A Simple FPTAS for Counting Edge Covers

Chengyu Lin [*]   Jingcheng Liu [†]   Pinyan Lu[‡]

Abstract

An edge cover of a graph is a set of edges such that every vertex has at least an adjacent edge in it. We design a very simple deterministic fully polynomial-time approximation scheme (FPTAS) for counting the number of edge covers for any graph. Previously, approximation algorithm is only known for 3 regular graphs and it is randomized. Our main technique is correlation decay, which is a powerful tool to design FPTAS for counting problems. In order to get FPTAS for general graphs without degree bound, we make use of a stronger notion called computationally efficient correlation decay, which is introduced in [18].

1   Introduction

An edge cover of a graph is a set of edges such that every vertex has at least an adjacent edge in it. For any graph without isolated vertices, there is at least one edge cover: the set of all edges. So the decision problem is trivial. There is also a polynomial time algorithm based on maximum machining to compute an edge cover with minimum cardinality. In this paper, we study the counting version: For a given input graph, we count the number of edge covers for that graph. Unlike the decision or optimization problem, counting edge covers is a #P-complete problem even when we restrict the input to 3 regular graphs. In this paper, we study the approximation version. For any given parameter $\epsilon > 0$, the algorithm outputs a number $\hat{N}$ such that $(1 - \epsilon)N \leq \hat{N} \leq (1 + \epsilon)N$, where N is the accurate number of edge covers of the input graph. We also require that the running time of the algorithm is bounded by $poly(n, 1/\epsilon)$, where n is the number of vertices of the given graph. This is called a fully polynomial-time approximation scheme (FPTAS). Our main result of this paper is an FPTAS for counting edge covers for any graph. Previously, approximation algorithm was only known for 3 regular graphs and the algorithm is randomized [3]. The randomized relaxation of FPTAS is called fully polynomial-time randomized approximation scheme (FPRAS), which uses random bits in the algorithm and requires that the final output is within the range $[(1 - \epsilon)N, (1 + \epsilon)N]$ with high probability.

Edge cover is related to many other graph problems such as (perfect) matching, k-factor problems and so on. All of them are talking about a set of edges which satisfies some local constraints defined on each vertex. For edge cover, it says that at least one incident edge should be chosen; while for matching, it is at most one edge. For generic constraints, it is the Holant framework [5, 6], which is well studied in terms of exactly counting [4, 6, 12], and recently in approximate counting [21, 22, 27]. For counting matchings, there is an FPRAS based on Markov Chain Monte Carlo (MCMC) for any graph [13]. Deterministic FPTAS is only known for graphs with bounded degree [2]. For counting perfect matchings, it is a long standing open question if there is an FPRAS (or FPTAS) for it. For bipartite graphs, there is an FPRAS for counting perfect matchings. The weighted version can be viewed as computing permanent of a non-negative matrix [15]. This is one great achievement of approximate counting. It is still widely

---

[*]Shanghai Jiaotong University. `linmrain@gmail.com`
[†]Shanghai Jiaotong University. `liuexp@gmail.com`
[‡]Microsoft Research Asia. `pinyanl@microsoft.com`

open if there exists an FPTAS for it or not. The current best deterministic algorithm can only approximate the permanent with an exponential large factor [10, 20]. There are many other counting problems, for which there is an FPRAS and we do not know if there is an FPTAS or not. In this paper, we give a complete FPTAS for a problem, for which even FPRAS was only known for very special family of instances.

Another view point of the edge cover problem is read twice monotone CNF formula (Rtw-Mon-CNF): Each edge is viewed as a Boolean variable and it is connected with two vertices (read twice); the constraint on each vertex is exactly a monotone OR constrain as at least one edge variable is assigned to be True. Counting number of solutions for a Boolean formula is another set of interesting problems studied both in exact counting and approximate counting. One famous example is the FPRAS for counting the solutions for a DNF formula [16, 17]. It is an important open question to derandomize it [11, 25]. Our FPTAS for counting edge covers can also be viewed as an FPTAS for counting solutions for a Rtw-Mon-CNF formula. If we do not restrict that each variable appears in at most two constraints, there is no FPTAS or FPRAS unless NP is equal to P or RP [8].

The common overall approach for designing approximate counting algorithms is to relate counting with probability distribution. This is usually referred as "counting vs sampling" paradigm when one mainly focuses on randomized counting. If we can compute (or estimate) the marginal probability, which in our problem is the probability of a given edge is chosen when we sample an edge cover uniformly at random, we can in turn approximate the counting. In randomized FPRAS, one estimates the marginal probability by sampling, and the most successful approach is sampling by Markov chain [14]. In deterministic FPTAS, one calculates the marginal probability directly, and the most successful approach is correlation decay as introduced in [1] and [26]. We elaborate a bit on the ideas. The marginal probability is estimated using only a local neighborhood around the edge. To justify the precision of the estimation, we show that far-away edges have little influence on the marginal probability. One most successful example is in anti-ferromagnetic two-spin systems [18, 19, 23], including counting independent sets [26]. The correlation decay based FPTAS is beyond the best known MCMC based FPRAS and achieves the boundary of approximability [9, 24]. To the best of our knowledge, that was the only example for which the best tractable range for correlation decay based FPTAS exceeds the sampling based FPRAS. This paper provides another such example. FPRAS was *the* solution concept for approximate counting [7]. The recent development of correlation decay based FPTAS is changing the picture. It is interesting to investigate the deep relation between these two approaches.

A number of tools were developed for establishing correlation decay property: self-avoiding walk tree, computation tree, potential function, dangling instance, bounded variables and so on. These are something like coupling argument, canonic path and so on to establish the rapidly mixing property of Markov Chains [14]. Armed with these powerful tools, there are recently many FPTAS's designed for various counting problems [18, 19, 21, 23, 27]. Many of these techniques are also used in this paper for designing and analyzing the FPTAS for counting edge covers.

Usually, the correlation decay property only implies FPTAS for system with bounded degree. The reason is that we need to explore a local neighborhood with radius of order $\log n$, then the total running time is $n^{\log n}$, which is not a polynomial, if there is no degree bound. To overcome this, we make use of a stronger notion called computationally efficient correlation decay as introduced in [18]. The observation is that when we go through a vertex with super-constant degree, the error is also decreased by a super-constant rate. Thus we do not need to explore a depth of $\log n$ if the degrees are large. The tradeoff relation between degree and decay rate defined by computationally efficient correlation decay can support FPTAS with unbounded degree systems. Previously, this notation was only used in anti-ferromagnetic two-spin systems. In this paper, we prove that the distribution defined by edge covers also satisfies this stronger

version of correlation decay and thus we give FPTAS for counting edge covers for any graph.

2 Preliminaries

2.1 Definitions

An edge cover of a graph is a set of edges such that every vertex has at least an adjacent edge in it. Given a graph $G = (V, E)$ with $e \in E$, we use $EC(G)$ to denote the set of all edge covers of graph G, and $P(G, e)$ to denote the marginal probability over $EC(G)$ that edge e is *not* chosen, or formally, with $X \sim EC(G)$ uniformly,

$$P(G, e) \triangleq \mathbb{P}(\text{edge } e \text{ is not chosen in } X) \tag{1}$$

In this paper, we deal with an extended notion of undirected graphs where dangling edges and free edges may be allowed.

Definition 1. *A **dangling edge** $e = (u, \_)$ of a graph is such singleton edge with only one end-point vertex u, as shown in the Figure 1a.*

*A **free edge** $e = (\_, \_)$ of a graph is such edge with no end-point vertex. Note that a free edge is not a dangling edge.*

We use graph to refer graph with or without dangling edges and free edges. Edges in the usual sense (i.e. neither dangling nor free), will be referred to as normal edges.

We remark that an alternative view to these combinatorial definitions is from Rtw-Mon-CNF. A dangling edge is simply a variable which only appears at one clause, and a free edge is a variable that does not appear at all, whereas normal edge just corresponds to variables appearing twice.

For a graph $G = (V, E)$, an edge $e = (u, v) \in E$ and a vertex $u \in V$, we define

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

Note that here in edge set E, duplicates are allowed. We may have multiple dangling edges $(v, \_)$, many free edges $(\_, \_)$ and even parallel edges between two vertices (u, v). For simplicity we treat $(v, \_)$ and $(\_, v)$ as the same.

For example, given a degree-3 vertex u with dangling edge e shown in Figure 1a , the result of $e_1 - u$ is shown in Figure 1b and the result of $G - e - u \triangleq (G - e) - u$ is shown in Figure 1c.
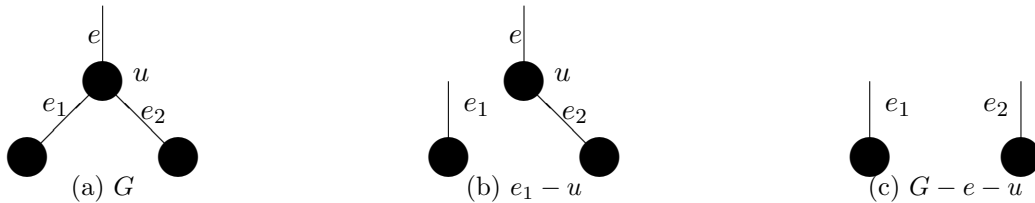

Figure 1: Dangling edges examples.

We use 0 to denote scalar value 0, and **0** to denote the all-zero vector, and $\{e_i\}_{i=1}^d$ denote the d-dimensional vector with i-th coordinate being e_i, so $\{e_i\} = \mathbf{0}$ means $\forall i, e_i = 0$. We also use the convention that when $d = 0, \prod_i^d p_i \triangleq 1$.

In general we use n to refer to the number of vertices in a given graph, and m to refer to the number of edges.

3   The Computation Tree Recursion

In this section, we provide a recursion for computing the marginal probability $P(G, e)$ with that of smaller instances.

3.1 e is free

Proposition 2.

$$P(G, e) = \frac{1}{2}$$

Proof. If e is a free edge, then any edge cover with e chosen is in one-to-one correspondence to an edge cover with e not chosen. Hence exactly half of the edge covers in $EC(G)$ does not choose e, so $P(G, e) = \frac{1}{2}$.   □

3.2 e is dangling

Lemma 3. *For graph $G = (V, E)$ with a dangling edge $e = (u, \_)$, denote the d edges incident with u except e as $e_1, e_2, \ldots, e_d$, let $G_i \triangleq G - e - u - \sum_{k=1}^{i-1} e_k$ (specifically, $G_1 \triangleq G - e - u$),*

$$P(G, e) = \frac{1 - \prod_{i=1}^d P(G_i, e_i)}{2 - \prod_{i=1}^d P(G_i, e_i)} \tag{2}$$

Proof. For $\boldsymbol{\alpha} \in \{0, 1\}^d$, let $EC_{\boldsymbol{\alpha}}(G - e - u)$ be the set of edge coverings in $G - e - u$ such that its restriction onto $\{e_i\}_{i=1}^d$ is consistent with $\boldsymbol{\alpha}$, denote $Z_{\boldsymbol{\alpha}} = \|EC_{\boldsymbol{\alpha}}(G - e - u)\|$, and $Z = \sum_{\boldsymbol{\alpha} \in \{0,1\}^d} Z_{\boldsymbol{\alpha}} = \|EC(G - e - u)\|$.

Also note that as long as $\boldsymbol{\alpha} \neq 0$, counting edge coverings with restriction $\boldsymbol{\alpha}$ is the same in either G, $G - e$, or $G - e - u$, so it's enough to work with $G - e - u$. Note that in $G - e - u$, for every i, e_i is either dangling or free, but not normal.

$$P(G, e) = \frac{\|EC(G - e)\|}{\|EC(G)\|} = \frac{\sum_{\boldsymbol{\alpha} \in \{0,1\}^d, \boldsymbol{\alpha} \neq \mathbf{0}} Z_{\boldsymbol{\alpha}}}{Z_{\mathbf{0}} + 2 \sum_{\boldsymbol{\alpha} \in \{0,1\}^d, \boldsymbol{\alpha} \neq \mathbf{0}} Z_{\boldsymbol{\alpha}}} = \frac{Z - Z_0}{2Z - Z_0} = \frac{1 - \frac{Z_{\mathbf{0}}}{Z}}{2 - \frac{Z_{\mathbf{0}}}{Z}}.$$

Now consider the term $\frac{Z_{\mathbf{0}}}{Z}$, it says the probability that a uniformly random edge cover drawn from $EC(G - e - u)$ picked none of $\{e_i\}_{i=1}^d$, so

$$\frac{Z_{\mathbf{0}}}{Z} = \mathbb{P}(\{e_i\} = \mathbf{0}) = \mathbb{P}(e_1 = 0) \prod_{i=2}^d \mathbb{P}\left(e_i = 0 \mid \{e_j\}_{j=1}^{i-1} = \mathbf{0}\right) = \prod_{i=1}^d P(G_i, e_i).$$

Hence by substitution we have $P(G, e) = \frac{1 - \prod_{i=1}^d P(G_i, e_i)}{2 - \prod_{i=1}^d P(G_i, e_i)}$.   □

We remark that for every i, e_i is dangling or free in G_i.

3.3 e is a normal edge

By definition we have

$$P(G, e) = \frac{\|EC(G - e)\|}{\|EC(G - e)\| + \|EC(G - e - u - v)\|}. \tag{3}$$

For $e = (u, v)$ as a normal edge, let $\{e_i\}$ be the set of edges incident with vertex u except e, and $\{f_i\}$ be the set of edges incident with vertex v except e, and $d_1 = \|\{e_i\}\|, d_2 = \|\{f_i\}\|$,
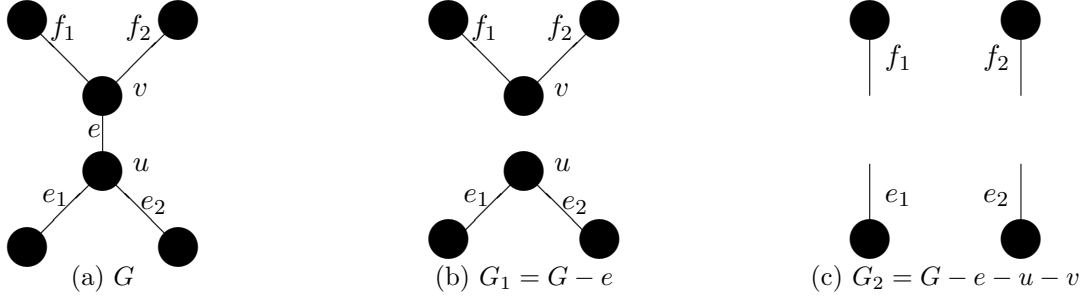
Figure 2: Normal edge examples.

now for $\boldsymbol{\alpha} \in \{0,1\}^{d_1}, \boldsymbol{\beta} \in \{0,1\}^{d_2}$, we use $E_{\boldsymbol{\alpha},\boldsymbol{\beta}}^G$ to denote the set of edge covers for G such that its restriction to $\{e_i\}_{i=1}^{d_1}$ is consistent with $\boldsymbol{\alpha}$, and restriction to $\{f_i\}_{i=1}^{d_2}$ is consistent with $\boldsymbol{\beta}$.

Denote $Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}^G \triangleq \|EC_{\boldsymbol{\alpha},\boldsymbol{\beta}}(G)\|$, $G_1 \triangleq G - e, G_2 \triangleq G - e - u - v$. As an illustration, given a normal edge $e = (u,v)$ in G as in Figure 2a, G_1 and G_2 are Figure 2b and Figure 2c respectively.

Note that as long as $\boldsymbol{\alpha} \neq \mathbf{0}, \boldsymbol{\beta} \neq \mathbf{0}$, working with G_1 and working with G_2 is the same with restriction to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, or formally,

$$\|EC(G-e)\| = \sum_{\boldsymbol{\alpha} \neq \mathbf{0}, \boldsymbol{\beta} \neq \mathbf{0}} Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{G_1} = \sum_{\boldsymbol{\alpha} \neq \mathbf{0}, \boldsymbol{\beta} \neq \mathbf{0}} Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{G_2}$$

Since only G_2 is involved, denote $Z_{\boldsymbol{\alpha},\boldsymbol{\beta}} \triangleq Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{G_2}, Z \triangleq \sum_{\boldsymbol{\alpha} \in \{0,1\}^{d_1}, \boldsymbol{\beta} \in \{0,1\}^{d_2}} Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}$, and $G_i^1 \triangleq G_2 - \sum_{k=1}^{i-1} e_k, G_i^2 \triangleq G_2 - \sum_{k=1}^{d_1} e_k - \sum_{k=1}^{i-1} f_k, G_i^3 \triangleq G_2 - \sum_{k=1}^{i-1} f_k$,

Lemma 4.

$$P(G,e) = 1 - \frac{1}{2 + \prod_{i=1}^{d_1} P(G_i^1, e_i) \cdot \prod_{i=1}^{d_2} P(G_i^2, f_i) - \prod_{i=1}^{d_1} P(G_i^1, e_i) - \prod_{i=1}^{d_2} P(G_i^3, f_i)} \quad (4)$$

Proof.

$$P(G,e) = \frac{\sum_{\boldsymbol{\alpha} \neq \mathbf{0}, \boldsymbol{\beta} \neq \mathbf{0}} Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}}{Z + \sum_{\boldsymbol{\alpha} \neq \mathbf{0}, \boldsymbol{\beta} \neq \mathbf{0}} Z_{\boldsymbol{\alpha},\boldsymbol{\beta}}}$$

$$= \frac{Z - \sum_{\boldsymbol{\alpha}} Z_{\boldsymbol{\alpha},\mathbf{0}} - \sum_{\boldsymbol{\beta}} Z_{\mathbf{0},\boldsymbol{\beta}} + Z_{\mathbf{0},\mathbf{0}}}{2Z - \sum_{\boldsymbol{\alpha}} Z_{\boldsymbol{\alpha},\mathbf{0}} - \sum_{\boldsymbol{\beta}} Z_{\mathbf{0},\boldsymbol{\beta}} + Z_{\mathbf{0},\mathbf{0}}}$$

$$= 1 - \frac{1}{2 + \mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}, \boldsymbol{\beta} = \mathbf{0}) - \mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}) - \mathbb{P}(\boldsymbol{\beta} = \mathbf{0})}$$

where $\mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}, \boldsymbol{\beta} = \mathbf{0}) \triangleq \frac{Z_{\mathbf{0},\mathbf{0}}}{Z}, \mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}) \triangleq \frac{\sum_{\boldsymbol{\beta}} Z_{\mathbf{0},\boldsymbol{\beta}}}{Z}, \mathbb{P}(\boldsymbol{\beta} = \mathbf{0}) \triangleq \frac{\sum_{\boldsymbol{\alpha}} Z_{\boldsymbol{\alpha},\mathbf{0}}}{Z}$.
Now consider the three terms respectively,

$$\mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}) = \mathbb{P}(\{e_i\} = \mathbf{0}) = \prod_{i=1}^{d_1} P(G_i^1, e_i)$$

$$\mathbb{P}(\boldsymbol{\beta} = \mathbf{0}) = \mathbb{P}(\{f_i\} = \mathbf{0}) = \prod_{i=1}^{d_2} P(G_i^3, f_i)$$

$$\mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}, \boldsymbol{\beta} = \mathbf{0}) = \mathbb{P}(\boldsymbol{\alpha} = \mathbf{0}) \cdot \mathbb{P}(\boldsymbol{\beta} = \mathbf{0} \mid \boldsymbol{\alpha} = \mathbf{0})$$

$$= \mathbb{P}(\{e_i\} = \mathbf{0}) \cdot \mathbb{P}(\{f_i\} = \mathbf{0} \mid \{e_i\} = \mathbf{0})$$

$$= \prod_{i=1}^{d_1} \mathbb{P}\left(e_i = 0 \mid \{e_j\}_{j=1}^{i-1} = \mathbf{0}\right) \cdot \prod_{i=1}^{d_2} \mathbb{P}\left(f_i = 0 \mid \{e_j\}_{j=1}^{d_1} = \mathbf{0}, \{f_j\}_{j=1}^{i-1} = \mathbf{0}\right)$$

$$= \prod_{i=1}^{d_1} P(G_i^1, e_i) \cdot \prod_{i=1}^{d_2} P(G_i^2, f_i)$$

5

Hence equation (4) is verified. □

Note that for every i, e_i is dangling or free in G_i^1, f_i is dangling or free in G_i^3, and in G_i^2, neither e_i nor f_i is normal.

4 Algorithm for Estimating Marginal Probability

Algorithm 1: Estimate $P(G, e)$ up to depth L

---
    function $P(G, e, L)$:

input : Graph G; edge e; Recursion depth L;
output: Estimate of $P(G, e)$ up to depth L .
begin
  if $L \leq 0$ **then**
    return $\frac{1}{2}$
  else if e *is free* **then**
    return $\frac{1}{2}$;
  else if e *is dangling* **then**
    $L' \leftarrow L - \lceil \log_6 (d+1) \rceil$;
    return $\frac{1 - \prod_{i=1}^{d} P(G_i, e_i, L')}{2 - \prod_{i=1}^{d} P(G_i, e_i, L')}$;
  else // e is normal
    $X \leftarrow \prod_{i=1}^{d_1} P(G_i^1, e_i, L)$;
    $Y \leftarrow \prod_{i=1}^{d_2} P(G_i^2, f_i, L)$;
    $Z \leftarrow \prod_{i=1}^{d_2} P(G_i^3, f_i, L)$;
    return $1 - \dfrac{1}{2 + X \cdot Y - X - Z}$;

We may compute the marginal probability $P(G, e)$ with the previous recursion, but that could take recursion depth of $O(n)$ which results in exponential computation time. So here we use a truncated computation tree for an estimate of $P(G, e)$.

We remark that the recursion depth used here is actually the so-called M-based depth introduced in [18] with $M = 6$.

Note that the normal case is invoked only once, so the algorithm keeps exploring in the third cases, until it hits the first 2 cases. We remark that an alternative view of the recursion depth is, we replace every node with degree greater than 6 with a 6-ary branching subtree. Now with this alternative view, it is easy to see that the nodes involved in the branching tree up to depth L is at most 6^L, and for the initial normal edge case it involves at most n subtrees, and for second-to-base-case nodes (i.e. nodes with $0 < L \leq \lceil \log_6 (d+1) \rceil$) they involve at most n extra base cases, so the algorithm $P(G, e, L)$ has running time $O(n^2 \cdot 6^L)$.

5 Correlation Decay Property

In the last section, we show an algorithm $P(G, e, L)$ for estimating the marginal probability $P(G, e)$, so here we establish the exponential correlation decay property, in the stronger sense with the M-based depth, of the estimation error in $P(G, e, L)$.

Theorem 5. *Given graph G, edge e and depth L,*

$$|P(G, e, L) - P(G, e)| \leq 3 \cdot (\frac{1}{2})^{L+1}$$

Such phenomenon is usually referred to as exponential correlation decay. Before we prove the main theorem, we will introduce a few useful propositions and lemmas.

Proposition 6.
$$P(G, e) \leq \frac{1}{2}$$

Proof. Although one may examine this case by case algebraically, this propositions can be seen quite obvious combinatorially in that, for any edge cover $X \in EC(G)$ s.t. $e \notin X$, $X + e$ is also an edge cover in G, and $\forall X, Y \in EC(G)$ s.t. $X \neq Y, e \notin X, e \notin Y$, we have $X + e \neq Y + e$. So the edge covers with e chosen is at least as many as the edge covers with e not chosen, hence the proposition follows. $\square$

We remark that our algorithm also guarantees that $P(G, e, L) \leq \frac{1}{2}$.

For notational convenience, given a d-dimensional vector $\mathbf{x} \in [0, \frac{1}{2}]^d$, we denote

$$f(\mathbf{x}) \triangleq \frac{1 - \prod_i x_i}{2 - \prod_i x_i}$$

Given a d_1-dimensional vector $\mathbf{x} \in [0, \frac{1}{2}]^{d_1}$ and two d_2-dimensional vectors $\mathbf{y}, \mathbf{z} \in [0, \frac{1}{2}]^{d_2}$, let

$$g(\mathbf{x}, \mathbf{y}, \mathbf{z}) \triangleq 1 - \frac{1}{2 + \prod_i x_i \cdot \prod_i y_i - \prod_i x_i - \prod_i z_i}$$

Lemma 7. *For d-variate function f, given estimated $\hat{\mathbf{x}}$ for true value $\mathbf{x}$ such that $\hat{\mathbf{x}} \in [0, \frac{1}{2}]^d, \mathbf{x} \in [0, \frac{1}{2}]^d$, let $\epsilon \triangleq \max_i |x_i - \hat{x}_i|$,*

$$|f(\hat{\mathbf{x}}) - f(\mathbf{x})| \leq \min \left\{ \frac{1}{2}, d \left(\frac{1}{2} \right)^{d-1} \right\} \cdot \epsilon$$

Proof. First we show for any d-dimensional vector $\mathbf{x} \in [0, \frac{1}{2}]^d$,

$$\sum_i^d \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| \leq \min \left\{ \frac{1}{2}, d \left(\frac{1}{2} \right)^{d-1} \right\} \tag{5}$$

For $d = 0, \sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = 0.$

For $d = 1, \sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \frac{1}{(2-x_1)^2} \leq \frac{4}{9}.$

For $d = 2, \sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \frac{x_1 + x_2}{(2-x_1 x_2)^2} \leq \frac{16}{49}.$

For $d = 3, \sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \frac{x_1 + x_2 + x_3}{(2-x_1 x_2 x_3)^2} \leq \frac{32}{75}.$

Now denote i^* be one of the indices of smallest x_i, since $x_i \leq \frac{1}{2}$, we have

$$\sum_i^d \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \sum_i^d \frac{\prod_{k \neq i}^d x_k}{(2 - \prod_i x_i)^2} \leq d \prod_{k \neq i^*}^d x_k \leq d \left(\frac{1}{2} \right)^{d-1}$$

So for $d \geq 4, \sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| \leq \frac{1}{2}$, we have verified the inequality relations (5).

Finally by (5) and Mean Value Theorem, $\exists \tilde{\mathbf{x}}$ s.t. $\forall i, \min \{x_i, \hat{x}_i\} \leq \tilde{x}_i \leq \max \{x_i, \hat{x}_i\}$.

$$|f(\hat{\mathbf{x}}) - f(\mathbf{x})| \leq \sum_i \left|\frac{\partial f(\tilde{\mathbf{x}})}{\partial x_i}\right| \cdot \epsilon \leq \min\left\{\frac{1}{2}, d\left(\frac{1}{2}\right)^{d-1}\right\} \cdot \epsilon$$

$\square$

Lemma 8. *Given estimated $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ for true value $\mathbf{x}, \mathbf{y}, \mathbf{z}$ respectively, such that $\mathbf{x} \in [0, \frac{1}{2}]^{d_1}, \mathbf{y}, \mathbf{z} \in [0, \frac{1}{2}]^{d_2}, \hat{\mathbf{x}} \in [0, \frac{1}{2}]^{d_1}, \hat{\mathbf{y}}, \hat{\mathbf{z}} \in [0, \frac{1}{2}]^{d_2}$, let $\epsilon \triangleq \max\{|x_i - \hat{x}_i|, |y_i - \hat{y}_i|, |z_i - \hat{z}_i|\}$,*

$$|g(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}) - g(\mathbf{x}, \mathbf{y}, \mathbf{z})| \leq 3\epsilon$$

Proof. First for any $\mathbf{x} \in [0, \frac{1}{2}]^{d_1}, \mathbf{y}, \mathbf{z} \in [0, \frac{1}{2}]^{d_2}$,

$$\sum_k^{d_1} \left|\frac{\partial g(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial x_k}\right| = \sum_k^{d_1} \frac{\prod_{i \neq k}^{d_1} x_i \cdot \left(1 - \prod_i^{d_2} y_i\right)}{(2 + \prod_i x_i \cdot \prod_i y_i - \prod_i x_i - \prod_i z_i)^2} \leq d_1 \cdot \frac{1}{2^{d_1-1}} \leq 1$$

$$\sum_k^{d_2} \left|\frac{\partial g(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial y_k}\right| = \sum_k^{d_2} \frac{\prod_{i \neq k}^{d_2} y_i \cdot \prod_i^{d_1} x_i}{(2 + \prod_i x_i \cdot \prod_i y_i - \prod_i x_i - \prod_i z_i)^2} \leq d_2 \cdot \frac{1}{2^{d_1+d_2-1}} \leq 1$$

$$\sum_k^{d_2} \left|\frac{\partial g(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial z_k}\right| = \sum_k^{d_2} \frac{\prod_{i \neq k}^{d_2} z_i}{(2 + \prod_i x_i \cdot \prod_i y_i - \prod_i x_i - \prod_i z_i)^2} \leq d_2 \cdot \frac{1}{2^{d_2-1}} \leq 1$$

Then by Mean Value Theorem, $\exists \tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}$ s.t. $\forall i, \min\{x_i, \hat{x}_i\} \leq \tilde{x}_i \leq \max\{x_i, \hat{x}_i\}, \min\{y_i, \hat{y}_i\} \leq \tilde{y}_i \leq \max\{y_i, \hat{y}_i\}, \min\{z_i, \hat{z}_i\} \leq \tilde{z}_i \leq \max\{z_i, \hat{z}_i\}$.

$$|g(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}) - g(\mathbf{x}, \mathbf{y}, \mathbf{z})| \leq \left(\sum_i \left|\frac{\partial g(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})}{\partial x_i}\right| + \sum_i \left|\frac{\partial g(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})}{\partial y_i}\right| + \sum_i \left|\frac{\partial g(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})}{\partial z_i}\right|\right) \epsilon \leq 3\epsilon$$

$\square$

Proof of Theorem 5. Note that the recursion for normal edge case is applied only once, so it is sufficient to show for free or dangling edge e,

$$|P(G, e, L) - P(G, e)| \leq (\frac{1}{2})^{L+1}, \text{for free or dangling edge } e$$

then the case of normal edge automatically follows from Lemma 8 that

$$|P(G, e, L) - P(G, e)| \leq 3 \cdot (\frac{1}{2})^{L+1}, \text{for normal edge } e$$

Now we prove by induction with induction hypothesis:

$$|P(G, e, L) - P(G, e)| \leq (\frac{1}{2})^{L+1}, \text{for free or dangling edge } e$$

For base case $L = 0, |P(G, e, L) - P(G, e)| \leq \frac{1}{2}$ holds when e is free or dangling.

Now suppose the induction hypothesis is true for $L < k$, we shall prove that it is true for $L = k$.

Case 1, e is free edge, then $|P(G, e, L) - P(G, e)| = 0$.

Case 2, $e = (u, \_)$ is dangling with $deg(u) = d + 1$, then by induction hypothesis we have $\epsilon \triangleq |P(G, e, L - \lceil \log_6 d + 1 \rceil) - P(G, e)| \leq \left(\frac{1}{2}\right)^{L - \lceil \log_6 d + 1 \rceil}$.

First by Lemma 7 we need to show that for $d \leq 4$,

$$\frac{1}{2^{1 + L - \lceil \log_6(d+1) \rceil}} \leq \frac{1}{2^L}$$

8

which is obvious because $\lceil \log_6 (d+1) \rceil \leq 1$.

Next we show for $d \geq 5$,

$$d \cdot \left(\frac{1}{2}\right)^{d-1+L-\lceil \log_6 (d+1) \rceil} \leq \left(\frac{1}{2}\right)^L$$

Namely for $d \geq 5$,

$$\log_2 d + \lceil \log_6 (d+1) \rceil \leq d - 1$$

For $d = 5, 6$, one can directly examine that as $\log_2 d < 3$ and $\log_6 6 = 1, \log_6 7 < 2$.

For $d \geq 7$, note that the function $f(x) = d - 2 - \log_2 d - \log_6 (d+1)$ is monotonically increasing, and $f(7) > 0$, so we have

$$\log_2 d + \log_6 (d+1) + 1 \leq d - 1$$

Therefore, the hypothesis for $L = k$ is verified.

$$|P(G, e, L) - P(G, e)| \leq (\frac{1}{2})^{L+1}, \text{for free or dangling edge } e$$

To sum up, the case of free or dangling edge and the case of normal edge together conclude the proof for our main theorem.

6   Counting Edge Covers

Finally, we present the procedures for approximately counting edge covers given good estimations of the marginal probability $P(G, e)$, hence an FPTAS for the approximate counting of edge covers problem.

Proposition 9. *Let* $Z(G) \triangleq \|EC(G)\|$ *and* $e_1, e_2, \ldots, e_m$ *be an enumeration of the edges* E. *Define* $G_1 \triangleq G, G_i \triangleq G_{i-1} - e_{i-1} - u_{i-1} - v_{i-1}, 1 < i \leq m$. *Then*

$$Z(G) = \frac{1}{\prod_{i=1}^m (1 - P(G_i, e_i))}$$

Proof. Recall that $EC(G) \neq \emptyset$, since the set of all edges E is an edge cover.

Now with $X \sim EC(G)$ uniformly, $\mathbb{P}(X = E)$ has two expressions,

$$\mathbb{P}(X = E) = \frac{1}{Z(G)}$$

$$\mathbb{P}(X = E) = \prod_i \mathbb{P}\left(e_i = 1 \mid \{e_j\}_{j=1}^{i-1} = \mathbf{1}\right)$$

$$= \prod_i (1 - P(G_i, e_i))$$

Therefore we have

$$Z(G) = \frac{1}{\prod_{i=1}^m (1 - P(G_i, e_i))}$$

$\square$

We now show the main theorem of this section. Let $Z(G, L) \triangleq \frac{1}{\prod_{i=1}^m (1 - P(G_i, e_i, L))}$ be the estimated number of edge covers given estimated $P(G_i, e_i, L)$

Theorem 10. *For* $0 < \epsilon < 1$, *take* $L = \log m + \log(8/\epsilon)$,

$$1 - \epsilon \leq \frac{Z(G, L)}{Z(G)} \leq 1 + \epsilon$$

Proof.

$$\frac{Z(G,L)}{Z(G)} = \prod_{i=1}^{m} \frac{1 - P(G_i, e_i)}{1 - P(G_i, e_i, L)}$$

By Theorem 5,

$$|P(G_i, e_i, L) - P(G_i, e_i)| \leq \frac{\epsilon}{4m}$$

Recall that $1 - P(G_i, e_i, L) \geq \frac{1}{2}$,

$$\frac{|P(G_i, e_i, L) - P(G_i, e_i)|}{1 - P(G_i, e_i, L)} \leq \frac{\epsilon}{2m}$$

Namely $\forall i$,

$$\left(1 - \frac{\epsilon}{2m}\right) \leq \frac{1 - P(G_i, e_i)}{1 - P(G_i, e_i, L)} \leq \left(1 + \frac{\epsilon}{2m}\right)$$

So we have

$$\left(1 - \frac{\epsilon}{2m}\right)^m \leq \prod_{i=1}^{m} \frac{1 - P(G_i, e_i)}{1 - P(G_i, e_i, L)} \leq \left(1 + \frac{\epsilon}{2m}\right)^m$$

$$1 - \epsilon \leq \frac{Z(G,L)}{Z(G)} \leq 1 + \epsilon$$

$\square$

To sum up, since $Z(G,L)$ involves m calls to $P(G,e,L)$, so run $Z(G,L)$ with $L = \log m + \log(8/\epsilon)$, is an FPTAS for counting edge covers with overall running time $O(m \cdot n^2 \cdot (m \cdot \frac{1}{\epsilon})^{\log_2 6})$.

References

[1] Antar Bandyopadhyay and David Gamarnik. Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models. *Random Structures & Algorithms*, 33(4):452–479, 2008.

[2] 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, 2007.

[3] Ivona Bezáková and William A Rummler. Sampling edge covers in 3-regular graphs. In *Mathematical Foundations of Computer Science 2009*, pages 137–148. Springer, 2009.

[4] Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures: extended abstract. In *STOC*, pages 635–644, 2013.

[5] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 715–724, New York, NY, USA, 2009. ACM.

[6] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of holant problems. *SIAM J. Comput.*, 40(4):1101–1132, 2011.

[7] Martin E. Dyer, Leslie Ann Goldberg, Catherine S. Greenhill, and Mark Jerrum. On the relative complexity of approximate counting problems. In Klaus Jansen and Samir Khuller, editors, *The 13th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, volume 1913 of *APPROX '00*, pages 108–119. Springer, 2000.

[8] Martin E. Dyer, Leslie Ann Goldberg, and Mark Jerrum. An approximation trichotomy for boolean #csp. *Journal of Computer and System Sciences*, 76(3-4):267–277, 2010.

[9] A. Galanis, D. Stefankovic, and E. Vigoda. Inapproximability of the partition function for the antiferromagnetic ising and hard-core models. *Arxiv preprint arXiv:1203.2226*, 2012.

[10] David Gamarnik and Dmitriy Katz. A deterministic approximation algorithm for computing the permanent of a 0, 1 matrix. *Journal of Computer and System Sciences*, 76(8):879–883, 2010.

[11] Parikshit Gopalan, Raghu Meka, and Omer Reingold. Dnf sparsification and a faster deterministic counting algorithm. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 126–135. IEEE, 2012.

[12] Sangxia Huang and Pinyan Lu. A dichotomy for real weighted holant problems. In *IEEE Conference on Computational Complexity*, pages 96–106, 2012.

[13] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM journal on computing*, 18(6):1149–1178, 1989.

[14] Mark Jerrum and Alistair Sinclair. *The Markov chain Monte Carlo method: an approach to approximate counting and integration*, pages 482–520. PWS Publishing Co., Boston, MA, USA, 1997.

[15] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51:671–697, July 2004.

[16] Richard M. Karp and Michael Luby. Monte-carlo algorithms for enumeration and reliability problems. In *FOCS*, pages 56–64, 1983.

[17] Richard M. Karp, Michael Luby, and Neal Madras. Monte-carlo approximation algorithms for enumeration problems. *J. Algorithms*, 10(3):429–448, 1989.

[18] Liang Li, Pinyan Lu, and Yitong Yin. Approximate counting via correlation decay in spin systems. In *Proceedings of SODA*, pages 922–940, 2012.

[19] Liang Li, Pinyan Lu, and Yitong Yin. Correlation decay up to uniqueness in spin systems. In *Proceedings of SODA*, pages 67–84, 2013.

[20] Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 644–652. ACM, 1998.

[21] Pinyan Lu, Menghui Wang, and Chihao Zhang. FPTAS for weighted fibonacci gates and its applications. *Submitted to SODA 2014*.

[22] Colin McQuillan. Approximating holant problems by winding. *CoRR*, abs/1301.2880, 2013.

[23] 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, 2012.

[24] 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, 2012.

[25] Luca Trevisan. A note on approximate counting for k-dnf. In *APPROX-RANDOM*, pages 417–426, 2004.

[26] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of STOC*, pages 140–149, 2006.

[27] Yitong Yin and Chihao Zhang. Approximate counting via correlation decay on planar graphs. In *Proceedings of SODA*, pages 47–66, 2013.