FPTAS for Counting Weighted Edge Covers

Jingcheng Liu[1], Pinyan Lu[2], and Chihao Zhang⋆[1]

[1] Shanghai Jiao Tong University, {liuexp, chihao.zhang}@gmail.com
[2] Microsoft Research. pinyanl@microsoft.com

Abstract. An edge cover of a graph is a set of edges in which each vertex has at least one of its incident edges. The problem of counting the number of edge covers is #**P**-complete and was shown to admit a fully polynomial-time approximation scheme (FPTAS) recently [10]. Counting weighted edge covers is the problem of computing the sum of the weights for all the edge covers, where the weight of each edge cover is defined to be the product of the edge weights of all the edges in the cover. The FPTAS in [10] cannot apply to general weighted counting for edge covers, which was stated as an open question there. Such weighted counting is generally interesting as for instance the weighted counting independent sets (vertex covers) problem has been exhaustively studied in both statistical physics and computer science. Weighted counting for edge cover is especially interesting as it is closely related to counting perfect matchings, which is a long-standing open question. In this paper, we obtain an FPTAS for counting general weighted edge covers, and thus solve an open question in [10]. Our algorithm also goes beyond that to certain generalization of edge cover.

1 Introduction

An edge cover for an undirected graph $G(V, E)$ is a set of edges $C \subseteq E$ such that for every $v \in V$, it holds that $N(v) \cap C \neq \varnothing$ where $N(v)$ is the set of edges incident to v. The problem of counting edge covers in an undirected graphs was known to be #**P**-hard and was recently shown to admit a fully polynomial-time approximation scheme (FPTAS)[10].

A natural generalization of the edge cover problem is to consider edge weights. That is, we assign a positive real number λ_e for every edge $e \in E$ and an edge cover C is of weight $w_C \triangleq \prod_{e \in C} \lambda_e$. Denote $EC(G)$ the set of edge covers of G, the problem of counting weighted edge covers is to compute

$$\sum_{C \in EC(G)} w_C = \sum_{C \in EC(G)} \prod_{e \in C} \lambda_e.$$

Such sum of product expression is usually called partition function in statistical physics, or graph polynomial in combinatorics, which are of great interests. For example, if we replace the edge cover constraint with matching constraint, then

⋆ The author is supported by NSF of China (61033002, ANR 61261130589)

we get the well-known matching polynomial. If we replace the constraint with vertex cover (or complementary independent set) constraint, and edge weights with vertex weights, we get the problem of counting weighted independent sets, which is also known as hard-core model in statistical physics. This problem is extensively studied and a complete understanding was not available until very recently [5,20,16,6,17]. There is a phase transition in terms of weights for the computational complexity of the problem. In [10], Lin et al. asked whether the problem of counting weighted edge covers also exhibits a phase transition in terms of edge weights. In particular, the method in [10] can be extended to that all edges are of uniform weight λ with $\lambda \geq \frac{4}{9}$, but not further. We answer the question by designing an FPTAS for counting weighted edge covers with arbitrary edge weights, even if they are not uniform, provided that they are constants.

In weighted edge covers, $\lambda_e > 1$ indicates that the edge e is preferred to be chosen and it is preferred not if $\lambda_e < 1$. If all the edge weights are the same and smaller than 1, then an edge cover with smaller cardinality contributes more in the sum. As the uniform edge weight approach zero (exponentially small in terms of the graph size), the weights from the minimum edge covers will dominate all the other terms. Provided that the graph has a perfect matching, the set of minimum edge covers is exactly the same as the set of perfect matchings. Therefore when the edge weights are exponentially small in terms of the graph size, the problem of counting weighted edge covers is essentially counting minimal edge covers, which is even stronger than counting perfect matchings, for which no polynomial-time approximation algorithm in general was known. It is widely open whether one can design an FPTAS for counting perfect matchings or not. But unfortunately, our FPTAS only works for constant edge weights, which is not exponentially small in terms of the input size.

It is worth noting that there is a similar situation for counting weighted matchings (not necessarily perfect). There is a fully polynomial-time randomized approximation scheme (FPRAS) for constant edge weights based on Markov chain Monte-Carlo method [7]. If one allows the weights go to infinity (exponentially large in terms of the graph size), counting matching is essentially the same as counting perfect matching provided that the graph has one since the contribution from those perfect matchings will dominate the others. But the known algorithm does not work for exponentially large weights either. In some sense, the constraint of perfect matching is upper and lower bounded by the constraints of edge cover and partial matching respectively. For both, we have approximate algorithms, while the perfect matching problem is widely open. With our new FPTAS for counting weighted edge covers, it is interesting to see that if we can play with these upper and lower bounds simultaneously to get an algorithm for counting perfect matchings. We remark that our algorithm for weighted edge cover is deterministic while the general algorithm for counting matchings is randomized and deterministic FPTAS is only known for graphs with bounded degree even in the unweighted setting [1].

We then consider another generalization of edge cover: We allow vertices stay uncovered in a "cover" and each of these (uncovered) vertices v contributes a weight (or penalty) $\mu_v \in [0,1]$ to the weight of the cover. Formally, we have

$$Z(G) \triangleq \sum_{\sigma \in \{0,1\}^E} \prod_{e \in E} \lambda_e^{\sigma(e)} \prod_{v \in V} \mu_v^{\delta(\sigma,v)},$$

where $\delta(\sigma, v)$ is defined to be

$$\delta(\sigma, v) = \begin{cases} 0, & \text{if } \sigma(e) = 1 \text{ for some } e \text{ incident to } v \\ 1, & \text{otherwise.} \end{cases}$$

This is similar to allowing omissible vertices for perfect matching [18]. The original edge cover is equivalent to the case that $\mu_v = 0$ for every $v \in V$. Our FPTAS can also be generalized to this generalization of weighted edge cover. Indeed, we shall state our theorem, algorithm and proof for this generalization directly and the ordinary weighted edge cover follows as a special case. Formally, we have the following main result.

Theorem 1. *For any constant $\lambda > 0$, there is an FPTAS to approximate $Z(G)$ for graphs $G(V, E)$ with edge weights $\lambda_e \geq \lambda$ for every edge $e \in E$ and vertex weights $\mu_v \in [0, 1]$ for every $v \in V$.*

1.1 Related Works

Counting edge covers was previously studied in [2] where a Markov chain Monte-Carlo based algorithm was given for 3-regular graphs. Later in [10], an FPTAS for general graphs was proposed.

Our technique for designing FPTAS is the correlation decay method. The technique was proved to be very powerful in obtaining FPTAS for counting problems, some notable examples include [1,20,14,9,11,15,12]. An crucial ingredient of our analysis is the use of *potential function* (or called *message* in some literature) to amortize error propagated [13,8,14,9,15,11].

The problem of counting (perfect) matching, edge cover and our generalization with vertex weights can be uniformly treated in the framework of Holant problems [19,3,4].

2 Preliminaries

2.1 Dangling edge

Following [10], we introduce dangling edges into our graph to simplify the description of our algorithm and proofs.

Definition 2. *A **dangling edge** $e = (u, \_)$ in a graph $G(V, E)$ is such an edge with exactly one end point $u \in V$.*

*A **free edge** $e = (\_, \_)$ is an edge with no end points.*

(a) G (b) G'

Fig. 1: Breaking up a normal edge into two dangling edges.

A graph with two dangling edges e_1, e_2 is depicted in Figure 1b.

It is natural to generalize $Z(G)$ to graphs with dangling edges. For a graph $G = (V, E)$, an edge $e = (u, v) \in E$ and a vertex $u \in V$, define

$$\begin{aligned} G - e \triangleq& (V, E - e) \\ e - u \triangleq& (\_, v) \quad \text{(note that here } v \text{ could be } \_) \\ G - u \triangleq& (V - u, \\ & \{e \mid e \in E, e \text{ is not incident with } u\} \\ & \cup \{e - u \mid e \in E, e \text{ is incident with } u\}) \end{aligned}$$

The definition of $G - u$ indicates that all edges incident to u in G become dangling in $G - u$.

2.2 Approximate counting from estimation of marginal probabilities

The definition of the partition function naturally induces a Gibbs measure on all configurations over E. From this joint distribution, we can also define marginal probability for a (dangling) edge e. For $c \in \{0, 1\}$, we define

$$Z_{e=c}(G) \triangleq \sum_{\substack{\sigma \in \{0,1\}^E \\ \sigma(e) = c}} \prod_{v \in V} \mu_v^{\delta(\sigma, v)} \prod_{e \in E} \lambda_e^{\sigma(e)}$$

The marginal probability that e is chosen ($\sigma(e) = 1$) or not ($\sigma(e) = 0$) can be expressed as

$$\mathbb{P}_G\left(e = 0\right) \triangleq \frac{Z_{e=0}(G)}{Z(G)}, \quad \mathbb{P}_G\left(e = 1\right) \triangleq \frac{Z_{e=1}(G)}{Z(G)}.$$

It is a standard routine to approximate the partition function $Z(G)$ if the marginal probability can be well-estimated.

Proposition 3. *There is an FPTAS for approximating the partition function of weighted edge cover provided an oracle $\mathcal{O}$ to estimate $\mathbb{P}_G\left(e = 1\right)$ where $G(V, E)$ is an arbitrary graph with (dangling) edge e. $\mathcal{O}$ takes input $G, e, \varepsilon > 0$ and is required to satisfy*

1. $\mathcal{O}$ *outputs an estimate* $\hat{p}$ *within time polynomial in* $|G|$ *and* $1/\varepsilon$*;*
2. $\exp(-\varepsilon) \cdot \hat{p} \leq \mathbb{P}_G (e = 1) \leq \exp(\varepsilon) \cdot \hat{p}$.

Proof. Let $G(V, E)$ be a graph and we now give an algorithm to estimate $Z(G)$ with the help of the oracle. Let $\boldsymbol{\sigma} \in \{0, 1\}^E$ be the configuration that $\boldsymbol{\sigma}(e) = 1$ for every $e \in E$. Then

$$\mathbb{P}_G(\boldsymbol{\sigma}) = \frac{\omega_{\boldsymbol{\sigma}}}{Z(G)}$$

where $w_{\boldsymbol{\sigma}}$ is the weight of configuration $\boldsymbol{\sigma}$ and it is easily computable. Thus in order to compute $Z(G)$, it is sufficient to estimate $\mathbb{P}_G(\boldsymbol{\sigma})$.

We fix an arbitrary order of edges in E, i.e., $E = \{e_1, \dots, e_m\}$ in which $e_i = (u_i, v_i)$ for every $1 \leq i \leq m$. Then

$$\mathbb{P}_G(\boldsymbol{\sigma}) = \mathbb{P}_G\left(\bigwedge_{i=1}^{m} \sigma(e_i) = 1\right) = \prod_{i=1}^{m} \mathbb{P}_G\left(e_i = 1 \middle| \bigwedge_{j=1}^{i-1} e_j = 1\right).$$

Define $G_1 \triangleq G$, $G_i \triangleq G_{i-1} - e_{i-1} - u_{i-1} - v_{i-1}$, for $2 \leq i \leq m$. We have $\mathbb{P}_G\left(e_i = 1 \middle| \bigwedge_{j=1}^{i-1} e_j = 1\right) = \mathbb{P}_{G_i}(e_i = 1)$. For every $1 \leq i \leq m$, we call the oracle with input $\left(G_i, e_i, \frac{\varepsilon}{2|E|}\right)$. Let $\hat{p}_i$ be the result of our i-th call, $\hat{p} = \prod_i^m \hat{p}_i$ and $\hat{Z} = \frac{\omega_{\boldsymbol{\sigma}}}{\hat{p}}$, then it holds that

$$\exp(-\varepsilon) \cdot Z(G) \leq \hat{Z} \leq \exp(\varepsilon) \cdot Z(G).$$

□

3 Approximation for Marginal Probabilities

In this section, we prove Theorem 1. By Proposition 3, we only need to estimate the marginal probabilities as following:

Lemma 4. *Let* $G(V, E)$ *be an instance of weighted edge cover with an edge* e*, and vertex weight* $\mu_v \leq 1$*, there is an algorithm* $\mathcal{A}$ *that efficiently approximates* $\mathbb{P}_G(e = 1)$*. More precisely,* $\mathcal{A}$ *takes as input* $G, e, \varepsilon > 0$ *and the following holds:*

1. $\mathcal{A}$ *outputs an estimate* $\hat{p}$ *within time polynomial in* $|G|$ *and* $1/\varepsilon$*;*
2. $\exp(-\varepsilon) \cdot \hat{p} \leq \mathbb{P}_G (e = 1) \leq \exp(\varepsilon) \cdot \hat{p}$.

The lemma together with Proposition 3 implies Theorem 1.

3.1 Computational tree recursion and the algorithm

We use computational tree recursion to compute $\mathbb{P}_G(e = 0)$, a good estimate of which is also a good estimate of $\mathbb{P}_G(e = 1)$. We express $\mathbb{P}_G(e = 0)$ as a function of marginal probabilities on smaller instances.

Free edge. If e is a free edge, then $\mathbb{P}_G\left(e=0\right)=\frac{1}{1+\lambda_e}$.

Normal edge. Assume $e=(u,v)$, we define a recursion to compute $R_G(e)\triangleq\frac{\mathbb{P}_G(e=1)}{\mathbb{P}_G(e=0)}$. Then $\mathbb{P}_G\left(e=0\right)=\frac{1}{1+R_G(e)}$.

To this end, we replace $e=(u,v)$ with dangling edges $e_1=(u,\_)$ and $e_2=(v,\_)$. Denote this new graph by $G'(V',E')$, as depicted in Figure 1a and 1b.

We further let $G_1\triangleq G'-e_2$, $G_2\triangleq G'-e_1-u$. It holds that

$$\begin{aligned}R_G(e)&=\frac{\mathbb{P}_{G'}\left(e_1=1,e_2=1\right)}{\mathbb{P}_{G'}\left(e_1=0,e_2=0\right)}\\&=\frac{\mathbb{P}_{G'}\left(e_1=1,e_2=0\right)}{\mathbb{P}_{G'}\left(e_1=0,e_2=0\right)}\cdot\frac{\mathbb{P}_{G'}\left(e_1=1,e_2=1\right)}{\mathbb{P}_{G'}\left(e_1=1,e_2=0\right)}\\&=\frac{\mathbb{P}_{G_1}\left(e_1=1\right)}{\mathbb{P}_{G_1}\left(e_1=0\right)}\cdot\frac{\mathbb{P}_{G_2}\left(e_2=1\right)}{\mathbb{P}_{G_2}\left(e_2=0\right)}\\&=R_{G_1}(e_1)\cdot R_{G_2}(e_2).\end{aligned}$$

This directly gives the recursion for $\mathbb{P}_G\left(e=0\right)$:

$$\mathbb{P}_G\left(e=0\right)=\frac{\mathbb{P}_{G_1}\left(e_1=0\right)\mathbb{P}_{G_2}\left(e_2=0\right)}{1-\mathbb{P}_{G_1}\left(e_1=0\right)-\mathbb{P}_{G_2}\left(e_2=0\right)+2\mathbb{P}_{G_1}\left(e_1=0\right)\mathbb{P}_{G_2}\left(e_2=0\right)}.$$

We remark that in the RHS of the recursion, both e_1 and e_2 are dangling edges in G_1 and G_2 respectively.

Dangling edge. Let $e=(u,\_)$ be the dangling edge. Denote $\hat{E}=\{e_i\mid 1\leq i\leq d\}$ the set of other edges incident to u. Let $G'\triangleq G-e-u$ as illustrated in Figure 2a and 2b.

(a) G (b) G'

Fig. 2: Dangling edges examples.

Define a family of graphs $\{G_i\}_{1\leq i\leq d}$ obtained by removing edges in $\hat{E}$ consecutively: $G_1\triangleq G'$, $G_i\triangleq G_{i-1}-e_{i-1}$, for $2\leq i\leq d$.

Let $\boldsymbol{\alpha}\in\{0,1\}^d$ be a configuration over $\hat{E}$. We use $Z_{\boldsymbol{\alpha}}(G)$ to denote the sum of weights over configurations of G whose restriction on $\hat{E}$ is consistent with $\boldsymbol{\alpha}$. Formally we let

$$Z_{\boldsymbol{\alpha}}(G)\triangleq\sum_{\substack{\sigma\in\{0,1\}^E\\ \sigma|_{\hat{E}}=\boldsymbol{\alpha}}}\prod_{v\in V}\mu_v^{\delta(\sigma,v)}\prod_{e\in E}\lambda_e^{\sigma(e)}.$$

Then by the definition of the marginal probability, we have

$$\begin{aligned}\mathbb{P}_G\left(e=0\right) &= \frac{Z_{e=0}(G)}{Z_{e=0}(G)+Z_{e=1}(G)} \\ &= \frac{\mu_u Z_{\mathbf{0}}(G') + \sum_{\boldsymbol{\alpha}\in\{0,1\}^d,\boldsymbol{\alpha}\neq\mathbf{0}} Z_{\boldsymbol{\alpha}}(G')}{(\mu_u+\lambda_e)Z_{\mathbf{0}}(G') + (1+\lambda_e)\sum_{\boldsymbol{\alpha}\in\{0,1\}^d,\boldsymbol{\alpha}\neq\mathbf{0}} Z_{\boldsymbol{\alpha}}(G')} \\ &= \frac{Z(G')-(1-\mu_u)Z_{\mathbf{0}}(G')}{(1+\lambda_e)Z(G')-(1-\mu_u)Z_{\mathbf{0}}(G')} \\ &= \frac{1-(1-\mu_u)\frac{Z_{\mathbf{0}}(G')}{Z(G')}}{1+\lambda_e-(1-\mu_u)\frac{Z_{\mathbf{0}}(G')}{Z(G')}}. \end{aligned} \tag{1}$$

The term $\frac{Z_{\mathbf{0}}(G')}{Z(G')}$ can be expressed as a product of probabilities:

$$\frac{Z_{\mathbf{0}}(G')}{Z(G')} = \mathbb{P}_{G'}\left(\hat{E}=\mathbf{0}\right) = \prod_{i=1}^{d}\mathbb{P}_{G'}\left(e_i=0 \,\middle|\, \bigwedge_{j=1}^{i-1} e_j=0\right) = \prod_{i=1}^{d}\mathbb{P}_{G_i}\left(e_i=0\right).$$

Plugging this into (1), we obtain

$$\mathbb{P}_G\left(e=0\right) = \frac{1-(1-\mu_u)\prod_{i=1}^{d}\mathbb{P}_{G_i}\left(e_i=0\right)}{1+\lambda_e-(1-\mu_u)\prod_{i=1}^{d}\mathbb{P}_{G_i}\left(e_i=0\right)}.$$

We remark the if e is the only incident edge to u (i.e. $d=0$), we have $\mathbb{P}_G\left(e=0\right) = \frac{\mu_u}{\lambda_e+\mu_u}$, which is consistent with the above recursion if we take the convention that an empty product is 1. We also note that every edge e_i in the RHS is a dangling or free edge of G_i.

The above recursion gives a computation tree to compute $\mathbb{P}_G\left(e=0\right)$. We truncate it get our Algorithm 1 to estimate $\mathbb{P}_G\left(e=0\right)$.

3.2 Analysis of correlation decay

We recall that $\lambda_e \geq \lambda$ for a constant $\lambda > 0$. We show that Algorithm 1 is a good estimator for $\mathbb{P}_G\left(e=0\right)$. Formally, denote $\mathbb{P}_G^{\ell}\left(e=0\right) \triangleq \texttt{compute}\left(\ell,\texttt{G},\texttt{e}\right)$, we prove

Lemma 5. *For every $\ell \geq 0$,*

$$\left|\mathbb{P}_G^{\ell}\left(e=0\right) - \mathbb{P}_G\left(e=0\right)\right| \leq \alpha\cdot(1+\lambda)^{-\ell/2}.$$

where $\alpha \triangleq \frac{1}{4}\ln\left(1+\frac{1}{\lambda}\right)\cdot\max\left\{1,\frac{2(1+\lambda)^3}{\lambda^4}\right\}$.

In order to establish this lemma, we first prove two auxiliary lemmas. Lemma 7 deals with the recursion for dangling edges and Lemma 8 provides a universal bound for marginal probabilities we estimate.

Algorithm 1: Estimating $\mathbb{P}_G(e=0)$

```
function compute(ℓ, G, e) :
input : Recursion depth ℓ; Graph G(V,E) with edge e
output: An estimate of P_G(e = 0)
begin
    if ℓ ≤ 0 then
        return 1/(1+λ_e);
    else if e is a free edge then
        return 1/(1+λ_e);
    else if e is a dangling edge then
        ℓ' ← ℓ − ⌈(d+1)/2⌉;
        return (1−(1−μ_u) ∏_{i=1}^{d} compute(ℓ',G_i,e_i)) / (1+λ_e−(1−μ_u) ∏_{i=1}^{d} compute(ℓ',G_i,e_i));
    else // e is a normal edge
        X ← compute(ℓ, G_1, e_1);
        Y ← compute(ℓ, G_2, e_2);
        return XY/(1−X−Y+2XY);
```

A powerful technique to prove the correlation decay property for a recursion system is to use potential function to amortize the error propagated.

Let $f : D^d \to \mathbb{R}$ be a d-ary function where $D \subseteq \mathbb{R}$ and $\phi : \mathbb{R} \to \mathbb{R}$ is an increasing differentiable continuous function. Denote $\Phi(x) \triangleq \phi'(x)$ and $f^\phi(\boldsymbol{x}) \triangleq \phi(f(\phi^{-1}(x_1), \ldots, \phi^{-1}(x_d)))$. The following proposition is a consequence of mean value theorem:

Proposition 6. *For every* $\boldsymbol{x} = (x_1, \ldots, x_d), \hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_d) \in D^d$*, it holds that*

1. $|f(\boldsymbol{x}) - f(\hat{\boldsymbol{x}})| = \frac{1}{|\Phi(\tilde{x})|} \cdot |\phi(f(\boldsymbol{x})) - \phi(f(\hat{\boldsymbol{x}}))|$ *for some* $\tilde{x} \in D$*;*
2. *Assume* $x_i = f(\boldsymbol{x_i})$ *and* $\hat{x}_i = f(\hat{\boldsymbol{x}}_{\boldsymbol{i}})$ *for all* $1 \le i \le d$*, then*

$$|\phi(f(\boldsymbol{x})) - \phi(f(\hat{\boldsymbol{x}}))| \le \left\|\nabla f^\phi(\tilde{\boldsymbol{x}})\right\|_1 \cdot \max_{1 \le i \le d} |\phi(f(\boldsymbol{x_i})) - \phi(f(\hat{\boldsymbol{x}}_{\boldsymbol{i}}))|$$

for some $\tilde{\boldsymbol{x}} \in D^d$*.*

The proof is standard, one can find it in, e.g. [15].

Lemma 7. *Let* $\boldsymbol{x} = (x_1, \ldots, x_d), \hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_d) \in \left(0, \frac{1}{1+\lambda}\right]^d$ *for some* $\lambda > 0$. *For every* $\hat{\lambda} > 0$ *and* $0 \le \hat{\mu} \le 1$, *define*

$$\begin{aligned} f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x}) &\triangleq \frac{1-(1-\hat{\mu})\prod_{i=1}^d x_i}{1+\hat{\lambda}-(1-\hat{\mu})\prod_{i=1}^d x_i}; \\ \Phi(x) &\triangleq \frac{1}{x(1-x)}; \\ \phi(x) &\triangleq \int \Phi(x)\,\mathrm{d}x = \ln\left(\frac{x}{1-x}\right); \\ f^{\phi}_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x}) &\triangleq \phi(f_{\hat{\lambda},\hat{\mu}}(\phi^{-1}(x_1), \ldots, \phi^{-1}(x_d))). \end{aligned}$$

Assume $x_i = f_{\lambda_i,\mu_i}(\boldsymbol{z}_i)$, $\hat{x}_i = f_{\lambda_i,\mu_i}(\hat{\boldsymbol{z}}_i)$ *for all* $1 \le i \le d$. *Then*

1. $f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x}) \le \frac{1}{1+\hat{\lambda}}$.
2. $\left|\phi(f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x})) - \phi(f_{\hat{\lambda},\hat{\mu}}(\hat{\boldsymbol{x}}))\right| \le (1+\lambda)^{-\frac{d+1}{2}} \max_{1\le i\le d} |\phi(f_{\lambda_i,\mu_i}(\boldsymbol{z}_i)) - \phi(f_{\lambda_i,\mu_i}(\hat{\boldsymbol{z}}_i))|$.

Proof. 1. $f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x})$ is monotonically decreasing with respect to each x_i, thus $f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x}) \le f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{0}) \le \frac{1}{1+\hat{\lambda}}$.

2. For every $\boldsymbol{x} \in \left(0, \frac{1}{1+\lambda}\right]^d$, it holds that

$$\begin{aligned} \left\|\nabla f^{\phi}_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x})\right\|_1 &= \Phi(f_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x})) \cdot \sum_{i=1}^d \left| \frac{\frac{\partial f_{\hat{\lambda},\hat{\mu}}(x_1,\ldots,x_d)}{\partial x_i}}{\Phi(x_i)} \right| \\ &= \frac{\left(1+\hat{\lambda}-(1-\hat{\mu})\prod_{i=1}^d x_i\right)^2}{\hat{\lambda}\left(1-(1-\hat{\mu})\prod_{i=1}^d x_i\right)} \cdot \frac{\hat{\lambda}(1-\hat{\mu})\prod_{i=1}^d x_i}{\left(1+\hat{\lambda}-(1-\hat{\mu})\prod_{i=1}^d x_i\right)^2} \cdot \sum_{i=1}^d (1-x_i) \\ &= \frac{(1-\hat{\mu})\prod_{i=1}^d x_i}{1-(1-\hat{\mu})\prod_{i=1}^d x_i}\left(d - \sum_{i=1}^d x_i\right) \\ &\le \frac{\prod_{i=1}^d x_i}{1-\prod_{i=1}^d x_i}\left(d - \sum_{i=1}^d x_i\right). \end{aligned}$$

Let $y \triangleq \left(\prod_{i=1}^d x_i\right)^{1/d}$ and note that $y \le \frac{1}{1+\lambda}$, we have

$$\left\|\nabla f^{\phi}_{\hat{\lambda},\hat{\mu}}(\boldsymbol{x})\right\|_1 \le \frac{dy^d(1-y)}{1-y^d} = \frac{dy^d}{\sum_{i=0}^{d-1} y^i} \le \frac{d}{\sum_{i=1}^d (1+\lambda)^i} \le (1+\lambda)^{-\frac{d+1}{2}}.$$

Then the lemma follows from Proposition 6. □

Lemma 8. *For an arbitrary* $G(V,E)$ *with dangling edge* $e = (u, \_)$ *and* $\ell \ge 0$. *It holds that*

$$\mathbb{P}^{\ell}_G(e=0), \mathbb{P}_G(e=0) \le \frac{1}{1+\lambda_e} \le \frac{1}{1+\lambda}$$

Proof. If e is a free edge, then the lemma naturally holds. Otherwise, the bound follows from the first part of Lemma 7. □

We are now ready to prove Lemma 5.

Proof (of Lemma 5).

- If e is a free edge, then $\left|\mathbb{P}_G^{\ell}(e=0) - \mathbb{P}_G(e=0)\right| = 0$.
- If $e = (u, \_)$ is a dangling edge, recall that $\phi(x) = \ln\left(\frac{x}{1-x}\right)$, we first prove that for every ℓ (may be negative), it holds that

$$\left|\phi\left(\mathbb{P}_G^{\ell}(e=0)\right) - \phi\left(\mathbb{P}_G(e=0)\right)\right| \le \ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-L/2} \tag{2}$$

where $L \triangleq \max\{\ell, 0\}$.
Denote $\hat{E} \triangleq \{e_1, \ldots, e_d\}$ the set of edges incident to e. If $d = 0$, we have $\mathbb{P}_G^{\ell}(e=0) = \mathbb{P}_G(e=0)$. So we assume $d \ge 1$ and apply induction on L. The base case is that $L = 0$, which means $\ell \le 0$.
Then

$$\begin{aligned}\left|\phi\left(\mathbb{P}_G^{\ell}(e=0)\right) - \phi\left(\mathbb{P}_G(e=0)\right)\right| &= \left|\phi\left(\frac{1}{1+\lambda_e}\right) - \phi\left(\frac{1-(1-\mu_u)\prod_{i=1}^d x_i}{1+\lambda_e-(1-\mu_u)\prod_{i=1}^d x_i}\right)\right| \\ &= \ln\left(\frac{1}{1-(1-\mu_u)\prod_{i=1}^d x_i}\right)\end{aligned}$$

where $x_i \triangleq \mathbb{P}_{G_i}(e_i=0)$. It follows from Lemma 8 that for every $1 \le i \le d$, $x_i \le \frac{1}{1+\lambda}$, thus

$$\left|\phi\left(\mathbb{P}_G^{\ell}(e=0)\right) - \phi\left(\mathbb{P}_G(e=0)\right)\right| \le -\ln\left(1-\frac{1}{(1+\lambda)^d}\right) \le \ln\left(1+\frac{1}{\lambda}\right).$$

Now assume $L = \ell > 0$ and (2) holds for smaller L. Then the induction hypothesis implies that

$$\varepsilon \triangleq \max_{1\le i\le d}\left|\phi\left(\mathbb{P}_{G_i}^{\ell'}(e_i=0)\right) - \phi\left(\mathbb{P}_{G_i}(e_i=0)\right)\right| \le \ln\left(1+\frac{1}{\lambda}\right)(1+\lambda)^{-L'/2}$$

where $L' = \max\left\{0, \ell - \lceil\frac{d+1}{2}\rceil\right\}$.
Applying Proposition 6, Lemma 7 and Lemma 8, we obtain

$$\begin{aligned}\left|\phi\left(\mathbb{P}_G^{\ell}(e=0)\right) - \phi\left(\mathbb{P}_G(e=0)\right)\right| &\le (1+\lambda)^{-\frac{d+1}{2}}\varepsilon \\ &\le (1+\lambda)^{-\frac{d+1}{2}}\cdot\ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-L'/2} \\ &\le \ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-\left(L-\lceil\frac{d+1}{2}\rceil+d+1\right)/2} \\ &\le \ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-L/2}.\end{aligned}$$

Recall that $\Phi(x) = \frac{1}{x(1-x)} \geq 4$ for $x \in (0,1)$. For all $\ell \geq 0$, Proposition 6 and Lemma 8 together imply that

$$\begin{aligned}\left|\mathbb{P}_G^{\ell}\left(e=0\right) - \mathbb{P}_G\left(e=0\right)\right| &\leq \frac{1}{4}\ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-L/2} \\ &= \frac{1}{4}\ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-\ell/2}.\end{aligned}$$

– If e is a normal edge, the recursion in this case is only applied once and we do not decrease ℓ. Then the algorithm only deals with dangling edges. Consider the recursion we defined in Section 3.1:

$$g(x,y) = \frac{xy}{1-x-y+2xy}.$$

It holds that

$$\|\nabla g\|_1 = \frac{y(1-y)+x(1-x)}{(1-x-y+2xy)^2} \leq \frac{x+y}{(1-x)^2(1-y)^2} \leq \frac{2(1+\lambda)^3}{\lambda^4}$$

whenever $x, y \in \left(0, \frac{1}{1+\lambda}\right]$. Thus we have

$$\begin{aligned}\left|\mathbb{P}_G^{\ell}\left(e=0\right) - \mathbb{P}_G\left(e=0\right)\right| &\leq \frac{2(1+\lambda)^3}{\lambda^4}\max_{i\in\{1,2\}}\left|\mathbb{P}_{G_i}^{\ell}\left(e_i=0\right) - \mathbb{P}_{G_i}\left(e_i=0\right)\right| \\ &\leq \frac{(1+\lambda)^3}{2\lambda^4}\ln\left(1+\frac{1}{\lambda}\right)\cdot(1+\lambda)^{-\ell/2}.\end{aligned}$$

□

3.3 Putting all together

In this section, we prove Lemma 4. It follows from Lemma 5 and Lemma 8 that

(1) $\left|\mathbb{P}_G^{\ell}\left(e=0\right) - \mathbb{P}_G\left(e=0\right)\right| \leq \alpha\cdot(1+\lambda)^{-\ell/2}$ for some constant α; and
(2) $\mathbb{P}_G^{\ell}\left(e=0\right), \mathbb{P}_G\left(e=0\right) \leq \frac{1}{1+\lambda} < 1$.

Choosing $\ell = O(\log\frac{1}{\varepsilon})$ is sufficient to ensure

$$\exp(-\varepsilon)\cdot\hat{p} \leq \mathbb{P}_G\left(e=1\right) \leq \exp(\varepsilon)\cdot\hat{p}$$

where $\hat{p} = 1 - \mathbb{P}_G^{\ell}\left(e=0\right)$.

Now we bound the running time of Algorithm 1. Denote $T(\ell)$ the running time with recursion depth ℓ and denote n the size of the graph. Since we only branch into the case of normal edge once, the following recursion for the case of dangling edge dominates the running time of our algorithm:

$$T(\ell) = d\cdot T(\ell - \Theta(d)) + O(n)$$

where d is the degree of the dangling edge in consideration. Solving the recursion gives $T(\ell) = O(n\exp(\ell))$. Taking $\ell = O(\log\frac{1}{\varepsilon})$ concludes the proof.

References

1. Mohsen Bayati, David Gamarnik, Dimitriy Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of STOC*, pages 122–127. ACM, 2007.
2. Ivona Bezáková and William A Rummler. Sampling edge covers in 3-regular graphs. In *Proceedings of MFCS*, pages 137–148. Springer, 2009.
3. Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In *Proceedings of STOC*, pages 715–724, 2009.
4. Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of Holant problems. *SIAM Journal on Computing*, 40(4):1101–1132, 2011.
5. Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. *SIAM Journal on Computing*, 31(5):1527–1541, 2002.
6. Andreas Galanis, Qi Ge, Daniel Štefankovič, Eric Vigoda, and Linji Yang. Improved inapproximability results for counting independent sets in the hard-core model. *Random Structures & Algorithms*, 2012.
7. Mark Jerrum and Alistair Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems*, pages 482–520, 1996.
8. Liang Li, Pinyan Lu, and Yitong Yin. Approximate counting via correlation decay in spin systems. In *Proceedings of SODA*, pages 922–940. SIAM, 2012.
9. Liang Li, Pinyan Lu, and Yitong Yin. Correlation decay up to uniqueness in spin systems. In *Proceedings of SODA*, pages 67–84, 2013.
10. Chengyu Lin, Jingcheng Liu, and Pinyan Lu. A simple FPTAS for counting edge covers. In *Proceedings of SODA*, pages 341–348, 2014.
11. Jingcheng Liu and Pinyan Lu. FPTAS for counting monotone CNF. *arXiv preprint arXiv:1311.3728*, 2013.
12. Pinyan Lu, Menghui Wang, and Chihao Zhang. FPTAS for weighted Fibonacci gates and its applications. In *Proceedings of ICALP*, pages 787–799, 2014.
13. Ricardo Restrepo, Jinwoo Shin, Prasad Tetali, Eric Vigoda, and Linji Yang. Improved mixing condition on the grid for counting and sampling independent sets. *Probability Theory and Related Fields*, 156(1-2):75–99, 2013.
14. Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. In *Proceedings of SODA*, pages 941–953. SIAM, 2012.
15. Alistair Sinclair, Piyush Srivastava, and Yitong Yin. Spatial mixing and approximation algorithms for graphs with bounded connective constant. In *Proceedings of FOCS*, pages 300–309. IEEE, 2013.
16. Allan Sly. Computational transition at the uniqueness threshold. In *Proceedings of FOCS*, pages 287–296. IEEE, 2010.
17. Allan Sly and Nike Sun. The computational hardness of counting in two-spin models on d-regular graphs. In *Proceedings of FOCS*, pages 361–369. IEEE, 2012.
18. Leslie G Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal on Computing*, 31(4):1229–1254, 2002.
19. Leslie G Valiant. Holographic algorithms. *SIAM Journal on Computing*, 37(5):1565–1594, 2008.
20. Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of STOC*, pages 140–149. ACM, 2006.