A Simple FPTAS for Counting Edge Covers

Chengyu Lin[1] Jingcheng Liu[1] Pinyan Lu[2]

[1]Shanghai Jiao Tong University

[2]Microsoft Research Asia

ACM-SIAM Symposium on Discrete Algorithms, 2014

Overview

Edge cover

Definition

For an undirected input graph $G = (V, E)$, an **edge cover** of G is a set of edges C covering all vertices.

Example

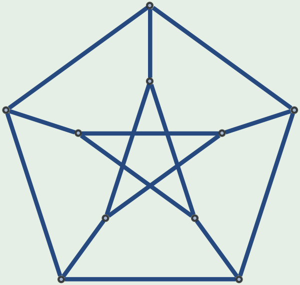


Figure : An edge cover for Petersen graph

Edge cover

Definition

For an undirected input graph $G = (V, E)$, an **edge cover** of G is a set of edges C covering all vertices.

Example

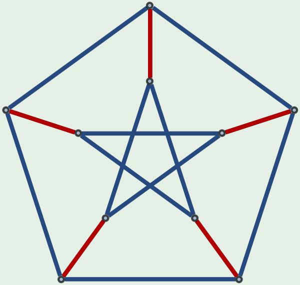


Figure : An edge cover for Petersen graph, with edges chosen being highlighted in red. Note that this is also a perfect matching.

Edge cover

Edge cover is related to many other problems such as:

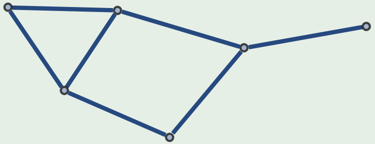
- Matching problem.
- Rtw-Mon-CNF. (read twice monotone CNF)
- Holant problem.
- ....

Relation to Matching

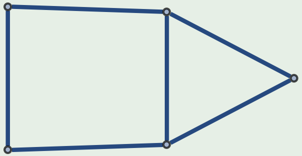
The minimal edge cover could be found via a greedy algorithm based on a maximum matching.

Example

Find edge covers by maximal matching?



(a) G has a perfect matching.



(b) G doesn't have a perfect matching.

Remark

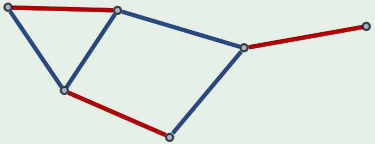
For a graph with a perfect matching, enumerating (sampling) perfect matchings is equivalent to enumerating (sampling) minimum edge covers.

Relation to Matching

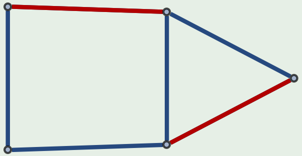
The minimal edge cover could be found via a greedy algorithm based on a maximum matching.

Example

Find edge covers by maximal matching?



(a) G has a perfect matching.



(b) G doesn't have a perfect matching.

Remark

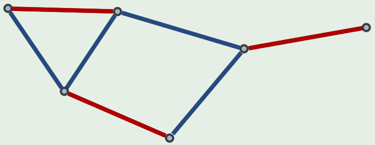
For a graph with a perfect matching, enumerating (sampling) perfect matchings is equivalent to enumerating (sampling) minimum edge covers.

Relation to Matching

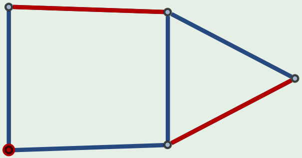
The minimal edge cover could be found via a greedy algorithm based on a maximum matching.

Example

Find edge covers by maximal matching?



(a) G has a perfect matching.



(b) G doesn't have a perfect matching.

Remark

For a graph with a perfect matching, enumerating (sampling) perfect matchings is equivalent to enumerating (sampling) minimum edge covers.

Relation to Rtw-Mon-CNF

TBA.

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- ....

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- ....

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- ....

A list of problems in their search, optimization, and counting versions.

Search problems:	**Optimizations:**	**Counting problems:**
SAT.	MAX-SAT.	#SAT.
Find a (perfect) matching.	Find a maximum matching.	Counting matchings.
Find an edge cover.	Find a minimum edge cover.	Counting edge covers.
. . . .	. . . .	. . . .

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- ....

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- ....

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- ....

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- . . . .

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- . . . .

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- . . . .

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- ....

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- ....

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- ....

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- . . . .

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- . . . .

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- . . . .

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- . . . .

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- . . . .

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- . . . .

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- ....

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- ....

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- ....

Counting Problems

A list of problems in their search, optimization, and counting versions.

Search problems:

- SAT.
- Find a (perfect) matching.
- Find an edge cover.
- . . . .

Optimizations:

- MAX-SAT.
- Find a maximum matching.
- Find a minimum edge cover.
- . . . .

Counting problems:

- #SAT.
- Counting matchings.
- Counting edge covers.
- . . . .

Why counting?

Besides theoretical computer science, counting problems are also related to many problems from other discipline such as:

- Partition function of Statistical physics.
- Graph polynomials.
- Sampling, learning and inference.
- Query evaluations of probabilistic database.
- . . . .

Why counting?

Besides theoretical computer science, counting problems are also related to many problems from other discipline such as:

- Partition function of Statistical physics.
- Graph polynomials.
- Sampling, learning and inference.
- Query evaluations of probabilistic database.
- ....

Why counting?

Besides theoretical computer science, counting problems are also related to many problems from other discipline such as:

- Partition function of Statistical physics.
- Graph polynomials.
- Sampling, learning and inference.
- Query evaluations of probabilistic database.
- . . . .

Why counting?

Besides theoretical computer science, counting problems are also related to many problems from other discipline such as:

- Partition function of Statistical physics.
- Graph polynomials.
- Sampling, learning and inference.
- Query evaluations of probabilistic database.
- ....

Approximate Counting

Many interesting problems in the exact counting regimes, including counting edge cover, is hard (#P-complete). Instead we look for these two types of polynomial time approximation scheme:

Definition (FPTAS)

For given parameter $\varepsilon > 0$ and an instance of a particular problem class, if the algorithm outputs a number $\hat{N}$ such that $(1-\varepsilon)N \leq \hat{N} \leq (1+\varepsilon)N$, where N is the accurate answer of the problem instance, and the running time is bounded by $poly(n, 1/\varepsilon)$ with n being the size of instance, this is called the **FPTAS (fully polynomial time approximation scheme)**.

Definition (FPRAS)

A randomized relaxation of FPTAS is known as **FPRAS (fully polynomial time randomized approximation scheme)**, which uses random bits and only outputs $\hat{N}$ to the desired precision with high probability.

Approximate Counting

Many interesting problems in the exact counting regimes, including counting edge cover, is hard (#P-complete). Instead we look for these two types of polynomial time approximation scheme:

Definition (FPTAS)

For given parameter $\varepsilon > 0$ and an instance of a particular problem class, if the algorithm outputs a number $\hat{N}$ such that $(1-\varepsilon)N \leq \hat{N} \leq (1+\varepsilon)N$, where N is the accurate answer of the problem instance, and the running time is bounded by $poly(n, 1/\varepsilon)$ with n being the size of instance, this is called the **FPTAS (fully polynomial time approximation scheme)**.

Definition (FPRAS)

A randomized relaxation of FPTAS is known as **FPRAS (fully polynomial time randomized approximation scheme)**, which uses random bits and only outputs $\hat{N}$ to the desired precision with high probability.

Approximate Counting

Many interesting problems in the exact counting regimes, including counting edge cover, is hard (#P-complete). Instead we look for these two types of polynomial time approximation scheme:

Definition (FPTAS)

For given parameter $\varepsilon > 0$ and an instance of a particular problem class, if the algorithm outputs a number $\hat{N}$ such that $(1-\varepsilon)N \leq \hat{N} \leq (1+\varepsilon)N$, where N is the accurate answer of the problem instance, and the running time is bounded by $poly(n, 1/\varepsilon)$ with n being the size of instance, this is called the **FPTAS (fully polynomial time approximation scheme)**.

Definition (FPRAS)

A randomized relaxation of FPTAS is known as **FPRAS (fully polynomial time randomized approximation scheme)**, which uses random bits and only outputs $\hat{N}$ to the desired precision with high probability.

Main Result

Previous work:

- Only an MCMC-based FPRAS is known for counting edge covers in graphs with maximum degree 3.
- The correlation decay based FPTAS for anti-ferromagnetic 2-spins systems (e.g. counting independent sets) goes beyond the best known MCMC based FPRAS and achieves the boundary of approximability.

Our main result is a correlation based FPTAS for counting edge covers for general graphs, which provides another example where the tractable range for correlation decay based FPTAS exceeds the sampling based FPRAS.

Main Result

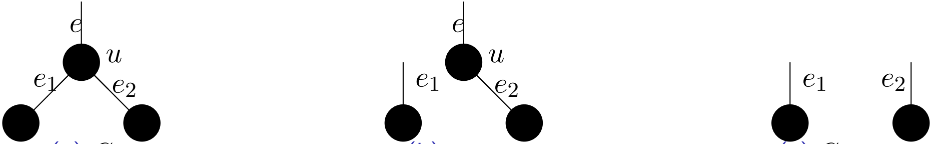
Previous work:

- Only an MCMC-based FPRAS is known for counting edge covers in graphs with maximum degree 3.
- The correlation decay based FPTAS for anti-ferromagnetic 2-spins systems (e.g. counting independent sets) goes beyond the best known MCMC based FPRAS and achieves the boundary of approximability.

Our main result is a correlation based FPTAS for counting edge covers for general graphs, which provides another example where the tractable range for correlation decay based FPTAS exceeds the sampling based FPRAS.

Definition

A **dangling edge** $e = (u, \_)$ of a graph is such singleton edge with exactly one end-point vertex u, as shown in the Figure 3a.



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

Figure : Dangling edges examples.

Counting v.s. Marginal Probability

Recall our counting problem:

Problem

Let $EC(G)$ be the set of edge covers.
Goal: estimate $|EC(G)|$.

Let X be an edge cover sampled uniformly from $EC(G)$, consider the following marginal probability:
for an edge e, we write $P(G,e) \triangleq \Pr(e \notin X)$.

Solution: estimate $P(G,e)$.

Counting v.s. Marginal Probability

Recall our counting problem:

Problem

Let $EC(G)$ be the set of edge covers.
Goal: estimate $|EC(G)|$.

Let X be an edge cover sampled uniformly from $EC(G)$, consider the following marginal probability:
for an edge e, we write $P(G, e) \triangleq \Pr(e \notin X)$.

Solution: estimate $P(G, e)$.

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.
Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)) .$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.

Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)) \,.$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

$$\Pr(X = E) = \Pr(\forall i, e_i \in X)$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.

Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)) .$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

$$\begin{aligned}\Pr(X = E) &= \Pr(\forall i, e_i \in X) \\ &= \Pr(e_1 \in X) \Pr(e_2 \in X \mid e_1 \in X) \cdots\end{aligned}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.

Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)) .$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

$$\begin{aligned} \Pr(X = E) &= \Pr(\forall i, e_i \in X) \\ &= \Pr(e_1 \in X) \Pr(e_2 \in X \mid e_1 \in X) \cdots \\ &= \prod_i \Pr(e_i \in X \mid \{e_j\}_{j=1}^{i-1} \subseteq X) \end{aligned}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.

Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)) .$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

$$\begin{aligned}\Pr(X = E) &= \Pr(\forall i, e_i \in X) \\ &= \Pr(e_1 \in X) \Pr(e_2 \in X \mid e_1 \in X) \cdots \\ &= \prod_i \Pr(e_i \in X \mid \{e_j\}_{j=1}^{i-1} \subseteq X) \\ &= \prod_i (1 - P(G_i, e_i)),\end{aligned}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.

Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)).$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

$$\begin{aligned}
\Pr(X = E) &= \Pr(\forall i, e_i \in X) \\
&= \Pr(e_1 \in X) \Pr(e_2 \in X \mid e_1 \in X) \cdots \\
&= \prod_i \Pr(e_i \in X \mid \{e_j\}_{j=1}^{i-1} \subseteq X) \\
&= \prod_i (1 - P(G_i, e_i)),
\end{aligned}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.

Therefore,

$$\frac{1}{|EC(G)|} = \prod_i (1 - P(G_i, e_i)).$$

Counting from marginal probability

Recall that the set of all edges E is an edge cover. For a randomly sampled edge cover X, what is $\Pr(X = E)$?

$$\Pr(X = E) = \frac{1}{|EC(G)|}$$

$$\begin{aligned}
\Pr(X = E) &= \Pr(\forall i, e_i \in X) \\
&= \Pr(e_1 \in X) \Pr(e_2 \in X \mid e_1 \in X) \cdots \\
&= \prod_i \Pr(e_i \in X \mid \{e_j\}_{j=1}^{i-1} \subseteq X) \\
&= \prod_i \left(1 - P(G_i, e_i)\right),
\end{aligned}$$

where $G_1 = G, G_{i+1} = G_i - e_i - u_i - v_i$.
Therefore,

$$\frac{1}{|EC(G)|} = \prod_i \left(1 - P(G_i, e_i)\right).$$

Estimating Marginal Probability

We focus on e is dangling.

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

where $G_1 \triangleq G - e - u$, and $G_{i+1} \triangleq G_i - e_i$.

Truncate with modified recursion depth:

$$P(G,e,L) = \frac{1 - \prod_{i=1}^{d} P(G_i, e_i, L - \lceil \log_6 (d+1) \rceil)}{2 - \prod_{i=1}^{d} P(G_i, e_i, L - \lceil \log_6 (d+1) \rceil)},$$

Estimating Marginal Probability

We focus on e is dangling.

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

where $G_1 \triangleq G - e - u$, and $G_{i+1} \triangleq G_i - e_i$.
Truncate with modified recursion depth:

$$P(G,e,L) = \frac{1 - \prod_{i=1}^{d} P(G_i, e_i, L - \lceil \log_6 (d+1) \rceil)}{2 - \prod_{i=1}^{d} P(G_i, e_i, L - \lceil \log_6 (d+1) \rceil)},$$

Correlation Decay

Proposition

Given graph G, edge e and depth L,

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

Thank you!