neq j,\ 1 \leq k \leq m) = \prod_{1\leq k\leq m} \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}_{k,j}}(c(v_k) \neq j) = \prod_{1\leq k\leq m} (1 - \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}_{k,j}}(c(v_k) = j)).$$

Substituting this expression we complete the proof. $\square$

3.2 Upper and lower bounds

The condition (2) allows us to obtain the following simple bounds.

Lemma 2. *For every $\mathbb{G}, \boldsymbol{L}$, node v and a color $i \in L(v)$*

$$\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = i) \leq \frac{1}{\beta}.$$

Proof. Observe that given an arbitrary coloring of the neighbors $v_1, \dots, v_m$ of v, there are at least $|L(v)| - \Delta(v) \ge \beta$ colors remaining. Then the upper bound holds. $\square$

From this simple bound we now establish a different upper bound and also a lower bound using the triangle free assumption.

Lemma 3. *There exist $\epsilon_0 = \epsilon_0(\alpha) \in (0,1)$ and $\beta > 0$ such that for every $\mathbb{G}, \boldsymbol{L}$, node v and a color $i \in L(v)$*

$$q^{-1}(1-\beta^{-1})^{\Delta} \le \mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i) \le \frac{1}{2\Delta(v)(1+\epsilon_0)}.$$

We note that the upper bounds of this lemma and Lemma 2 are not comparable, since values of $\Delta(v)$ could be smaller and larger than β.

Proof. We let $v_1, \dots, v_m$ denote the neighbors of v, $m = \Delta(v)$ and let v_{kr} denote the set of neighbors of v_k, other than v for $k = 1, \dots, m$. We will establish that for any coloring of nodes (v_{kr}), which we generically denote by $\boldsymbol{c}$, we have

$$q^{-1}(1-\beta^{-1})^{\Delta} \le \mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i|\boldsymbol{c}) \le \frac{1}{2m(1+\epsilon_0)}.$$

The corresponding inequality for the unconditional probability then follows immediately. Now observe that, since the girth is at least 4, then there are no edges between v_k. Then $\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i|\boldsymbol{c})$ is the probability $\mathbb{P}_{\mathbb{T}}(c(v)=i)$ that v is colored i in a depth-1 tree $\mathbb{T} \triangleq \{v, v_1, \dots, v_m\}$, where the lists $\hat{L}(v_k)$ of v_k are obtained from $L(v_k)$ by deleting the colors used by the neighbors v_{kr} by coloring $\boldsymbol{c}$. From the assumption (2) we have that the remaining lists $\hat{L}(v_k)$ have size at least $|L(v_k)| - \Delta(v_k) \ge \beta$ each. Let $t_i = \mathbb{P}_{\mathbb{T}}(c(v)=i)$. For each color $j \in L(v)$ let $t_{j,k} = 1/|\hat{L}(v_k)|$ if $j \in \hat{L}(v_k)$ and $=0$ otherwise. Proposition 2 then simplifies to

$$t_i = \frac{\prod_{1\le k\le m}(1-t_{i,k})}{\sum_{j\in L(v)}\prod_{1\le k\le m}(1-t_{j,k})} \le \frac{1}{\sum_{j\in L(v)}\prod_{1\le k\le m}(1-t_{j,k})}, \tag{4}$$

for every $i \in L(v)$, where $\prod_{1\le k\le m}$ is defined to be equal to unity when $m=0$. From the equality part, applying $t_{j,k} \le 1/\beta$, we get

$$t_i \ge |L(v)|^{-1}(1-\beta^{-1})^m \ge q^{-1}(1-\beta^{-1})^{\Delta},$$

and the lower bound is established.

We now focus on the upper bound and use the inequality part of (4). Thus it suffices to show that

$$\sum_{j\in L(v)}\prod_k (1-t_{j,k}) \ge 2(1+\epsilon_0)m \tag{5}$$

for some constant $\epsilon_0 > 0$. Using the first order Taylor expansion for $\log z$ around $z=1$,

$$\begin{aligned}\prod_{1\le k\le m}(1-t_{j,k}) &= \prod_{1\le k\le m} e^{\log(1-t_{j,k})} \\ &= \prod_{1\le k\le m} e^{-t_{j,k}-\frac{1}{2(1-\theta_{j,k})^2}t_{j,k}^2},\end{aligned}$$

for some $0 \le \theta_{j,k} \le t_{j,k}$, since $-1/z^2$ is the second derivative of $\log z$. Again using the bound $t_{j,k} \le 1/\beta$, we have $(1-\theta_{j,k})^2 \ge (1-1/\beta)^2$. We assume that β is a sufficiently large constant ensuring $(1-1/\beta)^2 > 1/2$. Thus we obtain the following lower bound

$$\prod_{1\le k\le m}(1-t_{j,k}) \ge \prod_{1\le k\le m} e^{-t_{j,k}-\frac{t_{j,k}^2}{2(1-1/\beta)^2}} \ge e^{-(1+\frac{1}{\beta})\sum_k t_{j,k}} \triangleq e^{-(1+\frac{1}{\beta})T_j},$$

where T_j stands for $\sum_k t_{j,k}$. Then

$$\sum_{j\in L(v)}\prod_{1\le k\le m}(1-t_{j,k}) \ge \sum_{j\in L(v)} e^{-(1+\frac{1}{\beta})T_j} \ge |L(v)|e^{-\frac{1}{|L(v)|}(1+\frac{1}{\beta})\sum_j T_j},$$

where we have used an inequality between the average arithmetic and average geometric. Finally we observe

$$\sum_{j\in L(v)} T_j = \sum_{j,k} t_{j,k} = \sum_{1\le k\le m}\sum_{j\in \hat{L}(v_k)} \frac{1}{|\hat{L}(v_k)|} = m.$$

Thus

$$\sum_{j\in L(v)}\prod_{1\le k\le m}(1-t_{j,k}) \ge |L(v)|e^{-\frac{m}{|L(v)|}(1+\frac{1}{\beta})} \ge (\alpha m+\beta)e^{-\frac{1}{\alpha}(1+\frac{1}{\beta})} > \alpha m e^{-\frac{1}{\alpha}(1+\frac{1}{\beta})}$$

The condition $\alpha > \alpha^{**}$ implies that there exists a sufficiently large β such that $\alpha e^{-\frac{1}{\alpha}(1+\frac{1}{\beta})} > 2$. We find $0 < \epsilon_0 < .1$ such that $\alpha e^{-\frac{1}{\alpha}(1+\frac{1}{\beta})} = 2(1+\epsilon_0)$. We obtain a required lower bound (5). □

4 Algorithm and complexity

4.1 Description of an algorithm

Our algorithm is based on the idea of trying to approximate the value of $\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i)$, by performing a certain recursive computation using (3) a fixed number of times d and then using a correlation decay principle to guarantee the accuracy of the approximation. Specifically, introduce a function Φ which takes as an input a vector $(\mathbb{G},\boldsymbol{L},v,i,d)$ and takes some values $\Phi(\mathbb{G},\boldsymbol{L},v,i,d) \in [0,1]$. The input $(\mathbb{G},\boldsymbol{L},v,i,d)$ to Φ is any vector, such that such that v is a node in $\mathbb{G}$, i is an arbitrary color, and d is an arbitrary non-negative integer. Function Φ is defined recursively in d. The quantity Φ "attempts" to approximate $\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i)$. The quality of the approximation is controlled by d. We define Φ as follows. For every input $(\mathbb{G},\boldsymbol{L},v,i,d)$ such that $i \notin L(v)$ we set $\Phi(\mathbb{G},\boldsymbol{L},v,i,d) = 0$. Otherwise we set the values as follows.

- When $d = 0$, we set $\Phi(\mathbb{G},\boldsymbol{L},v,i,d) = 1/|L(v)|$ for every input $(\mathbb{G},\boldsymbol{L},v,i)$. (It turns out that for our application the initialization values are not important, due to the decay of correlations).
- For every $d \ge 1$, if $\Delta(v) = 0$, then $\Phi(\mathbb{G},\boldsymbol{L},v,i,d) = 1/|L(v)|$ for all $i \in L(v)$. Suppose $\Delta(v) = m > 0$ and $v_1,\ldots,v_m$ are the neighbors of v. Then for every $i \in L(v)$ we define

$$\Phi(\mathbb{G},\boldsymbol{L},v,i,d) = \min\Big[\frac{1}{2(1+\epsilon_0)m},\frac{1}{\beta},\frac{\prod_{1\le k\le m}(1-\Phi(\mathbb{G}_v,\boldsymbol{L}_{k,i},v_k,i,d-1))}{\sum_{j\in L(v)}\prod_{1\le k\le m}(1-\Phi(\mathbb{G}_v,\boldsymbol{L}_{k,j},v_k,j,d-1))}\Big]. \quad (6)$$

The last part of the expression inside min[$\cdot$] corresponds directly to the expression (3) of Proposition 2. Specifically, if it was true that $\Phi(\mathbb{G}_v, \boldsymbol{L}_{k,j}, v_k, j, d-1) = \mathbb{P}_{\mathbb{G}_v, \boldsymbol{L}_{k,j}}(c(v_k) = j)$, then, by Lemmas 2,3, the minimum in (6) would be achieved by the third expression, and then the value of $\Phi(\mathbb{G}, \boldsymbol{L}, v, i, d)$ would be exactly $\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v) = i)$.

We will use the correlation decay property to establish that the difference between the two values, modulo rescaling, is diminishing as $d \to \infty$. Note that the computation of Φ can be done recursively in d and it involves a dynamic programming type recursion. The underlying computation is done essentially on a tree of graph list pairs $\mathbb{G}_s, \boldsymbol{L}_s$ generated during the recursion. We refer to this tree as *computation tree* with depth d.

We now describe our algorithm for approximately computing $Z(\mathbb{G}, \boldsymbol{L})$. The algorithm is parametrized by the "quality" parameter d.

Algorithm CountCOLOR

`INPUT:` `A graph/list pair` $(\mathbb{G}, \boldsymbol{L})$ `and a positive integer` d.
`BEGIN`
`Set` $\hat{Z} = 1, \hat{\mathbb{G}} = \mathbb{G}, \hat{\boldsymbol{L}} = \boldsymbol{L}$.
`While` $\hat{G} \neq \emptyset$, `find an arbitrary node` $v \in \hat{G}$ `and a color` $i \in \hat{L}(v)$. `Compute`

$$\hat{p}(v, i) \triangleq \Phi(\hat{G}, \hat{\boldsymbol{L}}, v, i, d). \tag{7}$$

`Set` $\hat{Z} = \hat{p}^{-1}(v, i)\hat{Z}, \hat{G} = \hat{G} \setminus \{v\}, \hat{L}(u) = \hat{L}(u) \setminus \{i\}$ `for all neighbors` u `of` v `in` $\hat{G}$, `and` $\hat{L}(u)$ `remains the same for all other nodes.`
`END`
`OUTPUT:` $\hat{Z}$.

4.2 Some properties

We now establish some properties of Φ.

Lemma 4. *The following holds for every* $\mathbb{G}, \boldsymbol{L}, v, i \in L(v), d \geq 0$.

$$\Phi(\mathbb{G}, \boldsymbol{L}, v, i, d) \leq \min\Big[\frac{1}{\beta}, \frac{1}{2(1+\epsilon_0)\Delta(v)}\Big], \tag{8}$$

$$\sum_{i \in L(v)} \Phi(\mathbb{G}, \boldsymbol{L}, v, i, d) \leq 1, \tag{9}$$

$$\Phi(\mathbb{G}, \boldsymbol{L}, v, i, d) \geq q^{-1}(1 - 1/\beta)^{\Delta}. \tag{10}$$

Proof. (9) follows directly from the definition of Φ. To show (8) we consider cases. For $d \geq 1$ this follow directly from the recursion (6). For $d = 0$, this follows since $\Phi(\mathbb{G}, \boldsymbol{L}, v, i, 0) = 1/|L(v)| \leq 1/(\alpha\Delta(v) + \beta)$ and $2(1+\epsilon_0) < 2.2 < \alpha$. We now establish (10). For the case $d = 0$ this follows since $1/|L(v)| \geq 1/q$. For the case $d \geq 1$ this follows from the recursion (6) since $1/\beta, 1/(2(1+\epsilon_0)\Delta(v)) > 1/q$ and the third term inside the minimum operator is at least $q^{-1}(1 - 1/\beta)^{\Delta}$, using upper bound $\Phi(\mathbb{G}, \boldsymbol{L}, v, i, d-1) \leq 1/\beta$ which we have from (8).

□

4.3 Complexity

We begin by analyzing the complexity of computing function Φ. Recall that $n = \max(|V|, |E|, q)$ is the size of the instance.

Proposition 3. *For any given node v, the function Φ can be computed in time $2^{O(d \log \|L\| \Delta)}$. In particular when $d = O(\log n)$, the overall computation is $2^{O(\log^2 n)}$. If in addition the size of the largest list $\|L\|$ is constant then the computation time is polynomial in n.*

Proof. Let $\phi(d)$ denote the complexity of computing function $\Phi(\cdot, d)$. Clearly, $\phi(0) = O(\|L\|)$. We now express $\phi(d)$ in terms of $\phi(d-1)$. Given a node v, in order to compute $\Phi(\mathbb{G}, \boldsymbol{L}, v, i, d)$ we first identify the neighbors $v_1, \ldots, v_m$ of v. Then we create graph/list pairs $\mathbb{G}_v, \boldsymbol{L}_{j,k}, 1 \le k \le m, j \in L(v)$, compute $\Phi(\cdot, d-1)$ for each of this graphs, and use this to compute $\Phi(\mathbb{G}, \boldsymbol{L}, v, i, d)$. The overall computation effort is then

$$\phi(d) = O(\|L\| \Delta \phi(d-1)).$$

Iterating over d we obtain $\phi(d) = O(\|L\|^{d+1} \Delta^d) = O(2^{(d+1) \log \|L\| \Delta}) = 2^{O(d \log \|L\| \Delta)}$. When $d = O(\log n)$, we obtain a bound $2^{O(\log^2 n)}$. If in addition $\|L\| = O(1)$, then the assumption (2) implies $\Delta = O(1)$, and then $\phi(d) = n^{O(1)}$. □

The following is then immediate.

Corollary 1. *Suppose $d = O(\log n)$. Then the complexity of the algorithm CountCOLOR is $2^{O(\log^2 n)}$. If in addition the size of the largest list $\|L\|$ is constant, then CountCOLOR is a polynomial time algorithm.*

5 Correlation decay

The following is the key correlation decay result.

Theorem 2. *Consider a triangle-free graph/list pair $(\mathbb{G}, \boldsymbol{L})$ satisfying (2). There exist constants $0 < \epsilon < 1, \beta > 0, c > 0$ which depend only on α, such that for all nodes v, colors $i \in L(v)$ and $d \ge 0$*

$$\max_{i \in L(v)} \Big| \log \mathbb{P}_{\mathbb{G}, \boldsymbol{L}}(c(v) = i) - \log \Phi(\mathbb{G}, \boldsymbol{L}, v, i, d) \Big| \le cn(1-\epsilon)^d. \tag{11}$$

This theorem is our key tool for using the values of Φ for computing the marginals $\mathbb{P}_{\mathbb{G}, \boldsymbol{L}}(c(v) = i)$. We first establish that this correlation decay result implies our main result, Theorem 1.

Proof of Theorem 1. We consider an arbitrary instance $(\mathbb{G}, \boldsymbol{L})$ with size n and arbitrary $\delta > 0$. We may assume without the loss of generality that n is at least a large constant bigger than C/δ, for any universal constant C, since we can simply extend the size of the instance by adding isolated nodes. The proof uses a standard idea of approximating marginals $\mathbb{P}_{\mathbb{G}, \boldsymbol{L}}(c(v) = i)$ and then using Proposition 1 for computing $Z(\mathbb{G}, \boldsymbol{L})$. From Proposition 1, if the algorithm CountCOLOR produces in every stage $k = 1, 2, \ldots, |V| - 1$ a value $\hat{p}(v, i)$ which approximates $\mathbb{P}_{\mathbb{G}_v, \boldsymbol{L}_k}(c(v_k) = i)$ with accuracy

$$1 - \frac{\delta}{n} \le \frac{\hat{p}(v, i)}{\mathbb{P}_{\mathbb{G}_v, \boldsymbol{L}_k}(c(v_k) = i)} \le 1 + \frac{\delta}{n} \tag{12}$$

then the output $\hat{Z}$ of the algorithm satisfies

$$\Big(1-\frac{\delta}{n}\Big)^n \le \Big(1-\frac{\delta}{n}\Big)^{|V|} \le \frac{Z(\mathbb{G},\boldsymbol{L})}{\hat{Z}} \le \Big(1+\frac{\delta}{n}\Big)^{|V|} \le \Big(1+\frac{\delta}{n}\Big)^n$$

Since $|V| \le n$ and n is at least a large constant, we obtain an arbitrary accuracy of the approximation. Thus it suffices to arrange for (12). We run the algorithm CountCOLOR with $d = \lceil \frac{3\log n}{\log\frac{1}{1-\epsilon}} \rceil$, where ϵ is the constant from Theorem 2. This choice of d gives $(1-\epsilon)^d \le 1/n^3$. Theorem 2 with the given value of d then implies

$$\Big|\log\frac{\mathbb{P}_{\hat{\mathbb{G}},\hat{\boldsymbol{L}}}(c(v)=i)}{\hat{p}(v,i)}\Big| = \Big|\log\frac{\mathbb{P}_{\hat{\mathbb{G}},\hat{\boldsymbol{L}}}(c(v)=i)}{\Phi(\hat{\mathbb{G}},\hat{\boldsymbol{L}},v,i,d)}\Big| \le O(n)\frac{1}{n^3} = O(\frac{1}{n^2}).$$

Thus

$$1-O(\frac{1}{n^2}) \le \exp\big(-O(\frac{1}{n^2})\big) \le \frac{\hat{p}(v,i)}{\mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i)} \le \exp\big(O(\frac{1}{n^2})\big) = 1+O(\frac{1}{n^2})$$

This gives us (12) for all $n > C/\delta$ where C is the universal constant appearing in $O(\cdot)$. This completes the analysis of the accuracy. The complexity part of the theorem follows directly from Corollary 1. □

The rest of the section is devoted to establishing this Theorem 2. The basis of the proof is the recursion (3). As before, let $v_1,\dots,v_m$ be the neighbors of v in $\mathbb{G}$, $m=\Delta(v)$. Observe that (11) holds trivially when $m=0$, since both expression inside the absolute value become $1/|L(v)|$ and the left-hand side becomes equal to zero. Thus we assume that $m \ge 1$. Denote by m_k the degree of v_k in the graph $\mathbb{G}_v$. In order to ease the notations, we introduce

$$\begin{aligned} x_i &= \mathbb{P}_{\mathbb{G},\boldsymbol{L}}(c(v)=i), \quad i \in L(v), \\ x_{i,k} &= \mathbb{P}_{\mathbb{G}_v,\boldsymbol{L}_{k,i}}(c(v_k)=i), \quad i \in L(v), 1 \le k \le m \\ x_i^* &= \Phi(\mathbb{G},\boldsymbol{L},v,i,d), \quad i \in L(v), \\ x_{i,k}^* &= \Phi(\mathbb{G}_v,\boldsymbol{L}_{k,i},v_k,i,d-1), \quad i \in L(v)\cap L_{i,k}(v_k), 1 \le k \le m \end{aligned}$$

Proposition 4. *There exists a constant $\epsilon > 0$ which depends on α only such that*

$$\frac{1}{m}\max_{i\in L(v)}\big|\log(x_i)-\log(x_i^*)\big| \le (1-\epsilon)\max_{j\in L(v),k:m_k>0}\frac{1}{m_k}\big|\log(x_{j,k})-\log(x_{j,k}^*)\big| \tag{13}$$

First we show how this result implies Theorem 2:

Proof of Theorem 2. Applying this proposition d times and using the fact that we are summing over $k : m_k > 0$, we obtain

$$\frac{1}{m}\max_{i\in L(v)}\big[\log(x_i)-\log(x_i^*)\big] \le M(1-\epsilon)^d,$$

where

$$M = \max_{l,s}\Big|\log\mathbb{P}_{\mathbb{G}_s,\boldsymbol{L}_s}(c(v)=l) - \log\Phi(\mathbb{G}_s,\boldsymbol{L}_s,v,l,0)\Big|$$

and the maximum is over all graph/list pairs $\mathbb{G}_s, \boldsymbol{L}_s$ appearing during the computation of Φ and over all colors l. Recall that if l does not belong to the list associated with node v and list vector $\boldsymbol{L}_s$, then $\mathbb{P}_{\mathbb{G}_s,\boldsymbol{L}_s}(c(v)=l) = \Phi(\mathbb{G}_s, \boldsymbol{L}_s, v, l, 0) = 0$ (the first is equal to zero by definition, the second by the way we set the values of Φ). Otherwise we have from Lemma 2 and part (10) of Lemma 4 that absolute value of the difference is at most

$$\log q + \Delta \log(\beta/(\beta-1)).$$

Since $m \le \Delta \le n$, β is a constant which only depends on α, and $q \le n$, then we obtain $M = O(n)$. $\square$

Thus we focus on establishing Proposition 4.

Proof of Proposition 4. Observe that for every $i \in L(v) \setminus L_{k,j}(v_k)$ we have $x_{i,k} = x^*_{i,k} = 0$. This is because the probability of node v_k obtaining color i is zero when this color is not in its list. Similarly, the corresponding value of Φ is zero, since we set it to zero for all colors not in the list. For every $i \in L(v)$ introduce

$$A_i \triangleq \prod_{1 \le k \le m} (1 - x_{i,k}) \tag{14}$$

and

$$A \triangleq \sum_{j \in L(v)} A_j \tag{15}$$

Introduce A^*_i, A^* similarly. Applying Proposition 2 we obtain

$$x_i = \frac{A_i}{A}, \tag{16}$$

$$x^*_i = \min\Big[\frac{1}{2(1+\epsilon_0)m}, \frac{1}{\beta}, \frac{A^*_i}{A^*}\Big]. \tag{17}$$

Let

$$\tilde{x}^*_i = \frac{A^*_i}{A^*}.$$

We claim that in order to establish (13) it suffices to establish the bound

$$\frac{1}{m}|\log x_i - \log \tilde{x}^*_i| \le (1-\epsilon) \max_{j \in L(v), k: m_k > 0} \frac{1}{m_k}\big|\log(x_{j,k}) - \log(x^*_{j,k})\big|$$

Indeed, if $\tilde{x}^*_i \ne x^*_i$, then $x^*_i = \min[\frac{1}{2(1+\epsilon_0)m}, \frac{1}{\beta}]$. On the other hand, by Lemmas 2,3 we have $x_i \le \min[\frac{1}{2(1+\epsilon_0)m}, \frac{1}{\beta}]$, implying $x_i \le x^*_i \le \tilde{x}^*_i$, and the bound for $\tilde{x}^*_i$ implies the bound (13).

We have

$$\max_{i \in L(v)} \big|\log(x_i) - \log(x^*_i)\big| = \max_{i \in L(v)} \Big|\log A_i - \log A^*_i - \log A + \log A^*\Big|. \tag{18}$$

We introduce auxiliary variables $y_i = \log(x_i), y_{i,k} = \log(x_{i,k})$. Similarly, let $y^*_i = \log(\tilde{x}^*_i), y^*_{i,k} = \log(x^*_{i,k})$. Define $\boldsymbol{y} = (y_{i,k}), \boldsymbol{y}^* = (y^*_{i,k})$. Observe that if $m_k = 0$ then for every color i $x_{i,k} = x^*_{i,k}$. This follows since both values are $1/|L_{i,k}|$ when $i \in L_{i,k}$ and zero otherwise. This implies $y_{i,k} = y^*_{i,k}$. Then we rewrite (18) as

$$\max_{i \in L(v)} |y_i - y_i^*| = \max_{i \in L(v)} \Big| \sum_{k:m_k>0} \log(1-\exp(y_{i,k})) - \sum_{k:m_k>0} \log(1-\exp(y^*_{i,k})) \\ - \log\Big(\sum_{j \in L(v)} \prod_{1 \le k \le m} (1-\exp(y_{j,k}))\Big) + \log\Big(\sum_{j \in L(v)} \prod_{1 \le k \le m} (1-\exp(y^*_{j,k}))\Big)\Big|, \tag{19}$$

where the sums $\sum_{1 \le k \le m}$ were replaced by $\sum_{k:m_k>0}$ due to our observation $y_{i,k} = y^*_{i,k}$ when $m_k = 0$.

For every i denote the expression inside the absolute value in the right-hand side of equation (19) by $\mathcal{G}_i(\boldsymbol{y})$. That is we treat $\boldsymbol{y}^*$ as constant and $\boldsymbol{y}$ as a variable. It suffices to prove that for each i

$$\mathcal{G}_i(\boldsymbol{y}) \le (1-\epsilon) \max_{j \in L(v), k:m_k>0} \frac{1}{m_k} \big| \log(x_{j,k}) - \log(x^*_{j,k})\big| \tag{20}$$

Observe that $\mathcal{G}_i(\boldsymbol{y}^*) = 0$. Let $g_i(t) = \mathcal{G}_i(\boldsymbol{y}^* + t(\boldsymbol{y} - \boldsymbol{y}^*)), t \in [0,1]$. Then g_i is a differentiable function interpolating between 0 and $\mathcal{G}_i(\boldsymbol{y})$. In particular, $g_i(1) = \mathcal{G}_i(\boldsymbol{y})$. Applying the Mean Value Theorem we obtain

$$\begin{aligned} |g_i(1) - g_i(0)| = |g_i(1)| &\le \sup_{0 \le t \le 1} |\dot{g}_i(t)| \\ &= \sup_{0 \le t \le 1} \Big| \nabla \mathcal{G}_i(\boldsymbol{y}^* + t(\boldsymbol{y} - \boldsymbol{y}^*))^T (\boldsymbol{y} - \boldsymbol{y}^*)\Big| \end{aligned}$$

where the supremum is over values of t. We use a short-hand notation

$$\Pi_j = \prod_{1 \le k \le m} (1 - \exp(y_{j,k} + t(y_{j,k} - y^*_{j,k})))$$

For each t we have

$$\nabla \mathcal{G}_i(\boldsymbol{y}^* + t(\boldsymbol{y} - \boldsymbol{y}^*))(\boldsymbol{y} - \boldsymbol{y}^*) = \sum_{k:m_k>0} \frac{-\exp(y_{i,k} + t(y_{i,k} - y^*_{i,k}))}{1 - \exp(y_{i,k} + t(y_{i,k} - y^*_{i,k}))} (y_{i,k} - y^*_{i,k}) \\ + \frac{\sum_{j \in L(v)} \sum_{1 \le k \le m} \frac{\exp(y_{j,k} + t(y_{j,k} - y^*_{j,k}))}{1 - \exp(y_{j,k} + t(y_{j,k} - y^*_{j,k}))} (y_{j,k} - y^*_{j,k}))\Pi_j}{\sum_{j \in L(v)} \Pi_j}.$$

Again using the fact $y_{j,k} = y^*_{j,k}$ when $m_k = 0$, we can replace the sum $\sum_{1 \le k \le m}$ by $\sum_{k:m_k>0}$ in the expression above. For each j we have from convexity of exp

$$\begin{aligned} \exp(y_{j,k} + t(y_{j,k} - y^*_{j,k})) &\le (1-t)\exp(y^*_{j,k}) + t\exp(y_{j,k}) \\ &= (1-t)x_{j,k} + t x^*_{j,k} \\ &\le \frac{1}{2(1+\epsilon_0)m_k}. \end{aligned}$$

where the last inequality follows from Lemma 3 and part (8) of Lemma 4. This bound is useful for terms with $m_k > 0$ (for this reason we only kept these terms in the sum $\sum_{k:m_k>0}$). Similarly using

Lemma 2 and again part (8) of Lemma 4 we obtain

$$
\begin{aligned}
\frac{1}{1-\exp(y_{j,k}+t(y_{j,k}-y^*_{j,k}))} &\le \frac{1}{1-(1-t)\exp(y^*_{j,k})-t\exp(y_{j,k})} \\
&= \frac{1}{1-(1-t)x_{j,k}-tx^*_{j,k}} \\
&\le \frac{1}{1-\frac{1}{\beta}}.
\end{aligned}
$$

We obtain

$$
\begin{aligned}
\sup_{0\le t\le 1}\left|\nabla\mathcal{G}_i(\boldsymbol{y}^*+t(\boldsymbol{y}-\boldsymbol{y}^*))(\boldsymbol{y}-\boldsymbol{y}^*)\right| &\le \sum_{k:m_k>0}\frac{1}{(1-\frac{1}{\beta})2(1+\epsilon_0)m_k}|y_{i,k}-y^*_{i,k}| \\
&+ \frac{\sum_{j\in L(v)}\sum_{k:m_k>0}(1-\frac{1}{\beta})^{-1}2^{-1}(1+\epsilon_0)^{-1}m_k^{-1}|y_{j,k}-y^*_{j,k}|\Pi_j}{\sum_{j\in L(v)}\Pi_j} \\
&\le \frac{m}{(1-\frac{1}{\beta})2(1+\epsilon_0)}\max_{j\in L(v),k:m_k>0}\frac{|y_{j,k}-y^*_{j,k}|}{m_k} \\
&+ \frac{m}{(1-\frac{1}{\beta})2(1+\epsilon_0)}\max_{j\in L(v),k:m_k>0}\frac{|y_{j,k}-y^*_{j,k}|}{m_k} \\
&= \frac{m}{(1-\frac{1}{\beta})(1+\epsilon_0)}\max_{j\in L(v),k:m_k>0}\frac{|y_{j,k}-y^*_{j,k}|}{m_k}.
\end{aligned}
$$

Combining with (19) we conclude

$$
\max_{i\in L(v)}\frac{|y_i-y^*_i|}{m} \le \frac{1}{(1-\frac{1}{\beta})(1+\epsilon_0)}\max_{j\in L(v),k:m_k>0}\frac{|y_{j,k}-y^*_{j,k}|}{m_k}.
$$

We now select a sufficiently large constant $\beta=\beta(\epsilon_0)$ such that

$$
1-\epsilon \triangleq \frac{1}{(1-\frac{1}{\beta})(1+\epsilon_0)} < 1.
$$

This completes the proof of Proposition 4. □

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

As we have mentioned the (spatial) correlation decay is known to hold for coloring problem in a stronger regime $\alpha>\alpha^*\approx 1.763\ldots$, then the regime $\alpha>\alpha^{**}$ considered in this paper [GMP05]. This decay of correlation is established in a conventional sense: for every node v the marginal probability $\mathbb{P}(c(v)=i)$ is asymptotically independent from changing a color on a boundary of the depth-d neighborhood $B(v,d)$ of v in the underlying graph. In fact it is established that the decay of correlation is exponential in d. It is natural to try to use this result directly as a method for computing approximately the marginals $\mathbb{P}(c(v)=i)$, for example by computing the marginal $\mathbb{P}_{B(v,d)}(c(v)=i)$ corresponding to the neighborhood $B(v,d)$, say using brute force computation. Unfortunately, this conventional correlation decay result is not useful because of the computation growth. In order to obtain ϵ-approximation of the partition function, we need order $O(\epsilon/n)$ approximation of the marginals, 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 $B(v,d)$ is then $O(\Delta^{\log n}) = n^{O(1)}$ - 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 the conventional graph theoretic sense helps.

7 Conclusions

We have established existence of a deterministic algorithm for counting the number of list colorings of a graph. While the analysis in this paper was restricted to the problem of list colorings, the algorithm generalizes to arbitrary constraint satisfaction (integer programming) type problems. Along with [BG06] and [Wei05] this work is an important step in the direction of developing a new powerful method for solving counting problems using insights from statistical physics. This method provides an alternative to the existing MCMC sampling based method.

The principle insight from this work is the advantage of establishing the correlation decay property on the *computation tree* as opposed to the original graph theoretic structure, as has been done primarily prior to this work. While we have established such correlation decay only in the regime $\alpha > 2.8432...$, we conjecture that it holds for much lower values of α. In fact, just as it is conjectured that the Markov chain is rapidly mixing in the regime $q \geq \Delta + 2$, we conjecture that the correlation decay on the computation tree holds in this regime as well. We expect that a similar correlation decay property holds for computation trees corresponding to other counting problems. Natural place to start is counting matchings, permanent of a matrix, or volume of a polytope. We note that while the volume of a convex body cannot be approximated using deterministic algorithms, [Ele86],[BF87], the presence of a linear programming structure might help, perhaps in some interesting sub-class of polytopes. It is also of interest to see the computational feasibility of the proposed algorithm. Finally, it would be of interest to see whether correlation decay on the computation tree has any implications for the mixing rate of the underlying Markov chain. It is known that the conventional decay of correlation implies rapid mixing on graphs with sub-exponential growth. It would be interesting to see whether any extra milage can be gained from looking at the correlation decay property of the computation tree.

Acknowledgements

The authors are very thankful to Devavrat Shah for contributing many important comments for this work.

References

[BF87] I. Bárány and Z. Füredy, *Computing the volume is difficult*, Discrete Comput. Geom. **2** (1987), no. 4, 319–326.

[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.

[BW02] G. Brightwell and P. Winkler, *Random colorings of a Cayley tree*, in Contemporary Combinatorics, B. Bollobas, ed., Bolyai Society Mathematical Studies, 2002, pp. 247–276.

[BW04] G.R. Brightwell and P. Winkler, *Graph homomorphisms and long range action*, in Graphs, morphisms and statistical physics (Nesetril and Winkler eds.), DIMACS series in discrete mathematics and computer science, 2004, pp. 29–47.

[DFHV04] M. Dyer, A. Frieze, T. Hayes, and E. Vigoda, *Randomly coloring constant degree graphs*, in Proceedings of 45th IEEE Symposium on Foundations of Computer Science, 2004.

[Ele86] G. Elekes, *A geometric inequality and the complexity of computing volume*, Discrete Comput. Geom. **1** (1986), no. 4, 289–292.

[FV06] A. Frieze and E. Vigoda, *Survey of Markov chains for randomly sampling colorings*, *To appear in* Festschrift for Dominic Welsh (2006).

[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.

[Jer95] M. Jerrum, *A very simple algorithm for estimating the number of k-colourings of a low-degree graph*, Random Structures and Algorithms **7** (1995), no. 2, 157–165.

[Jon02] J. Jonasson, *Uniqueness of uniform random colorings of regular trees*, Statistics and Probability Letters **57** (2002), 243–248.

[JS06] K. Jung and D. Shah, *On correctness of belief propagation algorithm*, Preprint (2006).

[Sha05] D. Shah, *Max weight independent set and matching via max-product*, Preprint (2005).

[SS97] J. Salas and A. D. Sokal, *Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem*, J. Statist. Phys. **86** (1997), no. 3-4, 551–579.

[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.*

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

[Wei05] D. Weitz, Counting down the tree*, Submitted. (2005).*

[YFW00] J. Yedidia, W. Freeman, and Y. Weiss, Understanding Belief Propagation and its generalizations*, Mitsubishi Elect. Res. Lab. (2000), no. TR-2001-22.*