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

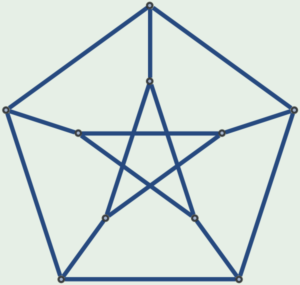


Figure : An edge cover for Petersen graph

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

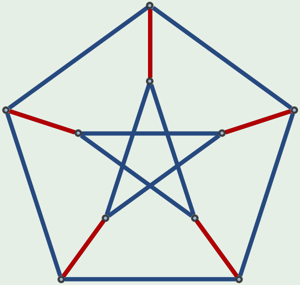


Figure : An edge cover for Petersen graph, with edges chosen being highlighted in red. Note that this is also a perfect matching.

Edge cover

Edge cover is related to many other problems such as:

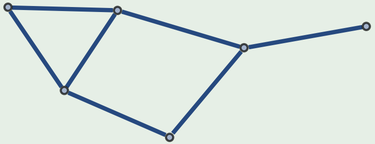
- Matching problem.
- Rtw-Mon-CNF. (read twice monotone CNF)
- Holant problem.
- . . . .

Relation to Matching

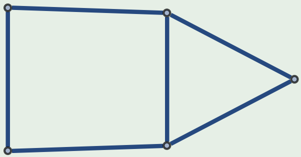
The minimal edge cover could be found via a greedy algorithm based on a maximum matching.

Example

Find edge covers by maximal matching?



(a) G has a perfect matching.



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

Remark

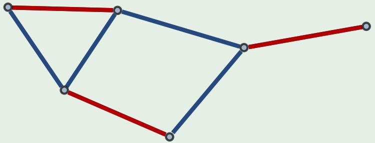
For a graph with a perfect matching, enumerating (sampling) perfect matchings is equivalent to enumerating (sampling) minimum edge covers.

Relation to Matching

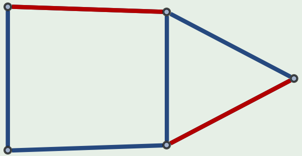
The minimal edge cover could be found via a greedy algorithm based on a maximum matching.

Example

Find edge covers by maximal matching?



(a) G has a perfect matching.



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

Remark

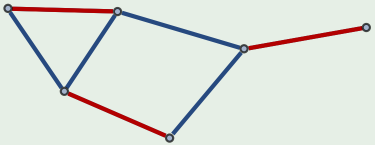
For a graph with a perfect matching, enumerating (sampling) perfect matchings is equivalent to enumerating (sampling) minimum edge covers.

Relation to Matching

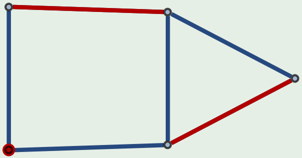
The minimal edge cover could be found via a greedy algorithm based on a maximum matching.

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.
- . . . .

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.
- . . . .

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.

Counting v.s. Marginal Probability

TBA.

¡++¿