A Simple FPTAS for Counting Edge Covers

Chengyu Lin Jingcheng Liu Pinyan Lu

July 6, 2013

Abstract

An edge cover of a graph is a set of edges such that every vertex has at least an adjacent edge in it. Previously, approximation algorithm is only known for 3 regular graphs and it is randomized. We design a very simple deterministic fully polynomial-time approximation scheme (FPTAS) for counting the number of edge covers for any graph. 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 [Li, Lu, Yin SODA 2012].

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 an polynomial time algorithm based on maximum machining to compute a 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 output 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 a FPTAS for counting edge covers for any graph. Previously, approximation algorithm is only known for 3 regular graphs and the algorithm is randomized [**?**]. The randomized version of FPTAS is called FPRAS, which uses random bits in the algorithm and we require 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 [CLX09, CLX11], which is well studied in terms of exactly counting [CLX11, HL12, CGW13], recently in approximate counting [McQ13, **?**]. For counting matchings, there is a FPRAS based on Markov Chain Monte Carlo (MCMC) for any graph [] and deterministic FPTAS is only known for graphs with bounded degree [BGK+07]. For counting perfect matchings, it is a long standing open question if there is a FPRAS or FPTAS for it. For bipartite graphs, there is a FPRAS for counting perfect matchings []. The weighted version can be viewed as computing permanent of a non-negative matrix. This is one great achievement of approximate counting. It is still widely open if there exists a FPTAS for it or not. The current best deterministic algorithm can only approximate the permanent with an exponential large factor []. There are many other counting problems, where there is a FPRAS and we do not know if there is a FPTAS or not []. In this paper, we give a complete FPTAS for a problem, where 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 []. It is an important open question to derandomized it []. Our FPTAS for counting edge covers can also be viewed as a 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 [].

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 (estimate) the marginal probability, which in our problem is the probability of a given edge is chosen when we sample a edge cover uniformly at random, we can in turn to approximate count. In randomized FPRAS, one estimate the marginal probability by sampling, and the most successful approach is sampling by Markov chain. In deterministic FPTAS, one calculate the marginal probability directly, and the most successful approach is correlation decay as introduce in [BG08] and [Wei06]. 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 distribution. One most successful example is in anti-ferromagnetic two-spin systems [], including counting independent sets [Wei06]. The correlation decay based FPTAS is beyond the best known MCMC based FPRAS and achieves the boundary of tractable and intractable []. 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 offers another such example. FPRAS was *the* solution concept for approximate counting [], the recently development of correlation decay based FPTAS is changing the picture. It is interesting to investigate the deep relation between these two approaches.

A set of tools was developed for establishing correlation decay property. These are something like coupling argument, canonic path and so on for establish to the rapid mixing property for Markov Chains. They are self-avoid walk tree, computational tree, potential function, bounded variables and so on. Armed with these powerful tools, there are recently many FPTAS designed for many counting problems []. Many of these techniques are also used in this paper for designing and analyzing the FPRAS 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 stronger notion called computationally efficient correlation decay as introduced in [LLY12]. The observation is that when we go through a vertex with sup 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.*

So we use graph to refer graphs with or without dangling edges or free edges. Edges in the usual sense (i.e. not dangling and not free), will be refered to as normal edges.

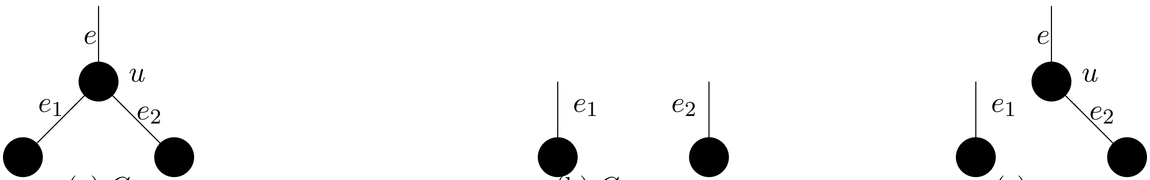
We remark that an alternative view to these combinatorial definitions is from Rtw-Mon-CNF. A dangling edge is just a variable which only appears at one clause, and a free edge is a variable that does not appear at all, whereas normal edge just corresponds to variables appearing twice.

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

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

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 example, given a degree-3 vertex u with dangling edge e shown in Figure 1a , the result of $G - e - u$ is shown in Figure 1b.



(a) G (b) $G - e - u$ (c) $e_1 - u$

Figure 1: Dangling edges examples.

We use 0 to denote scalar value 0, and $\mathbf{0}$ to denote vector value 0, 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 Recursion

First we show the recursion for computing the marginal probability $P(G, e)$.

3.1 e is free edge

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 any edge cover with e not chosen. Hence exactly half of the edge covers in $EC(G)$ doesn't choose e, so $P(G, e) = \frac{1}{2}$. □

3.2 e is dangling

Proposition 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 $G_i = G - e - u - \sum_{k=1}^{i-1} e_k$ (specifically, $G_1 = G - e - u$), we have*

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

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

Also note that as long as $\alpha \neq 0$, counting edge coverings with restriction α 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.

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

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

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

Hence concludes the proof.

∎

3.3 e is 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 $\{f_i\}$ is the set of edges incident with vertex v except e, and $d_1 = \|\{e_i\}\|, d_2 = \|\{f_i\}\|$, now for $\alpha \in \{0,1\}^{d_1}, \beta \in \{0,1\}^{d_2}$, we use $E^G_{\alpha,\beta}$ to denote the set of edge coverings in G such that its restriction to $\{e_i\}_{i=1}^{d_1}$ is consistent with α, and restriction to $\{f_i\}_{i=1}^{d_2}$ is consistent with β.

Denote $Z^G_{\alpha,\beta} \triangleq \|EC_{\alpha,\beta}(G)\|$, $G_1 \triangleq G-e, G_2 \triangleq G-e-u-v$, now for $\alpha \neq \mathbf{0}, \beta\neq\mathbf{0}$, we also have working with G_1 and working with G_2 is the same with restriction to α and β, or formally,

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

Since we are only working with G_2, we may simply denote $Z_{\alpha,\beta} \triangleq Z^{G_2}_{\alpha,\beta}$, now let $Z = \sum_{\alpha,\beta} Z_{\alpha,\beta}$, and $\mathbb{P}(\alpha=0,\beta=0) \triangleq \frac{Z_{\mathbf{0},\mathbf{0}}}{Z}, \mathbb{P}(\alpha=\mathbf{0}) \triangleq \frac{\sum_\beta Z_{\mathbf{0},\beta}}{Z}, \mathbb{P}(\beta=\mathbf{0}) \triangleq \frac{\sum_\alpha Z_{\alpha,\mathbf{0}}}{Z}$ we have,

Proposition 4.

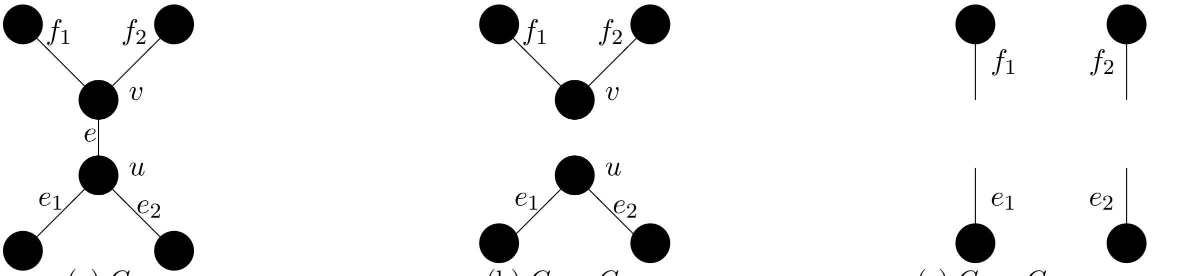
$$P(G,e) = 1 - \frac{1}{2+\mathbb{P}(\alpha=0,\beta=0) - \mathbb{P}(\alpha=0) - \mathbb{P}(\beta=0)}$$

Proof.

$$\begin{aligned}
P(G,e) &= \frac{\sum_{\alpha\neq\mathbf{0},\beta\neq\mathbf{0}} Z_{\alpha,\beta}}{Z + \sum_{\alpha\neq\mathbf{0},\beta\neq\mathbf{0}} Z_{\alpha,\beta}} \\
&= \frac{Z - \sum_\alpha Z_{\alpha,\mathbf{0}} - \sum_\beta Z_{\mathbf{0},\beta} + Z_{\mathbf{0},\mathbf{0}}}{2Z - \sum_\alpha Z_{\alpha,\mathbf{0}} - \sum_\beta Z_{\mathbf{0},\beta} + Z_{\mathbf{0},\mathbf{0}}} \\
&= 1 - \frac{1}{2+\mathbb{P}(\alpha=0,\beta=0) - \mathbb{P}(\alpha=0) - \mathbb{P}(\beta=0)}
\end{aligned}$$

∎

Let
$G_i^1 \triangleq G-e-u-v-\sum_{k=1}^{i-1} e_k$,
$G_i^2 \triangleq G-e-u-v-\sum_{k=1}^{d_1} e_k - \sum_{k=1}^{i-1} f_k$,
$G_i^3 \triangleq G-e-u-v-\sum_{k=1}^{i-1} f_k$,
so we have



(a) G (b) $G_1 = G - e$ (c) $G_2 = G - e - u - v$

Figure 2: Normal edge examples.

Proposition 5.

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

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

$$\mathbb{P}(\alpha = 0, \beta = 0) = \prod_{i=1}^{d_1} P(G_i^1, e_i) \cdot \prod_{i=1}^{d_2} P(G_i^2, f_i)$$

Proof.

$$\begin{aligned}
\mathbb{P}(\alpha = 0) =& \mathbb{P}(\{e_i\} = \mathbf{0}) = \prod_{i=1}^{d_1} P(G_i^1, e_i) \\
\mathbb{P}(\beta = 0) =& \mathbb{P}(\{f_i\} = \mathbf{0}) = \prod_{i=1}^{d_2} P(G_i^3, f_i) \\
\mathbb{P}(\alpha = 0, \beta = 0) =& \mathbb{P}(\alpha = 0) \cdot \mathbb{P}(\beta = 0 \mid \alpha = 0) \\
=& \mathbb{P}(\{e_i\} = \mathbf{0}) \cdot \mathbb{P}(\{f_i\} = \mathbf{0} \mid \{e_i\} = \mathbf{0}) \\
=& \prod_{i=1}^{d_1} \mathbb{P}\left(e_i = 0 \mid \{e_j\}_{j=1}^{i-1} = \mathbf{0}\right) \cdot \prod_{i=1}^{d_2} \mathbb{P}\left(f_i = 0 \mid \{e_j\}_{j=1}^{d_1} = \mathbf{0}, \{f_j\}_{j=1}^{i-1} = \mathbf{0}\right) \\
=& \prod_{i=1}^{d_1} P(G_i^1, e_i) \cdot \prod_{i=1}^{d_2} P(G_i^2, f_i)
\end{aligned}$$

□

Corollary 6.

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

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 Computation Tree for Marginal Probability

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 $P(G, e)$.

Note the recursion depth used here is actually the so-called M-based depth introduced in [LLY12] with $M = 6$. We remark that actually M could take any value as long as $M \geq 6$.

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 $1 - \dfrac{1}{2 + X \cdot Y - X - Z}$;

4.1 Analysis of the Algorithm

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. Let $B(L)$ be the set of vertices in the recursion tree involved, and $R(L) \triangleq \|B(L)\|$, by the recursion on $P(G, e, L)$ in the third case we have the recursive relation for $R(L)$,

$$R(L) \leq dR(L - \lceil \log_6 (d+1) \rceil), L > 0$$
$$R(L) = 1, L \leq 0$$

Therefore we conclude that $R(L) \leq d^{1+\frac{L}{\log_6 (d+1)}} \leq d \cdot 6^L$, in other words, the running time of the above algorithm with recursion depth L is at most $O(n \cdot 6^L)$.

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 7. *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 refered to as exponential correlation decay. Before we prove the main theorem, we will introduce a few useful propositions and lemmas.

Proposition 8.

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

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 $P(G,e,L)$, since $\frac{1-\prod_i x_i}{2-\prod_i x_i} = \frac{1}{2} - \frac{\prod_i x_i}{2(2-\prod_i x_i)}$, and $X \cdot Y - X - Z \leq 0$.

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 9. *For d-variate function $f(\mathbf{x})$,and a d-dimensional vector $\mathbf{x} \in [0, \frac{1}{2}]^d$,*

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

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

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

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

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

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

Next consider $d = 2$, $\sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \frac{x_1+x_2}{(2-x_1x_2)^2} \leq \frac{16}{49}$.

Finally for $d = 3$, $\sum_i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \frac{x_1+x_2+x_3}{(2-x_1x_2x_3)^2} \leq \frac{32}{75}$. □

Lemma 10. *Given a d_1-dimensional vector $\mathbf{x} \in [0, \frac{1}{2}]^{d_1}$ and two d_2-dimensional vectors $\mathbf{y}, \mathbf{z} \in [0, \frac{1}{2}]^{d_2}$,*

$$\begin{aligned}
\sum_i \left| \frac{\partial g(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial x_i} \right| \leq& 1 \\
\sum_i \left| \frac{\partial g(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial y_i} \right| \leq& 1 \\
\sum_i \left| \frac{\partial g(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial z_i} \right| \leq& 1
\end{aligned}$$

Proof.

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

□

Now we are ready for the main theorem.

Proof. First we note that by Mean Value Theorem, given estimated $\hat{\mathbf{x}}$, let $\mathbf{x}$ be the true value, let $\epsilon = \max_i |x_i - \hat{x_i}|$. we have for d-variate function f,

$$|f(\hat{\mathbf{x}}) - f(\mathbf{x})| \leq \sum_i \left|\frac{\partial f(\mathbf{x})}{\partial x_i}\right| \cdot \epsilon \leq d \cdot \left(\frac{1}{2}\right)^{d-1} \cdot \epsilon \tag{4}$$

Given estimated $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$, let $\epsilon = \max\{|x_i - \hat{x_i}|, |y_i - \hat{y_i}|, |z_i - \hat{z_i}|\}$.

$$|g(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}) - g(\mathbf{x},\mathbf{y},\mathbf{z})| \leq \left(\sum_i \left|\frac{\partial g(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial x_i}\right| + \sum_i \left|\frac{\partial g(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial y_i}\right| + \sum_i \left|\frac{\partial g(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial z_i}\right|\right) \cdot \epsilon \leq 3\epsilon \tag{5}$$

Also note that the recursion for normal edge case is applied only once, so it's 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 5 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 simple induction. 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. When e is normal, since the normal case only appears once, we have the maximal estimation error for the first two cases that $\epsilon \leq \frac{1}{2}$, so to sum up we have $|P(G,e,L) - P(G,e)| \leq 3 \cdot \frac{1}{2}$.

Now suppose for $L < k$ we have the induction hypothesis true, now we try to show it's 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 $\epsilon \leq \frac{1}{2}^{L-\lceil\log_6 d+1\rceil}$.

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

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

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 $\log_6 (d+1) < 2$.

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 second part of 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 11. *Let $Z(G) \triangleq \|EC(G)\|$, order the set of edges in G in any order as $\{e_i\}$, define $G_1 \triangleq G, G_i \triangleq G_{i-1} - e_{i-1} - u_{i-1} - v_{i-1}, 1 < i \leq m$.*

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

Proof. Note 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 equivalent expressions,

$$\begin{aligned}
\mathbb{P}(X = E) =& \frac{1}{Z(G)} \\
\mathbb{P}(X = E) =& \prod_i \mathbb{P}\left(e_i = 1 \mid \{e_j\}_{j=1}^{i-1} = \mathbf{1}\right) \\
=& \prod_i (1 - P(G_i, e_i))
\end{aligned}$$

Therefore we have

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

□

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

Theorem 12. *For $0 < \epsilon < 1$, take $L = \log m + \log(8/\epsilon)$,*

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

Proof.

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

By Theorem 7,

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

Since $1 - P(G_i, e_i) \ge \frac{1}{2}$,

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

Namely $\forall i$,

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

So we have

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

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

□

To sum up, run $Z(G, L)$ with $L = \log m + \log(8/\epsilon)$, is the FPTAS for counting edge covers, the total running time is $O(m \cdot n \cdot (m \cdot \frac{1}{\epsilon})^{\log_2 6})$.

References

[BG08] Antar Bandyopadhyay and David Gamarnik, *Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models*, Random Structures & Algorithms **33** (2008), no. 4, 452–479.

[BGK⁺07] Mohsen Bayati, David Gamarnik, Dimitriy Katz, Chandra Nair, and Prasad Tetali, *Simple deterministic approximation algorithms for counting matchings*, Proceedings of STOC, 2007, pp. 122–127.

[CGW13] Jin-Yi Cai, Heng Guo, and Tyson Williams, *A complete dichotomy rises from the capture of vanishing signatures: extended abstract*, STOC, 2013, pp. 635–644.

[CLX09] Jin-Yi Cai, Pinyan Lu, and Mingji Xia, *Holant problems and counting CSP*, Proceedings of the 41st annual ACM symposium on Theory of computing (New York, NY, USA), STOC '09, ACM, 2009, pp. 715–724.

[CLX11] Jin-Yi Cai, Pinyan Lu, and Mingji Xia, *Computational complexity of holant problems*, SIAM J. Comput. **40** (2011), no. 4, 1101–1132.

[HL12] Sangxia Huang and Pinyan Lu, *A dichotomy for real weighted holant problems*, IEEE Conference on Computational Complexity, 2012, pp. 96–106.

[LLY12] Liang Li, Pinyan Lu, and Yitong Yin, *Approximate counting via correlation decay in spin systems*, Proceedings of SODA, 2012, pp. 922–940.

[McQ13] Colin McQuillan, *Approximating holant problems by winding*, CoRR **abs/1301.2880** (2013).

[Wei06] Dror Weitz, *Counting independent sets up to the tree threshold*, Proceedings of STOC, 2006, pp. 140–149.