A Simple FPTAS for Counting Edge Covers

Chengyu Lin   Jingcheng Liu   Pinyan Lu

July 4, 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 a given input graph, we count the number of edge covers for that graph. This 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 $(1-\epsilon)N \le \hat{N} \le (1+\epsilon)N$, 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 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 algorithms 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, and k-factor problems. All of them are talking about a set of edges which satisfies some local constraint 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, which is well studied in terms of exactly counting, recently in approximate counting. For counting matchings, there is a FPRAS based on MCMC and deterministic FPTAS is only known for graphs with bounded degree. 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. The weighted version is exactly computing permanent of a non-negative matrix. This is one great achievement. It is still widely open if there exists a FPTAS for it. 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 such as counting the number of solution for a DNF formula []. In this paper, we give a complete FPTAS for a problem, where even FPRAS is only known for very special family of graphs.

Another view point of edge cover 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 CNF 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 problem studied both in exact counting and approximate counting. One famous example is the FPRAS for counting the solutions for a DNF formula. It is important open question if we can derandomized it. Our FPTAS for counting edge covers can also be viewed as a FPTAS for counting the 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, we estimate the marginal probability by sampling, and the most successful approach is sampling by Markov chain. In FPTAS, one calculate the marginal probability directly, and the most successful approach is correlation decay as introduce in [**?**] and [**?**]. 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 system, including counting independent set. The correlation decay based FPTAS is beyond the best known MCMC based FPRAS and approaches 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 question to establish more 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 rapid mixing for Markov Chains. There are Self avoid walk tree, computational tree, potential function, bounded variables and so on. Armed with these powerful tools, there are recently many FPTAS were designed for many counting problems. Many of these techniques are also used in designing and analyzing the FPRAS for counting edge covers.

Usually, the correlation decay property only implies FPTAS for system with bounded degree such as []. The reason is that we need to explore a local neighborhood with radius of order $\log n$, then the total running time is sup polynomial $n^{\log n}$ if there is no degree bound. To overcome this, we make use of stronger notion called computationally efficient correlation decay as introduced in [**?**]. The observation is that we will go through a vertex with sup constant degree, the error is also decreased by a supper constant. 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 system. Previously, this notation is only used in anti-ferromagnetic two-spin system. 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 edge 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 in the usual sense 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, and graphs with only normal edges(i.e. graphs in the usual sense) will be refered to as normal graphs.

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.

2.2 Notations

For a graph $G = (V, E)$ and an edge (may or may not be normal) $e \in E$, let $G - e \triangleq (V, E - e)$, and $G - v \triangleq (V - \{v\}, \{(x, y) : x \neq v, y \neq v, (x, y) \in E\} \cup \{(x, \_) : (v, x) \in E\})$.

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) $G - e - u$

Figure 1: Dangling graphs 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 $\forall i, e_i = 0$.

3 The Recursion

Now we're ready for the recursion.

3.1 Nontrivial Dangling Graphs

Proposition 2. *For nontrivial dangling 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, $G_1 = G - e - u$), so we have*

$$P(G,e) = \frac{1 - \prod_{i=1}^{d} P(G_i, e_i)}{2 - \prod_{i=1}^{d} P(G_i, e_i)} = \frac{1}{2} - \frac{0.5 \prod_{i=1}^{d} P(G_i, e_i)}{2 - \prod_{i=1}^{d} P(G_i, e_i)} \tag{2}$$

Proof. For $\alpha \in \{0,1\}^d$, let E_α 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 $Z_\alpha = \|E_\alpha\|$, and $Z = \sum_{\alpha \in \{0,1\}^d} Z_\alpha$.

Also note that 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 $G - e - u$.

$$\begin{aligned} P(G,e) =& \frac{\text{number of solutions with } e \text{ not chosen}}{\text{total number of solutions}} \\ =& \frac{\sum_{\alpha \in \{0,1\}^d, \alpha \neq \mathbf{0}} Z_\alpha}{Z_{\mathbf{0}} + 2\sum_{\alpha \in \{0,1\}^d, \alpha \neq \mathbf{0}} Z_\alpha} \\ =& \frac{1 - \frac{Z_{\mathbf{0}}}{Z}}{2 - \frac{Z_{\mathbf{0}}}{Z}}. \end{aligned}$$

Now consider the term $\frac{Z_{\mathbf{0}}}{Z}$, it says the probability over all solutions in $G - e - u$ that none of $\{e_i\}_{i=1}^d$ is picked, 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. □

Corollary 3. *For nontrivial dangling graphs,*

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

In fact this corrolary comes no surprise, because by looking combinatorially, picking a dangling edge should definitely yields more solutions.

As a side note, note that $\forall i, G_i$ is a dangling graph (maybe trivial dangling graphs though), although e_i can be a free edge.

3.2 Trivial Dangling Graphs

TBA.

3.3 General Graphs

Here we focus on graphs with no dangling edges and no free edges, as trivial dangling graphs is just trivial base cases, and nontrivial dangling graphs has been handled in the previous section.

Here's a typical example of converting a general graph to dangling graphs. Say we picked $e = (u, v)$ out of any general graph as in figure 2a, again we want to write the recursion of $P(G, e)$ for G. By definition we have

$$P(G, e) = \frac{(\text{number of solutions in } G - e)}{(\text{number of solutions in } G - e) + (\text{number of solutions in } G - e - u - v)}. \tag{3}$$

For $\alpha \in \{0,1\}^{d_1}, \beta \in \{0,1\}^{d_2}$, let $E^G_{\alpha,\beta}$ be 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, where $\{e_i\}$ is 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 $d_1 = \|\{e_i\}\|, d_2 = \|\{f_i\}\|$.

Denote $Z^G_{\alpha,\beta} \triangleq \left\|E^G_{\alpha,\beta}\right\|$, $G_1 \triangleq G - e, G_2 \triangleq G - e - u - v$, C_1 be the number of solutions in G_1, C_2 be the number of solutions in G_2, now we have

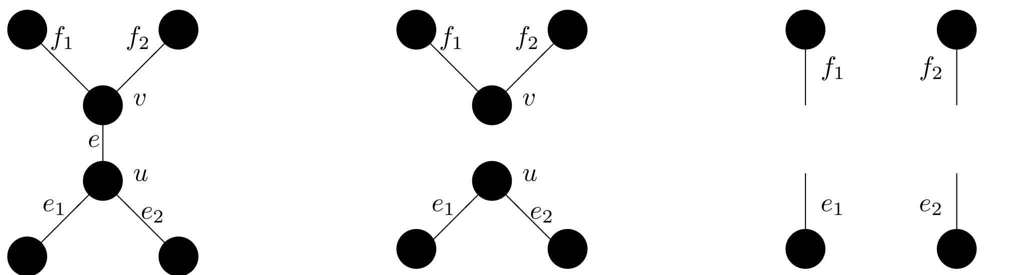
Proposition 4.

$$C_1 = \sum_{\alpha \neq \mathbf{0}, \beta \neq \mathbf{0}} Z^{G_1}_{\alpha,\beta} = \sum_{\alpha \neq \mathbf{0}, \beta \neq \mathbf{0}} Z^{G_2}_{\alpha,\beta}$$

$$C_2 = \sum_{\alpha,\beta} Z^{G_2}_{\alpha,\beta}$$

And denote $Z = \sum_{\alpha,\beta} Z^{G_2}_{\alpha,\beta}$, *we also have,*

$$P(G, e) = \frac{\sum_{\alpha \neq \mathbf{0}, \beta \neq \mathbf{0}} Z^{G_2}_{\alpha,\beta}}{Z + \sum_{\alpha \neq \mathbf{0}, \beta \neq \mathbf{0}} Z^{G_2}_{\alpha,\beta}} = 1 - \frac{1}{2 + \mathbb{P}(\alpha = 0, \beta = 0) - \mathbb{P}(\alpha = 0) - \mathbb{P}(\beta = 0)}$$



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

Figure 2: General graphs examples.

First consider the term $\mathbb{P}(\alpha = 0, \beta = 0) = \frac{\sum_{\alpha = \mathbf{0}, \beta = \mathbf{0}} Z^{G_2}_{\alpha,\beta}}{Z}$, it says the probability in edge coverings of G_2 that none of the edges $\{e_i\}$ and none of $\{f_i\}$ is chosen.

Let

$G^1_i \triangleq G - e - u - v - \sum_{k=1}^{i-1} e_k$,

$G^2_i \triangleq G - e - u - v - \sum_{k=1}^{d_1} e_k - \sum_{k=1}^{i-1} f_k$,

$G^3_i \triangleq G - e - u - v - \sum_{k=1}^{i-1} f_k$,

so we have

Proposition 5.

$$
\begin{aligned}
\mathbb{P}(\alpha = 0) &= \prod_{i=1}^{d_1} P(G_i^1, e_i) \\
\mathbb{P}(\beta = 0) &= \prod_{i=1}^{d_1} P(G_i^3, f_i) \\
\mathbb{P}(\alpha = 0, \beta = 0) &= \prod_{i=1}^{d_1} P(G_i^1, e_i) \cdot \prod_{i=1}^{d_2} P(G_i^2, f_i)
\end{aligned}
$$

Proof.

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

□

Note that $\forall i$, both G_i^1, G_i^2 must be dangling graphs.(FIXME [1])

An over-simplified version of the computation tree and algorithm naturally follows. Despite its simplicity, it's FPTAS for constant bounded degree graph. In the next section, we establish its correlation decay property, and overcome the degree bound by a stronger notion called computationally efficient correlation decay.

[1]Can we say nontrivial dangling graphs if we restrict graphs to have at least 6 vertices?

Algorithm 1: Tree-depth based estimate of $P(G, e)$

function $\hat{P}(G, e, C, L)$:

input : Graph G; edge e; boundary configuration C; recursion depth L;
output: Estimate of $P(G, e)$ using boundary configuration C.
begin
    if G *is trivial dangling graphs* **then**
        return $C \upharpoonright e$
    else if G *is non-dangling graphs* **then**
        $X \leftarrow \prod_{i=1}^{d_1} \hat{P}(G_i^1, e_i, C, L)$;
        $Y \leftarrow \prod_{i=1}^{d_2} \hat{P}(G_i^2, f_i, C, L)$;
        $Z \leftarrow \prod_{i=1}^{d_2} \hat{P}(G_i^3, f_i, C, L)$;
        return $1 - \dfrac{1}{2 + X \cdot Y - X - Z}$;
    else if e *is free* **then**
        return $\frac{1}{2}$;
    else // e is dangling and G is nontrivial dangling
        $L' \leftarrow L - 1$;
        return $\frac{1 - \prod_{i=1}^{d} \hat{P}(G_i, e_i, C, L')}{2 - \prod_{i=1}^{d} \hat{P}(G_i, e_i, C, L')}$;

4 Correlation Decay Property

In this section, we establish the correlation decay property first on nontrivial dangling graphs, then generalize it to general graphs and graphs without degree bound.

Definition 6. *The recursion on $P(G, e, C, L)$ is said to exhibit exponential correlation decay phenomenon if for every two boundary conditions C, C' and some constant $\alpha < 1$,*

$$|P(G, e, C, L) - P(G, e, C', L)| = O(\alpha^L)$$

FIXME: is this equivalent to saying L is just the distance of e from the set of edges in which C, C' differs?

4.1 Nontrivial Dangling Graphs

Proposition 7. *If every G_i is nontrivial dangling graphs, and e_i is not free,*

$$\left| \sum_i \frac{\partial P(G, e)}{\partial P(G_i, e_i)} \right| < 1$$

Proof. Denote i^* be one of the indices of smallest $P(G_i, e_i)$, since for nontrivial dangling graphs,

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

$$\begin{aligned}\left|\sum_i \frac{\partial P(G,e)}{\partial P(G_i,e_i)}\right| =& \frac{\sum_i \prod_{k\neq i} P(G_k,e_k)}{\left(2-\prod_i P(G_i,e_i)\right)^2} \\ \leq& d \prod_{k\neq i^*} P(G_k,e_k) \\ \leq& d\left(\frac{1}{2}\right)^{d-1}\end{aligned}$$

So for $d \geq 3$ we already have $\left|\sum_i \frac{\partial P(G,e)}{\partial P(G_i,e_i)}\right| < 1$.

Note that $d \geq 1$ because the graph is nontrivial.

Now first consider $d = 1$, $\left|\sum_i \frac{\partial P(G,e)}{\partial P(G_i,e_i)}\right| = \frac{1}{(2-P(G_1,e_1))^2} < \frac{4}{9}$.

Next consider $d = 2$, $\left|\sum_i \frac{\partial P(G,e)}{\partial P(G_i,e_i)}\right| = \frac{P(G_1,e_1)+P(G_2,e_2)}{(2-P(G_1,e_1)P(G_2,e_2))^2} < \frac{16}{49}$. □

4.2 General Graphs

Note that the recursion for general graph is applied only once, so it's sufficient to show that the sum of the partial derivatives is bounded.

Proposition 8.

$$\begin{aligned}\frac{\partial \mathbb{P}\left(\alpha=0,\beta=0\right)}{\partial P(G_i^1,e_i)} &\leq (\frac{1}{2})^{d_1+d_2-1} \\ \frac{\partial \mathbb{P}\left(\alpha=0,\beta=0\right)}{\partial P(G_i^2,f_i)} &\leq (\frac{1}{2})^{d_1+d_2-1}\end{aligned}$$

Proof.

$$\begin{aligned}\frac{\partial \mathbb{P}\left(\alpha=0,\beta=0\right)}{\partial P(G_k^1,e_k)} =& \frac{\prod_{i=1}^{d_1} P(G_i^1,e_i)\cdot\prod_{i=1}^{d_2} P(G_i^2,f_i)}{\partial P(G_k^1,e_k)} \\ =& \prod_{i=1,i\neq k}^{d_1} P(G_i^1,e_i)\cdot\prod_{i=1}^{d_2} P(G_i^2,f_i) \\ \leq& (\frac{1}{2})^{d_1+d_2-1}\end{aligned}$$

Similarly for $\frac{\partial \mathbb{P}(\alpha=0,\beta=0)}{\partial P(G_i^2,f_i)}$. □

Corollary 9.

$$\sum_k \frac{\partial \mathbb{P}\left(\alpha=0,\beta=0\right)}{\partial P(G_k^1,e_k)} + \sum_k \frac{\partial \mathbb{P}\left(\alpha=0,\beta=0\right)}{\partial P(G_k^2,f_k)} \leq (d_1+d_2)(\frac{1}{2})^{d_1+d_2-1} \leq 1$$

Proposition 10.

$$\frac{\partial \mathbb{P}\left(\alpha=0\right)}{\partial P(G_i^1, e_i)} \le (\frac{1}{2})^{d_1-1}$$
$$\frac{\partial \mathbb{P}\left(\beta=0\right)}{\partial P(G_i^2, f_i)} \le (\frac{1}{2})^{d_2-1}$$

Corollary 11.

$$\sum_i \frac{\partial \mathbb{P}\left(\alpha=0\right)}{\partial P(G_i^1, e_i)} + \sum_i \frac{\partial \mathbb{P}\left(\beta=0\right)}{\partial P(G_i^2, f_i)} \le d_1 \left(\frac{1}{2}\right)^{d_1-1} + d_2 \left(\frac{1}{2}\right)^{d_2-1} \le 2$$

4.3 Computationally Efficient Correlation Decay

TODO: define the recursion depth of an edge.

Definition 12. *Let T be a rooted tree, $M \ge 2$ be a constant, the M-based depth $L_M(e)$ of a edge e in T is defined recursively as follows: $L_M(e) = 0$ if ???*

5 Computation Tree and an FPTAS Algorithm

Algorithm 2: Estimate $P(G,e)$ using $B_M^*(L)$

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

input : Graph G; edge e; boundary configuration C; M-based depth L;
output: Estimate of $P(G,e)$ from vertices in $B_M^*(L)$ using boundary configuration C.
begin
  if *G is trivial dangling graphs* **then**
    return $C \upharpoonright e$
  else if *G is non-dangling graphs* **then**
    $X \leftarrow \prod_{i=1}^{d_1} P(G_i^1, e_i, C, L)$;
    $Y \leftarrow \prod_{i=1}^{d_2} P(G_i^2, f_i, C, L)$;
    $Z \leftarrow \prod_{i=1}^{d_2} P(G_i^3, f_i, C, L)$;
    return $1 - \dfrac{1}{2 + X \cdot Y - X - Z}$;
  else if *e is free* **then**
    return $\frac{1}{2}$;
  else // e is dangling and G is nontrivial dangling
    $L' \leftarrow L - \lceil \log_M (d+1) \rceil$;
    return $\frac{1-\prod_{i=1}^{d} P(G_i,e_i,C,L')}{2-\prod_{i=1}^{d} P(G_i,e_i,C,L')}$;