5-2016

Sampling and Counting Crossing-Free Matchings

Wenbo Sun
ws3109@rit.edu

Sampling and Counting Crossing-Free Matchings

by

Wenbo Sun

THESIS

Presented to the Department of Computer Science
of the Golisano College of Computing and Information Sciences
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Computer Science

Rochester Institute of Technology

May 2016

Sampling and Counting Crossing-Free Matchings

APPROVED BY

SUPERVISING COMMITTEE:

Dr. Ivona Bezáková, Advisor — Date
Department of Computer Science
Rochester Institute of Technology

Dr. Stanisław Paweł Radziszowski, Reader — Date
Department of Computer Science
Rochester Institute of Technology

Dr. Christopher Homan, Observer — Date
Department of Computer Science
Rochester Institute of Technology

Sampling and Counting Crossing-Free Matchings

Abstract

Sampling of combinatorial structures is an important statistical tool used in applications in a number of areas ranging from statistical physics, data mining, to biological sciences. Of comparable importance is the computation of the corresponding partition function, which, in the case of the uniform distribution, is equivalent to the problem of counting all such structures. For self-reducible combinatorial structures, once we can produce an almost uniform sample from them, then we can approximately count them.

Using a Markov chain Monte Carlo method, this thesis presents polynomial-time algorithms to approximately count and almost uniformly sample crossing-free matchings for certain input classes of graphs. Since the problem in its generality appears to be difficult, we made natural restrictions on the input graphs. Namely, we consider vertices arranged in a grid in the plane, where edges are line segments connecting the vertices and a matching is crossing-free if no two matching edges intersect. For appropriate bounds on the dimensions of the grid and the edge lengths, we show that a natural Markov chain is rapidly mixing and that the problem is self-reducible.

Table of Contents

List of Figures

Chapter 1

Introduction

In this section we will introduce the definitions of a matching, a perfect matching and a crossing-free matching, as well as a related application in statistical physics.

1.1 Problem description

Given a graph $G = (V, E)$, a *matching* M is a subgraph of G, such that every vertex in M has degree at most one. We say that a matching is *perfect* if every vertex has degree exactly one. In this thesis we will work with geometric graphs where vertices are given as points in the plane and every pair of vertices is connected by the line segment between the corresponding points, provided it does not pass through any other point. We say that a matching in a geometric graph is *crossing-free* if no two matching edges intersect in the plane. The main focus of this thesis is to design algorithms to count crossing-free matchings.

1.2 Motivation and related work

Counting of matchings has arisen in numerous applications. One of the applications in statistical physics is called the monomer-dimer model. Given an undirected graph $G = (V, E)$ and a matching $M \subseteq E$, every edge of M is called a "dimer"; every vertex that is not covered by M is called a "monomer". A matching M in G is also called a "monomer-dimer" cover of G. Clearly, if M is a perfect matching, then there are only dimers.

In chemistry, fullerene is a molecule consisting entirely of carbon atoms. Each carbon is three-connected to other carbon atoms by one double bond and two single bonds. We define a "fullerene graph" [KKMS09] as a 3-regular 3-connected planar graph with pentagon or hexagon faces. Given two fullerene graphs G_1 and G_2 (G_1 and G_2 have same number of vertices), we want to know which one is more stable. It has been discovered that if $P_m(G_1) \geq P_m(G_2)$, then G_1 is more stable than G_2, where $P_m(G)$ is the number of perfect matchings in graph G. Therefore, designing an efficient algorithm to count perfect matchings is necessary.

This initial motivation for designing algorithms that count perfect or all matchings, has led to numerous studies of these problems under a variety of restrictions. For general graphs, the problems are known to be #P-complete [Val79a, Val79b] but a *fully polynomial-time randomized approximation scheme* (so called *fpras*) exists for some types of input, most notably for perfect matchings in planar graphs [Kas63] and bipartite graphs [JSV04] and for all matchings (with no restrictions) in general graphs [JS89]. Com-

putational geometry community has studied matchings under geometric constraints [Wet14, SW06], leading to a natural question of designing counting algorithms for matchings under such constraints. For counting of crossing-free matchings, Micha Sharir and Emo Welzl [SW06] showed that a set of n points has at most $O(10.43^n)$ crossing-free matchings. Manuel Wettstein [Wet14] also showed that the number of crossing-free perfect matchings can be computed in time $O(2^n n^4)$.

Instead of using a randomized scheme, Mohsen Bayati and David Gamarnik [BGK+07] constructed a *deterministic fully polynomial-time approximation scheme* (*fptas*) for computing the total number of matchings. Their algorithm is able to efficiently (in polynomial-time) count the number of matchings in bounded degree graphs, and in sub-exponential time for general graphs (the complexity is $\exp(O(\sqrt{n}\log_2 n))$, where n is the number of vertices).

Markov chain community has also studied other structures with geometric constraints, such as the recent results on sampling triangulations of lattice points [CMSS15,CMSS13]. We are not aware of any Markov chain work on crossing-free matchings.

Chapter 2

Related concepts

Counting of matchings is a #P-complete problem, which is at least as hard as NP-complete problems. Although hardness of counting matchings under geometric restrictions is unclear, instead of trying to find a polynomial-time algorithm for exact counting, we attempt to bypass the hardness result by counting approximately. We present the relationship between *approximate counting* and *almost-uniform sampling* (approximate means the ratio of the output and the correct answer is arbitrarily close to 1, where the running time depends on the approximation ratio). In Section 2.3, we give a pseudocode of the algorithm reducing approximate counting to almost-uniform sampling.

2.1 The class #P

Valiant [Val79a, Val79b] introduced the class #P of counting problems. A class of the hardest problems in #P, the so-called #P-*complete problems*, are the problems in #P that any problem in #P can be reduced to by a polynomial-time counting reduction. Actually, many counting problems are #P-complete, and, unless P = NP, we do not have any polynomial-time algorithm for them.

2.2 Connection between counting and sampling

Fortunately, the existence of #P-complete problems does not rule out the possibility of designing polynomial-time approximation algorithms for them. Jerrum, Valiant, and Vazirani [JVV86] show the intimate relation between "almost uniform" sampling and approximate counting. We first introduce some important definitions.

Definition 2.2.1. *Let f be a function in* #P*. A* fully polynomial randomized approximation scheme (fpras) *for f is a randomized algorithm that on input (x, ϵ), where $x \in \Sigma^*$ and $\epsilon > 0$, outputs a random variable N such that*

$$\Pr[e^{-\epsilon} f(x) \leq N \leq e^{\epsilon} f(x)] \geq \frac{3}{4}, \tag{2.1}$$

and runs in time polynomial in x and ϵ^{-1}.

Notice that N converges to $f(x)$ as ϵ converges to 0. Actually, the number $\frac{3}{4}$ can be replaced by any number between in the open interval $\left(\frac{3}{4}, 1\right)$ (see Claim 2.2.1).

Claim 2.2.1 ([Sin09])**.** *If there exists an* fpras *for f, we can boost the confidence from $\frac{3}{4}$ to $1 - \delta$ at the cost of a slowdown by a factor of $O(\log \delta^{-1})$.*

Before we introduce the definition of the *almost uniform sampler*, we first introduce the definition of the *total variation distance*, which used to measure the closeness of two probability distributions.

Definition 2.2.2. *For two probability distribution μ and η on Ω, the total variation distance is defined as*

$$d_{\mathrm{TVD}}(\mu, \eta) = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \eta(x)| = \max_{A \subseteq \Omega} |\mu(A) - \eta(A)|, \tag{2.2}$$

where $\mu(A) = \sum_{a \in A} \mu(a)$.

Figure 2.1 shows the total variation distance between two Gamma distributions.

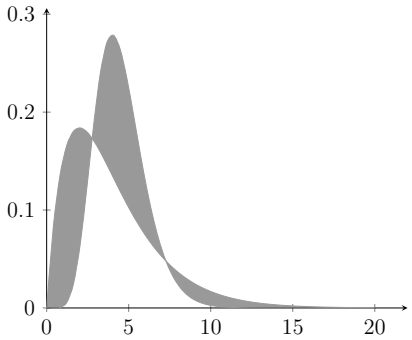


Figure 2.1: The total variation distance of the above two Gamma distributions is the half of the area of the gray region.

Definition 2.2.3 ([Jer03])**.** *A sampling problem is specified by a relation $S \subseteq \Sigma^* \times \Sigma^*$ between problem instances x and "solution" $w \in S(x)$. An* almost uniform sampler *for a solution set $S \subseteq \Sigma^* \times \Sigma^*$ is a randomized algorithm that takes as input an instance $x \in \Sigma^*$ and a sampling tolerance $\delta > 0$, outputs a solution $W \in S(x)$ such that the total variation distance between the distribution of W and the uniform distribution on $S(x)$ is at most δ. The sampler is* fully polynomial *if it runs in time bounded by a polynomial in x and $\log \delta^{-1}$. The* fully polynomial almost-uniform sampler *is also referred to as an* fpaus.

Theorem 2.2.1 reveals the connection between *fpras* and *fpaus* for so-called *self-reducible* problems. We will not define the concept of *self-reducibility* in this text. Instead of going through a formal definition, we will demonstrate the concept in Section 2.3, Algorithm 1, line 4. Intuitively, we use sampling to estimate the ratio of the size of the original set (this is the count we are interested in) and the size of a smaller set that can be described in the same manner as the original set. Then, we continue the process recursively, obtaining a sequence of ratios. The product of these ratios yields the wanted count.

Theorem 2.2.1 ([FF93]). *For all* self-reducible #P *problems, there exists an* fpras *for counting if and only if there exists a polynomial-time algorithm for (almost) uniform sampling.*

There is also an important conjecture for a specific subset of #P-complete problems:

Conjecture 2.2.1 ([Sin09]). *If a problem A is* NP*-complete, then its counting version #A is* #P*-complete.*

2.3 Reducing approximate counting to sampling

Algorithm 1 shows how to reduce [Jer03] approximate counting to uniform sampling.

Algorithm 1: Reducing approximate counting to uniform sampling

input : A finite combinatorial set Ω_i described implicitly by a small input

output: An estimate of $|\Omega_i|$

if $|\Omega_i|$ *can be computed directly* **then**
  `/* For example, once` Ω_i `is small enough, we can directly get the value of` $|\Omega_i|$ `by enumeration. */`
  return $|\Omega_i|$
else
  Partition the set Ω_i into two subsets, Ω_{i+1} and $\Omega_i - \Omega_{i+1}$, according to some properties so that Ω_{i+1} can be also described implicitly by a small input
  Sample sufficiently many times from Ω_i according to the uniform distribution to estimate the ratio p_{i+1}, where $p_{i+1} = |\Omega_{i+1}|/|\Omega_i|$
  Recursively estimate $|\Omega_{i+1}|$, then let $|\Omega_{i+1}|_{estimate}$ denote the return value
  return $|\Omega_{i+1}|_{estimate}/p_{i+1}$
end

To see what is happening more clearly, let Ω_i denote the set we get after the i^{th} partition (notice that $\Omega_0 = \Omega$, and after the i^{th} partition, we have two subsets Ω_i and $\Omega_{i-1} - \Omega_i$). Let $p_i = |\Omega_i|/|\Omega_{i-1}|$ and assume that we only need n partitions, then we have

$$\frac{1}{|\Omega|} = \frac{1}{|\Omega_n|} \prod_{i=1}^{n} \frac{|\Omega_i|}{|\Omega_{i-1}|} \tag{2.3}$$

$$= \frac{1}{|\Omega_n|} \prod_{i=1}^{n} p_i. \tag{2.4}$$

Thus, $|\Omega|$ can be estimated by

$$|\Omega| = |\Omega_n| \left(\prod_{i=1}^{n} p_i \right)^{-1}. \tag{2.5}$$

The equation (2.5) gives us an insight that once we could estimate p_i accurately, we may be able to get a good estimate of $|\Omega|$. The main problem is that, for any given set Ω_i, how to find a good split into Ω_i and $\Omega_i - \Omega_{i+1}$, such that Ω_{i+1} can be further split in the same manner.

Furthermore, notice that if $|\Omega_i|$ is exponentially larger than $|\Omega_{i+1}|$, we then have to sample from Ω exponentially many times in order to get a sample from Ω_{i+1}, which means our algorithm also needs exponential time. Let us see why would this happen. Assume we have a finite set Ω_i, and we split it into Ω_{i+1} and $\Omega_i - \Omega_{i+1}$, such that $|\Omega_i| = \exp(|\Omega_{i+1}|)$ ($\exp(|\Omega_{i+1}|)$ is a number that is exponentially larger than $|\Omega_{i+1}|$). We keep sampling over Ω_i. Let X be the number of trials required until we get one sample in Ω_{i+1}; let p denote the probability that a sample is in Ω_{i+1}, then $p = 1/\exp(|\Omega_{i+1}|)$. Then $\mathrm{E}[X]$ can be computed by the following procedures:

$$\mathrm{E}[X] = \mathrm{E}[X|X=1] \cdot \Pr(X=1) + \mathrm{E}[X|X>1] \cdot \Pr(X>1) \tag{2.6}$$

It is easy to find out that

$$\mathrm{E}[X|X=1] = 1, \tag{2.7}$$

$$\Pr(X=1) = p, \tag{2.8}$$

$$\mathrm{E}[X|X>1] = 1 + \mathrm{E}[X], \tag{2.9}$$

$$\Pr(X>1) = 1 - p. \tag{2.10}$$

According to the above equations, we have

$$\mathrm{E}[X] = \frac{1}{p} = \exp(|\Omega_{i+1}|). \tag{2.11}$$

Therefore, to build an efficient algorithm, it is necessary to bound $|\Omega_{i+1}|/|\Omega_i|$ by a constant or a polynomial of the input size.

Chapter 3

Introduction to Markov chains

The Markov chain approach is a powerful general method for obtaining approximation algorithms for hard combinatorial enumeration problems. We can usually design a Markov chain which can be used to sample over a finite state space according to some specific distributions. In this chapter, we first introduce the basic notions of the theory of Markov chains; second, we introduce the concept of Metropolis processes, which provide us with a recipe for constructing a Markov chain as a sampler that can sample over the state space according to the uniform distribution; and finally, we describe a technique named "canonical paths" that can be used to bound the "mixing time" of Markov chains.

3.1 Markov chains

This section is devoted to the definition and some important properties of a Markov chain which will be used throughout the thesis. In this thesis we only deal with discrete-time Markov chains on a finite state space Ω.

Definition 3.1.1. *A* Markov chain (P, Ω) *is a pair* (Ω, P)*, where* Ω *is the* state space *and* P *is the* transition matrix*. The matrix* P *is* $|\Omega| \times |\Omega|$ *where*

each entry $P(x, y)$ denotes the probability of moving from x to y in a single step. Therefore, $P(x, y) \geq 0$ for all x, y in Ω and $\sum_y P(x, y) = 1$ for all x in Ω.

Figure 3.1 shows a four-state Markov chain:

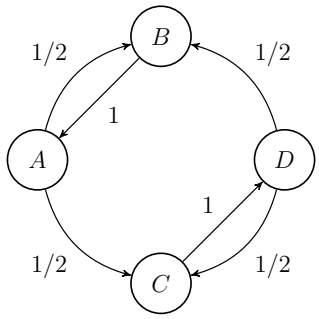


Figure 3.1: A Markov chain.

The transition matrix P describes one-step transition probabilities; the t-step transition probabilities are given by

$$P^t(x, y) = \begin{cases} I(x, y) & \text{if } t = 0, \\ P(x, y) & \text{if } t = 1, \\ \sum_{z \in \Omega} P^{t-1}(x, z) P(z, y) & \text{if } t > 1. \end{cases} \tag{3.1}$$

Notice that $P^t(x, y) = \Pr(X_t = y | X_0 = x)$ and that $P^t = \prod_{i=1}^{t} P$. In order to simplify our notation, instead of using (P, Ω), we will refer to a Markov chain by P.

Definition 3.1.2. *A* stationary distribution *of a Markov chain with transition matrix P is a probability distribution $\pi : \Omega \to [0, 1]$ satisfying*

$$\pi(y) = \sum_{x \in \Omega} \pi(x) P(x, y). \tag{3.2}$$

If we treat π as a vector, (3.2) *can also be written as*

$$\pi = \pi P. \tag{3.3}$$

The stationary distribution reveals the long run proportion of time that the Markov chain was in any state. For example, assume that we have a Markov chain with n states ($n > 0$), that converges to its unique stationary distribution, that is the uniform distribution. Then we run the Markov chain for t steps. As t goes to infinity, no matter what the initial state is, the proportion of time spent in each state will converge to $\frac{1}{n}$.

Definition 3.1.3. *A Markov chain P is* irreducible *if for all x, y, there exists some t such that $P^t(x, y) > 0$. It is* aperiodic *if for all x, y we have* $\gcd\{t : P^t(x, y) > 0\} = 1$. *A Markov chain is* ergodic *if it is both irreducible and aperiodic.*

Theorem 3.1.1 states a necessary and sufficient condition for convergence of a Markov chain to its stationary distribution regardless of the initial state.

Theorem 3.1.1. *If a Markov chain P is ergodic then it has a unique stationary distribution π, and for every distribution μ on Ω, we have* $\lim_{t\to\infty} \mu P^t(x, y) = \pi(y)$.

Clearly, the above theorem says that, given an ergodic Markov chain, for all x, y in Ω, we have $\lim_{t\to\infty} \Pr(X_t = y | X_0 = x) = \pi(y)$.

We now show what will happen if a Markov chain is reducible or periodic.

If a Markov chain is reducible, then it may have more than one stationary distribution. Look at Figure 3.2, if the initial state is B, then the Markov chain will stay in B forever, the stationary distribution is $[0, 1, 0, 0]$; if the initial state is C, the stationary distribution is always $[0, 0, 1, 0]$.

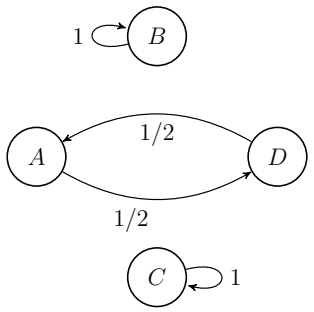


Figure 3.2: A reducible Markov chain.

If a Markov chain is periodic, then even if it has a unique stationary distribution, it may never converge to it. Look at Figure 3.3, after we run the Markov chain for n times, as n goes to infinity, no matter what the initial state is, the proportion of times we visit each state will converge to $[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$, which is the stationary distribution of the Markov chain. But the Markov chain will never converge, the distribution over the state space will always among $\{[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]\}$.

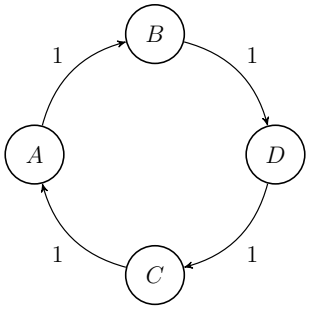


Figure 3.3: A periodic Markov chain.

Definition 3.1.4 ([Sin09])**.** *A* lazy *version of a Markov chain P is a new Markov chain Q defined as $Q = (P + I)/2$.*

Note that Q stays in the same state with probability $\frac{1}{2}$, otherwise it follows the moves of P. The lazy version of the Markov chain shown in Figure 3.3 is shown in Figure 3.4.

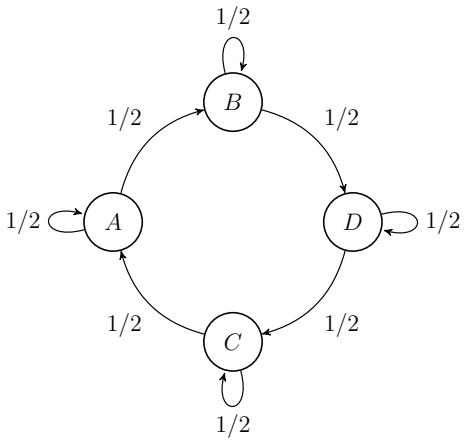


Figure 3.4: The lazy version of the Markov chain in Figure 3.3.

Lemma 3.1.2. *Let P be an irreducible Markov, then its lazy version Q is both irreducible and aperiodic, and it has the same stationary distribution as P.*

Because of Lemma 3.1.2, we usually do not worry about the periodicity, since no matter what the original Markov chain is, instead of simulating the original one, we can simulate its lazy version.

We have known that an ergodic Markov chain has a unique stationary distribution π, and after we run it for sufficiently many steps, it will eventually converge to π, regardless of the initial state. Then we have to find answers to the following to problems:

1. How to construct a Markov chain whose stationary distribution is π over Ω?

2. How many steps should we run the Markov chain until it converges to π?

In Section 3.2 we will introduce the *Metropolis process*, which is a recipe to construct a *reversible* Markov chain (see Definition 3.1.5) to sample over Ω. We will prove by Claim 3.1.1 that its stationary distribution is exactly the stationary distribution over Ω. In Section 3.3 we will introduce the concept of *mixing time* of the Markov chain, that is the minimum number of steps for the Markov chain to converge to its stationary distribution, as well as some techniques for bounding it.

Definition 3.1.5. *Let $\pi > 0$ be a probability distribution over Ω. A Markov chain P is said to be* reversible *with respect to π if for all x, y in Ω, $\pi(x)P(x,y) = \pi(y)P(y,x)$.*

Claim 3.1.1. *If a Markov chain P is reversible with respect to π, then π is a stationary distribution for P.*

Proof. By definition of matrix multiplication,

$$(\pi P)(x) = \sum_y \pi(y)P(y,x) = \sum_y \pi(x)P(x,y) = \pi(x). \tag{3.4}$$

□

3.2 The Metropolis process

Given a large but finite set Ω and a positive weight function $\omega : \Omega \to \mathbb{R}^+$, the Metropolis process is a scheme to design a Markov chain with a target stationary distribution $\pi(x) = \frac{\omega(x)}{Z}$, where $Z = \sum_{y\in\Omega} \omega(y)$. For a counting problem, for all $x \in \Omega$, $\omega(x) = 1$. In order to construct a Metropolis process, we need two ingredients.

The first one is the *neighborhood structure*. The main idea is that for any $x, y \in \Omega$, x and y are neighbors if and only if we can go from x to y by doing some "local changes", and vice versa. We denote by $x \sim y$ that x and y are neighbors (we assume that $x \sim y$ if and only if $y \sim x$). Furthermore, if we view the neighborhood structure as edges of a graph on the vertex set Ω, then this graph has to be connected.

The second one is the *proposal distribution*. For each $x \in \Omega$, the proposal distribution is a function $\kappa(x, \cdot)$. There are several properties of κ:

- $\forall x \in \Omega, \kappa(x, x) > 0$.

- $\forall x, y \in \Omega, \kappa(x, y) = \kappa(y, x)$
- $\sum_{y \in \text{neighbor}(x)} \kappa(x, y) = 1 - \kappa(x, x)$.

Then we can construct the transition matrix P by following these steps:

1. Given the current state x, for all $y \in \text{neighbor}(x)$, we select y with probability of $\kappa(x, y)$.
2. Make a transition to y with the probability of $\min\left\{1, \frac{\pi(y)}{\pi(x)}\right\} = \min\left\{1, \frac{\omega(y)}{\omega(x)}\right\}$.

Notice that x will transit to y with the probability $P(x, y) = \kappa(x, y) \cdot \min\left\{1, \frac{\omega(y)}{\omega(x)}\right\}$. We now prove that the stationary distribution of P is actually π.

Proof. Given $x, y \in \Omega$, and assume that $\omega(y) \leq \omega(x)$,

$$\begin{aligned}
\pi(x)P(x, y) &= \frac{\omega(x)}{Z}\kappa(x, y)\frac{\omega(y)}{\omega(x)} \\
&= \frac{\omega(y)}{Z}\kappa(x, y) \\
&= \frac{\omega(y)}{Z}\kappa(x, y) \cdot \min\left\{1, \frac{\omega(x)}{\omega(y)}\right\} \\
&= \pi(y)P(y, x).
\end{aligned}$$

Therefore, P is reversible and by claim (3.1.1), π is its stationary distribution. □

In this work, we are dealing with counting problems, and, for every element x in Ω, $\omega(x) = 1$. Then, $P(x, y) = \kappa(x, y)$ will always hold.

More details about the Metropolis process can be found in [Sin09].

3.3 Techniques for bounding the mixing time of a Markov chain

We first introduce the concept of *mixing time* of Markov chains, which is the minimum number of steps it takes before the distribution of states is within some total variation distance of the stationary distribution. The formal mathematical definition is as follows:

Definition 3.3.1. *Given an ergodic Markov chain P with stationary distribution π and a TVD ϵ, its mixing time is defined as*

$$\tau_x(\epsilon) = \min\{n : d_{\mathrm{TVD}}\left(P^n(x, \cdot), \pi\right) \leq \epsilon\}. \tag{3.5}$$

If a mixing time independent of the initial state is desired, then we use the following definition:

$$\tau(\epsilon) = \max_{x \in \Omega} \tau_x(\epsilon).$$

A Markov chain is said to be *rapidly mixing* if $\tau(\epsilon)$ is bounded by a polynomial in the size of the input and $\log(\epsilon^{-1})$.

To bound the mixing time, we will use a technique called *canonical paths*. Before we formally introduce it, we briefly review the drawbacks of other approaches. Coupling [Ald83] and stopping times [AD86] are techniques that are widely used for bounding mixing times of Markov chains. For *coupling*, Anil Kumar and Ramesh [KR01] studied a very similar Markov chain to the one we will use in this thesis, and demonstrated that every Markovian coupling for this chain on matchings takes expected exponential time to converge.

Stopping times have been used successfully in scenarios where coupling had been partially successful used (card shuffling, counting or sampling of independent sets, colorings [BDGJ99, BDK08, AD86, Vig01, DFJ02]), and the analysis of the mixing times is relatively complicated.

The basic concept of the canonical paths is to define paths between all pairs of states, and then argue that no edge is "overloaded". To describe the concept mathematically, we view the Markov chain P as an undirected graph (the Markov chain we use is reversible), whose vertex set is Ω and edge set is $E = \{(x, y) \in \Omega^2 \mid P(x, y) > 0\}$. We denote γ_{xy} as the canonical path from state x to y, which is a sequence of legal transitions from x to y. We usually define this path in some sort of canonical order to get from x to y, hence its name. Let

$$\Gamma = \{\gamma_{xy} | x, y \in \Omega\} \tag{3.6}$$

denote the set of all canonical paths. We now define ϱ as the measure of *congestion*:

$$\varrho(\Gamma) := \max_{t=(u,v)} \left\{ \frac{1}{\pi(u)P(u,v)} \sum_{x,y:\gamma_{xy} \text{ uses } t} \pi(x)\pi(y)|\gamma_{xy}| \right\}, \tag{3.7}$$

where $|\gamma_{xy}|$ is the length of the path γ_{xy}. The connection between congestion ϱ and mixing time τ is revealed by the following theorem.

Theorem 3.3.1. *Let P be a finite*[1]*, reversible, ergodic, lazy Markov chain. Let Γ be a set of canonical paths, ϱ be the congestion. Then the mixing time*

[1]Finite means that the state space Ω is finite.

of P satisfies

$$\tau_x(\epsilon) \leq 2\varrho(\ln \pi(x) + \ln \epsilon^{-1}), \tag{3.8}$$

where $\varrho = \varrho(\Gamma)$ is the congestion with respect to Γ.

The proof of the theorem can be found in [Sin92].

Chapter 4

Sampling and counting of crossing-free matchings

In this chapter, we design Markov chains for crossing-free matchings and investigate their mixing times. For general point configurations and no edge length restrictions, the problem appears to be very difficult. Therefore, we aim to prove rapid mixing for several natural restrictions such as points arranged in a grid setting, or restrictions on the lengths of the edges in the matchings.

4.1 Geometric graphs

In this section, we introduce the definition of a *geometric graph*, which will be the standard input of our sampling and counting algorithm.

Definition 4.1.1 (Geometric graphs)**.** *Geometric graphs are graphs where vertices are in the plane and edges are straight line segments connecting pairs of vertices, provided that they do not pass through other points. We denote such graph by $G = (V, E)$ where V is the point set and E is the edge set. If we allow all such straight line segments connecting all pairs of vertices, then we use the notation of $G = V$.*

4.2 Using sampling to count

In the remainder of this chapter, we will prove the following theorem.

Theorem 4.2.1. *If we have an* fpaus *for sampling crossing-free matchings in a graph G, with running time $\tau(n, \epsilon)$ (n is the number of vertices in G), then we have an* fpras *for counting all crossing-free matchings of the graph, with running time $O\left(\text{poly}(n, \epsilon) \cdot \tau(n, \epsilon)\right)$.*

For a given graph G, we consider a sequence of graphs $G_i := (V_i, E_i)$ for $1 \leq i \leq n$, where $V_i = \{v_1, v_2, \ldots, v_i\}$, and E_i is the set of edges that only connect vertices in V_i (notice that $G = G_n$). Let $|\Omega(G_i)|$ denote the number of crossing-free matchings in G_i. For convenience, make $|\Omega(G_0)| = 1$. The estimate of $|\Omega(G)|$ could be expressed as

$$|\Omega(G)| = (\rho_1 \rho_2 \ldots \rho_n)^{-1}, \tag{4.1}$$

where

$$\rho_i = \frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|}. \tag{4.2}$$

Therefore, in order to estimate $|\Omega(G)|$, it suffices to approximate each ρ_i. We will do this by using the Markov chain.

4.2.1 Bounding the number of samples

We first need to bound the ρ_i away from zero: ρ_i indicates the success probability of the Markov chain sampling procedure. In the counting problem for general matchings, Jerrum and Sinclair [JSV04] bound the above ratio by

a constant. However, in the case of crossing-free matchings, a constant lower bound appears hard. Fortunately, we can use a polynomial bound.

Notice that v_i can connect with any single vertex among $v_1, v_2, \ldots, v_{i-1}$. Let A_j denote the set of crossing-free matchings that contain the edge (v_i, v_j), where $1 \leq j \leq i-1$. Then we have

$$|\Omega(G_i) \backslash \Omega(G_{i-1})| = \sum_{j=1}^{i-1} |A_j| \tag{4.3}$$

Observe that A_j can be mapped injectively into $\Omega(G_{i-1})$ by removing the edge $e = (v_j, v_i)$, we know that $|A_j| \leq |\Omega(G_{i-1})|$. We can get a lower bound of the ratio by the following inequalities:

$$\begin{aligned} |\Omega(G_i)| - |\Omega(G_{i-1})| &\leq |\Omega(G_i) \backslash \Omega(G_{i-1})| \\ &= \sum_{j=1}^{i-1} |A_j| \\ &\leq (i-1)|\Omega(G_{i-1})|. \end{aligned} \tag{4.4}$$

Then from (4.4), we have

$$\frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|} \geq \frac{1}{i} \geq \frac{1}{n}. \tag{4.5}$$

We will need to compute how many samples do we need to get a good estimate of $|\Omega(G)|$, namely we want to compute a number N such that

$$e^{-\epsilon}|\Omega(G_n)| \leq N \leq e^{\epsilon}|\Omega(G_n)|, \tag{4.6}$$

where $0 < \epsilon \leq 1$ and $n > 0$ (see Definition 2.2.1). To estimate ρ_i (see (4.2)), we build an almost uniform sampler S with $\delta = \frac{\epsilon}{3n^2}$ that can sample a crossing-free matching M_i randomly on G_i. Let Z_i be an indicator random variable of

the event that $M_i \in \Omega(G_i)$. Let $\mu_i = \mathrm{E}[Z_i]$, then we have

$$\rho_i - \frac{\epsilon}{3n^2} \leq \mu_i \leq \rho_i + \frac{\epsilon}{3n^2}. \tag{4.7}$$

Since

$$\frac{1}{n} \leq \rho_i \leq 1, \tag{4.8}$$

we get the following result:

$$\begin{aligned} \rho_i - \frac{\epsilon}{3n^2} &= \rho_i - \frac{1}{n} \cdot \frac{\epsilon}{3n} \\ &\geq \rho_i - \rho_i \cdot \frac{\epsilon}{3n} \\ &= \rho_i(1 - \frac{\epsilon}{3n}); \end{aligned} \tag{4.9}$$

$$\begin{aligned} \rho_i + \frac{\epsilon}{3n^2} &= \rho_i + \frac{1}{n} \cdot \frac{\epsilon}{3n} \\ &\leq \rho_i + \rho_i \cdot \frac{\epsilon}{3n} \\ &= \rho_i(1 + \frac{\epsilon}{3n}). \end{aligned} \tag{4.10}$$

Combine (4.9), (4.10), and (4.7), we know that μ_i can be bounded by

$$\rho_i \left(1 - \frac{\epsilon}{3n}\right) \leq \mu_i \leq \rho_i \left(1 + \frac{\epsilon}{3n}\right). \tag{4.11}$$

Because we have assumed that $n \geq 1$ and $0 < \epsilon \leq 1$, we can get a further lower bound of μ_i,

$$\begin{aligned} \mu_i &\geq \rho_i \left(1 - \frac{\epsilon}{3n}\right) \\ &\geq \rho_i \left(1 - \frac{1}{3}\right) \\ &\geq \frac{2}{3n}. \end{aligned} \tag{4.12}$$

According to the theory of large numbers, if we get s random independent samples in $\Omega(G_i)$ and s is sufficiently large, the mean of the samples $Z_i^{(1)}, \ldots, Z_i^{(s)}$ will provide a good estimate of ρ_i.

Note that $\mathrm{Var}[Z_i] = \mu_i(1-\mu_i)$ and (4.12), we have

$$\begin{aligned} \frac{\mathrm{Var}[Z_i]}{\mu_i^2} &= \frac{1}{\mu_i} - 1 \\ &\leq \frac{3n}{2}. \end{aligned} \tag{4.13}$$

If we set $s = \lceil 60n^2\epsilon^{-2} \rceil$, then

$$\begin{aligned} \frac{\mathrm{Var}[\overline{Z}_i]}{\mu_i^2} &= \frac{\mathrm{Var}[Z_i]}{s \cdot {\mu_i}^2} \\ &\leq \frac{3n}{2s} \\ &\leq \frac{\epsilon^2}{37n}. \end{aligned} \tag{4.14}$$

Let $N := (\prod_{i=1}^{n} \overline{Z}_i)^{-1}$. notice that $\mathrm{E}[\overline{Z}_1 \overline{Z}_2 \ldots \overline{Z}_n] = \mu_1\mu_2\ldots\mu_n$, we have

$$\begin{aligned} \frac{\mathrm{Var}[\overline{Z}_1 \overline{Z}_2 \ldots \overline{Z}_n]}{(\mu_1\mu_2\ldots\mu_n)^2} &= \frac{\mathrm{E}[{\overline{Z}_1}^2 {\overline{Z}_2}^2 \ldots {\overline{Z}_n}^2]}{(\mu_1\mu_2\ldots\mu_n)^2} - 1 \\ &= \prod_{i=1}^{n} \frac{\mathrm{E}[{\overline{Z}_i}^2]}{{\mu_i}^2} - 1 \\ &= \prod_{i=1}^{n} \left(1 + \frac{\mathrm{Var}[\overline{Z}_i]}{{\mu_i}^2}\right) - 1 \\ &\leq \left(1 + \frac{\epsilon^2}{37n}\right)^n - 1 \\ &\leq \exp\left(\frac{\epsilon^2}{37}\right) - 1 \\ &\leq \frac{\epsilon^2}{36}. \end{aligned} \tag{4.15}$$

By Chebyshev's inequality,

$$\left(1-\frac{\epsilon}{3}\right)\mu_1\mu_2\ldots\mu_n \leq \overline{Z_1}\,\overline{Z_2}\ldots\overline{Z_n} \leq \left(1+\frac{\epsilon}{3}\right)\mu_1\mu_2\ldots\mu_n \tag{4.16}$$

with probability at least $\frac{3}{4}$. We also have the following weaker inequality:

$$e^{-\epsilon/2}\mu_1\mu_2\ldots\mu_n \leq \overline{Z_1}\,\overline{Z_2}\ldots\overline{Z_n} \leq e^{\epsilon/2}\mu_1\mu_2\ldots\mu_n. \tag{4.17}$$

From (4.11) we have

$$e^{-\epsilon/2}\rho_1\rho_2\ldots\rho_n \leq \mu_1\mu_2\ldots\mu_n \leq e^{\epsilon/2}\rho_1\rho_2\ldots\rho_n. \tag{4.18}$$

Combining (4.17) and (4.18), it implies that

$$e^{-\epsilon}\rho_1\rho_2\ldots\rho_n \leq \overline{Z_1}\,\overline{Z_2}\ldots\overline{Z_n} \leq e^{\epsilon}\rho_1\rho_2\ldots\rho_n \tag{4.19}$$

with probability at least $\frac{3}{4}$. Recall that $|\Omega(G)| = (\rho_1\rho_2\ldots\rho_n)^{-1}$ and $N := (\prod_{i=1}^{n}\overline{Z_i})^{-1}$, the proof of (4.6) is done.

To summarize, we perform n iterations and for each we need $\lceil 60n^2\epsilon^{-2}\rceil$ samples. Therefore, the number of samples we need is $O(n^3\epsilon^{-2})$. This proves Theorem 4.2.1 where $\text{poly}(n,\epsilon) = O(n^3\epsilon)$.

We closely followed a computation from Jerrum [Jer03].

4.3 Design of the Markov chain

By Section 4.2.1 we know that a polynomial number of samples suffices to estimate the count. Now we build our Markov chain sampler. Actually, it

is a minor modification of the Jerrum-Sinclair chain [JS89] so that it satisfies our crossing-free requirement.

Let $G = (V, E)$ be a geometric graph as described in Section 4.1. Let Ω denote the set of crossing-free matchings on G, we define the following Markov chain on Ω, for any $M \in \Omega$:

1. Select two vertices u, v uniformly at random and connect them, let e denote the edge (u, v).

2. If e intersects with any edges in M, or passes through any vertices, then $M' \leftarrow M$.

3. Otherwise, there are three exclusive transitions:

   - "$\uparrow$": If both u and v are unmatched in M, then $M' \leftarrow M \cup e$.
   - "$\downarrow$": If $e \in M$, then $M' \leftarrow M \backslash e$.
   - "$\leftrightarrow$": If exactly one of u and v is matched in M, let e' be the edge that contains u or v, then $M' \leftarrow M \backslash e \cup e'$.

4. With probability of $\frac{1}{2}$ do nothing. Otherwise, $M \leftarrow M'$.

To see more clearly the transitions of the Markov chain, given the current matching (see Figure 4.1),

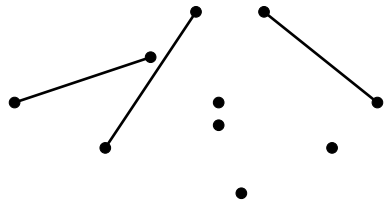
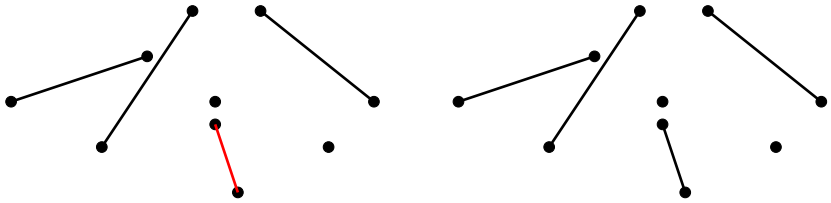


Figure 4.1: The original matching M.

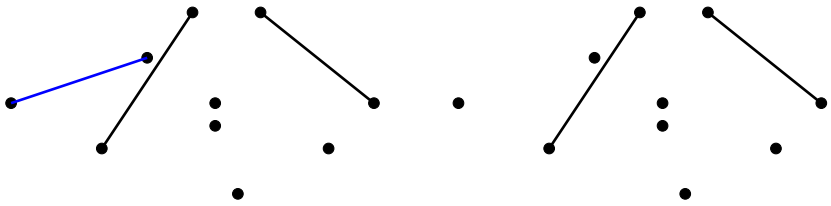
The three exclusive transitions are shown in following figures:



(a) The selected red edge e is not in M.

(b) $M' \leftarrow M \cup e$.

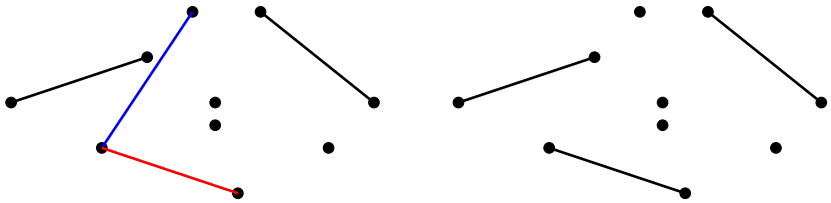
Figure 4.2: The $\uparrow$ transition.



(a) The selected blue edge e is in M.

(b) $M' \leftarrow M \backslash e$.

Figure 4.3: The $\downarrow$ transition.



(a) The selected red edge e is not in M, and one of its endpoints has been matched (the blue edge e').

(b) $M' \leftarrow M \backslash e' \cup e$.

Figure 4.4: The $\leftrightarrow$ transition.

This Markov chain is irreducible since for any two matchings M_1, M_2, we can move from M_1 to M_2 simply by removing all edges in M_1, then adding all edges in M_2; it is also aperiodic because of the self-loops.

4.4 Bounding the mixing time of the Markov chain

From now on we will work with geometric graphs with n vertices arranged in a grid of dimensions $h \times w$, where h is the number of rows, w is the number of columns (see Figure 4.5). Therefore, $n = h \cdot w$. We also restrict the length of edges by a constant c, where the length of an edge is the Manhattan distance between its two endpoints. Let $\Omega(G)$ denote the set of crossing-free matchings (of all sizes) in the graph G.

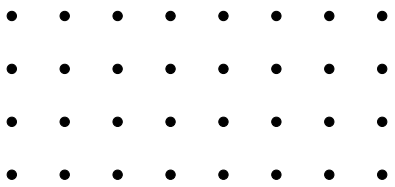


Figure 4.5: A $h = 4, w = 8$ grid.

First, we need to modify the Markov chain in Section 4.3 a little bit. Since we bound the length of edges by c, the step 2 of the Markov chain should be replaced by

- If e intersects with any edges in M, or passes through any vertices, or the length of e is greater than c, then $M' \leftarrow M$.

From now on, we will work with this new Markov chain.

To bound the mixing time, we first need to define a set of canonical paths Γ. Given two matchings I (initial) and F (final), we connect I and F by a canonical path $\gamma_{IF} := \{I = M_0 \rightarrow M_1 \rightarrow \ldots \rightarrow M_l = F\}$. Notice that the transition from M_i to M_{i+1} ($0 \leq i \leq l-1$, where l is the length of the canonical path) is from $\{\uparrow, \downarrow, \leftrightarrow\}$. While moving from I to F through the path, we only need to add and remove edges in $I \oplus F$, and leave edges in $I \cap F$ untouched. Let us consider what $I \oplus F$ looks like (see Figure 4.6).

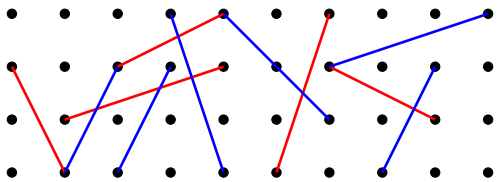


Figure 4.6: The color of I is blue, of F is red. This is what $I \oplus F$ looks like (edges in $I \cap F$ disappeared).

To move from I to F, we process the edges in $I \oplus F$ in the direction from left to right. To illustrate more clearly, we add in Figure 4.6 some black lines whose x coordinate is $(i + 0.5)$, where $1 \leq i \leq w - 1$. After adding these lines, the original graph becomes Figure 4.7.

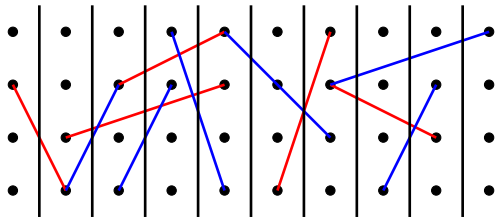
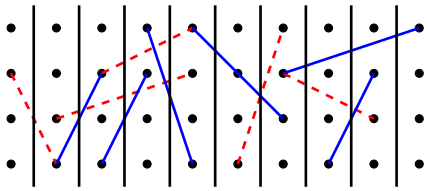
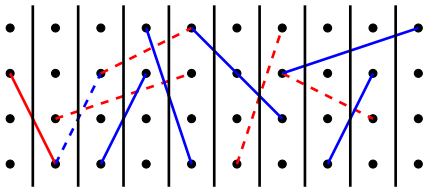


Figure 4.7: After adding black lines into Figure 4.6.

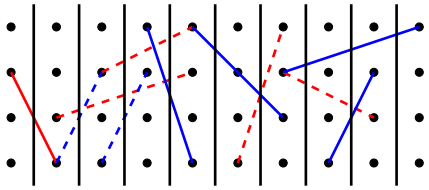
Processing edges from left to right means that we will not process the edges that cross the $(i+1)^{th}$ black line before finish processing edges that cross the i^{th} black line. For those red edges that can not be added, we leave them untouched and move to the next black line . To see more clearly, please look at the following figures:



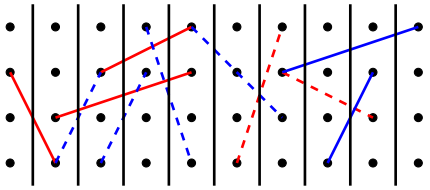
(a) At the beginning, the current component is I.



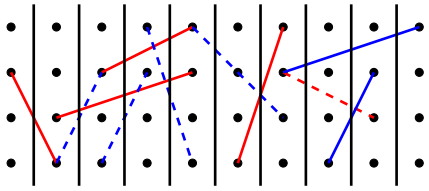
(b) After processing the 1^{st} black line (one $\leftrightarrow$ transition).



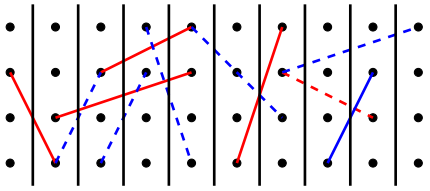
(c) After processing the 3^{rd} black line (one $\downarrow$ transition).



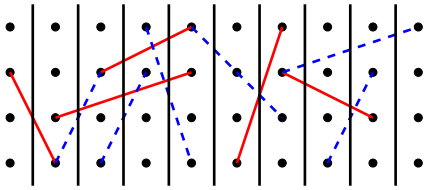
(d) After processing the 4^{th} black line (one $\uparrow$ transition, one $\downarrow$ transition, one $\leftrightarrow$ transition).



(e) After processing the 6^{th} black line (one $\uparrow$ transition).



(f) After processing the 7^{th} black line (one $\downarrow$ transition).



(g) After processing the 8^{th} black line (one $\downarrow$ transition and one $\uparrow$ transition). The current component is exactly F.

Figure 4.8: To move from I to F, we process edges from left to right. Dashed edges mean they currently do not exist.

There are no figures for the component after processing the 2^{nd}, 5^{th},

and 9^{th} black line because we did not touch any edges while processing these three black line.

Recall the definition of congestion of canonical paths. For the matching problem, (3.7) can be rewritten as

$$\varrho(\Gamma) := \max_{t=(M,M')} \left\{ \frac{1}{\pi(M)P(M,M')} \sum_{(I,F)\in \mathrm{cp}(t)} \pi(M)\pi(M')|\gamma_{IF}| \right\}, \tag{4.20}$$

where $|\gamma_{IF}|$ is the length of the path γ_{IF}, and $\mathrm{cp}(t)$ is the set of (I,F) pairs whose canonical path γ_{IF} uses transition t. Furthermore, we can bound (4.20) by the following procedures:

$$\varrho(\Gamma) = \max_{t=(M,M')} \left\{ \frac{1}{|\Omega| \cdot P(M,M')} \sum_{(I,F)\in \mathrm{cp}(t)} |\gamma_{IF}| \right\} \tag{4.21}$$

$$\leq \frac{n^2}{|\Omega|} \sum_{(I,F)\in \mathrm{cp}(t)} |\gamma_{IF}| \tag{4.22}$$

$$\leq \frac{n^3}{|\Omega|} |\mathrm{cp}(t)|. \tag{4.23}$$

(4.21) is because we are sampling from the uniform distribution; (4.22) is because M and M' are differ at exactly one edge, and there are n^2 different edges; (4.23) is because moving from I to F requires at most n transitions (since there are n vertices, if both I and F are perfect matchings, the slowest way to move from I to F is removing the $n/2$ edges in I, then adding the $n/2$ edges in F).

We need to find a way to estimate $|\mathrm{cp}(t)|$, and we hope that

$$|\mathrm{cp}(t)| \leq |\Omega| \cdot \mathrm{poly}(n). \tag{4.24}$$

To bound $\mathrm{cp}(t)$, let us first consider the symmetric difference of the following two matchings:

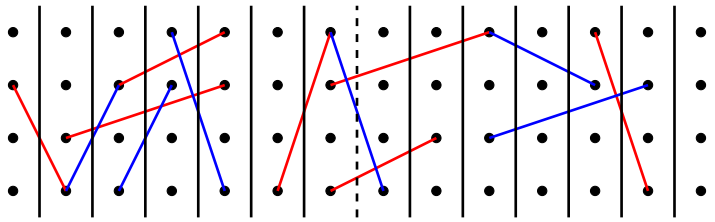


Figure 4.9: $I \oplus F$.

Given a transition $t = (M, M')$, let us see what $M \cup M'$ looks like:

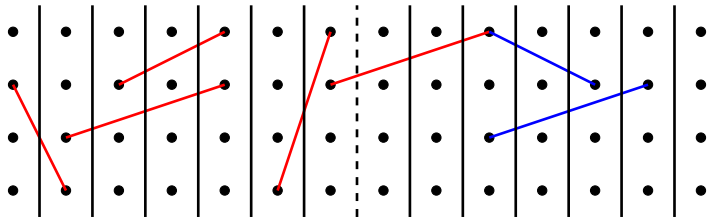


Figure 4.10: $M \cup M'$.

In Figure 4.10, the dashed black line is where we process the transition t, where t is the $\leftrightarrow$ transition over the adjacent two red and blue edges (the blue edge is in M and the red edge is in M'). Then we consider what $I \oplus F \oplus (M \cup M')$ looks like (see Figure 4.11).

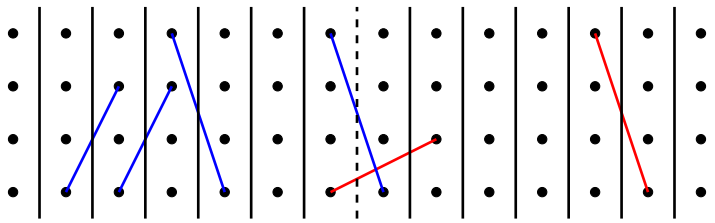


Figure 4.11: $I \oplus F \oplus (M \cup M')$.

For convenience, let $C = I \oplus F \oplus (M \cup M')$, and b_t denote the corresponding black line where we process the transition t. Then C can be divided into the following three parts:

- C_{left}: set of edges that are on the left of b_t and do not cross with it .
- C_{right}: set of edges that are on the right of b_t and do not cross with it.
- $C_{current}$: set of edges that cross b_t.

In Figure 4.11, it is easily to find out that C agrees with I over C_{left}, and agrees with F on C_{right}.

Given a transition $t = (M, M')$, for all $(I, F) \in \text{cp}(t)$, we define the encoding function η_t, where

$$\eta_t(I, F) = I \oplus F \oplus (M \cup M'). \tag{4.25}$$

There are two important claims about $\eta_t(I, F)$:

Claim 4.4.1. *Let $C = \eta_t(I, F)$, then $C_{left} \cup C_{right}$ is a crossing-free matching.*

Proof. Since that C_{left} are edges that are strictly on the left of b_t, C_{right} are edges that are strictly on the right of b_t. It is obvious that while we are processing the transition t, we have finished processing edges in C_{left}, and have not touched edges in C_{right}. It means that C agrees with I on C_{left}, and agrees with F on C_{right}. Both C_{left} and C_{right} are crossing-free matchings. Since they are mutually exclusive, then $C_{left} \cup C_{right}$ is a crossing-free matching. □

Claim 4.4.2. *$\eta_t(I, F)$ is injective function that $\eta_t : \mathrm{cp}(t) \to \Omega \times B$, where B is the set of edge sets that may cross the black line.*

Proof. According to Claim 4.4.1, for any t and all (I, F) in $\mathrm{cp}(t)$, $(C_{left} \cup C_{right}) \in \Omega$ always holds. The troublesome occurs only in $C_{current}$ since there might exist crossings. To construct the injective function, we first define a set B that $B = \bigcup_{i=0}^{d} B_i$, where B_i is the set of edge sets, such that for each set in B_i, there are i edges cross the black line, and d is the maximum number of edges that cross b_t. For every (I, F) in $\mathrm{cp}(t)$, we can select a crossing-free matching in Ω (corresponds to $C_{left} \cup C_{right}$), and a set of edges in B (corresponds to the troublesome occurs in $C_{current}$), then uniquely rebuild (I, F). Thus, η_t is an injective function. □

According to Claim 4.4.2, we will have the following inequality:

$$|\mathrm{cp}(t)| \leq |\Omega| \cdot |B|. \tag{4.26}$$

Recall the definition of B, that $B = \bigcup_{i=0}^{d} B_i$, then $|B|$ can be computed by

$$B = \sum_{i=0}^{d} |B_i|, \tag{4.27}$$

where

$$|B_i| = \binom{m}{i}, \tag{4.28}$$

where m is the number of potential edges (an edge is potential means it cross the black line).

We first try to bound m. Given a black line b, let b_x denote its x coordinate. Without loss of generality, assume that $b_x - c \geq 1$ and $b_x + c \leq w$. For edges that can cross b_x, their endpoints must locate in a $2c \times h$ window, and each vertex can be connected with fewer than c^2 vertices (see the rectangular region in Figure 4.12). Therefore, we have $m \leq c^3 h$.

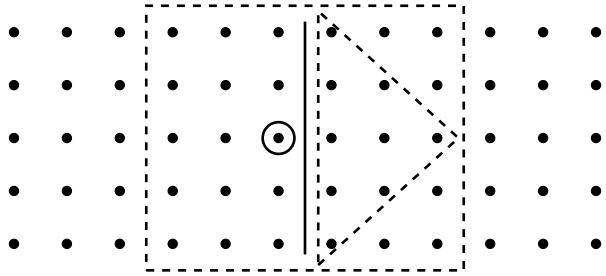


Figure 4.12: When $c = 3$, $h = 5$, $w = 12$, the blue region contains $2ch = 30$ vertices that can be the endpoints of potential edges; the triangular region contains $c^2 = 9$ vertices that might allowed to be connected with the circled vertex. Therefore, there are $c^3 h = 135$ potential edges.

Then we try to bound b, since b is the maximum number of crossings on the black line, then b must be less than ch (see Figure 4.13).

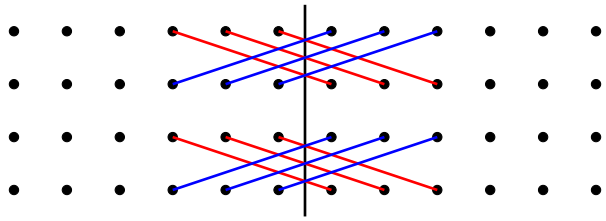


Figure 4.13: $h = 4$, $w = 12, c = 4$, 12 crossings.

We have successfully bounded b and m. Recall (4.27) and (4.28), B can

be bounded by:

$$
\begin{aligned}
|B| &= \sum_{i=0}^{d} |B_i| \\
&\leq \sum_{i=0}^{ch} \binom{c^3 h}{i} \\
&\leq (ch+1)\binom{c^3 h}{ch} \\
&\leq (ch+1) \cdot \frac{(c^3 h)^{ch}}{(ch)!} \\
&\leq (ch+1) \cdot \frac{(c^3 h)^{ch}}{\sqrt{2\pi ch} \cdot \left(\frac{ch}{e}\right)^{ch}} \qquad (4.29) \\
&= O\left(\sqrt{h} \cdot (c^2 e)^{ch}\right). \qquad (4.30)
\end{aligned}
$$

(4.29) is following the Stirling's formula.

According to (4.30), let us consider the following three cases of the arrangement of the input grid.

1. If $h = O(1)$, then

$$|B| = O(1). \qquad (4.31)$$

2. If $h = O(\log n)$, then

$$|B| = O(\sqrt{\log n} \cdot n^d), \qquad (4.32)$$

   where d is a function of c. Thus, d is a constant.

3. If $h = O(n)$, then

$$|B| = O\left(\sqrt{n} \cdot a^n\right), \qquad (4.33)$$

   where $a = (c^2 e)^c$.

It seems that when $h = \text{poly}(n)$, $|B|$ will be in exponential in the input size.

We have seen that $h = O(\log n)$ is the boundary between polynomial and exponential mixing time. Therefore, we have to restrict the arrangement of input to be $h = O(\log n)$, then the congestion ϱ can be bounded by

$$\begin{aligned}\varrho(\Gamma) &\leq \frac{n^3}{|\Omega|}|\text{cp}(t)| \\ &\leq n^3|B| \\ &= O(\sqrt{\log n} \cdot n^{d+3}).\end{aligned} \tag{4.34}$$

Combine (3.8) and (4.34), we have

$$\tau(\epsilon) = O\left(\sqrt{\log n} \cdot n^{d+3}(n^2 + \ln \epsilon^{-1})\right). \tag{4.35}$$

Finally, we have got an upper bound on the running time. Remember that there are n different ratios, and for each ratio we need to sample at most $\lceil 60n^2\epsilon^{-2} \rceil$ times, and we spend $\tau(\epsilon)$ time on each sample. Therefore, the running time of the algorithm is

$$\text{T}(n, \epsilon) = O\left(\sqrt{\log n} \cdot n^{d+6}\epsilon^{-2}(n^2 + \ln \epsilon^{-1})\right). \tag{4.36}$$

We now have proved the two following theorems about sampling and counting of crossing-free matchings:

Theorem 4.4.1 (Sampling)**.** *For a geometric graph where the points are arranged in a $h \times w$ grid where $h = O(\log n)$, $n = hw$, and all edges lengths are bounded by a constant c, the mixing time of the Markov chain is*

$$\tau(\epsilon) = O\left(\sqrt{\log n} \cdot n^{d+3}(n^2 + \ln \epsilon^{-1})\right),$$

where d is a constant.

Theorem 4.4.2 (Counting)**.** *Under the assumptions of Theorem 4.4.1, there is an fpras for counting all crossing-free matchings of the graph. The running time of the algorithm is*

$$\mathrm{T}(n,\epsilon) = O\left(\sqrt{\log n}\cdot n^{d+6}\epsilon^{-2}(n^2+\ln\epsilon^{-1})\right),$$

where d is a constant.

According to Theorem 4.4.1 and Theorem 4.4.2, we have the following pseudocode of the approximate counting algorithm:

Algorithm 3: Counting crossing-free matchings in grids

input : A grid G with n vertices, where $n = hw$; an error tolerance of output ϵ; a length restriction of edges c

output: An estimate of the number of crossing-free matchings in G

$product \leftarrow 1$
$num \leftarrow \lceil 60n^2\epsilon^{-2} \rceil$
/* num denotes the number of samples we need for estimating each ratio, and num is less than $\lceil 60n^2\epsilon^{-2} \rceil$. */
$t \leftarrow$ a sufficiently large number
/* t denotes the number of steps we run the Markov chain. t should be sufficiently large for the Markov chain to converge. */
for $i \leftarrow 1$ **to** n **do**
    $product \leftarrow product \times$ sample$(G, i, \epsilon, c, num, t)$
end
return $1/product$

Function sample$(G, i, \epsilon, c, num, t)$ **is**
    if $i = 1$ **then**
        return 1
    else
        $M \leftarrow G_i$
        $count \leftarrow 0$
        for $j \leftarrow 1$ **to** num **do**
            Run the Markov chain on M for t steps
            if *M contains the i^{th} vertex* **then**
                $count \leftarrow count + 1$
            end
        end
        return $count/s$
    end
end

Chapter 5

Conclusion and future work

After adding restrictions on the input graphs, we proved that the mixing time is bounded by a polynomial of the input size. Apparently, both the number of samples and the mixing time of the Markov chain are bounded by a very loose upper bound. The question is, can we bound them tighter? Once we can, can we then sample crossing-free matchings in more general graphs?

5.1 Towards a tighter bound on the number of samples

Recall (4.3), (4.4) and (4.5). Notice that $|A_j|$ is increasing when j is increasing. Even $|A_{i-1}|$ is much smaller than $|\Omega(G_{i-1})|$. Therefore,

$$\frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|} \gg \frac{1}{n}. \tag{5.1}$$

We still do not know whether we can bound the above ratio by a constant. If we can, according to Jerrum [Jer03], for each ratio we only need $O(\epsilon^{-2}n)$ instead of $O(\epsilon^{-2}n^2)$ samples.

We can also try to define another self-reducible model. Since there are w columns of vertices, we define a sequence of graphs G_i where $1 \leq i \leq w$, such that G_i contains the first i columns of vertices (then, $|G_0| = 1$). Then

$|\Omega(G)|$ can be estimated by

$$|\Omega(G)| = |\Omega(G_n)| = \left(\prod_{i=1}^{n} \frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|} \right)^{-1}. \tag{5.2}$$

At this point we do not know how to map G_i to G_{i-1} and how to find a lower bound on $\frac{|\Omega(G_{i-1})|}{|\Omega(G_i)|}$. It seems hard to bound it by a constant. It may not be possible to bound it by a polynomial of n since we currently have only w subgraphs (formerly we had n subgraphs). Then the number of samples will both be affected by the arrangement of the grid and the number of vertices.

5.2 Towards a tighter bound on the mixing time of the Markov chain

Recall the definition of $|B|$, which is the set of red and blue edges that are crossing the current black line. Our previous bound on the mixing time is directly affected by $|B|$. Is it possible to bound $|B|$ or $\mathrm{E}[|B|]$ by a polynomial of n? (Bounding it by a constant seems impossible.) Can we find another kind of canonical paths, or do we need to use, or develop, other techniques for this problem?

5.3 Counting the number of matchings on more general graphs and with fewer restrictions

Up to now we have made restrictions on the arrangement of vertices and the length of the edges. Is it possible to sample crossing-free matchings after removing restrictions on the dimensions of the grid, or on the length

of edges? More generally, is it possible to sample crossing-free matchings in general graphs?

5.4 Relation to dynamic programming

Our current restrictions on the input imply that the input has the so-called treewidth parameter bounded by a polylogarithmic factor. Often, problems on instances with bounded treewidth can be solved exactly by a dynamic programming approach, as has been shown in the celebrated Courcelle's Theorem [Cou90]. However, this dynamic programming approach does not generalize to more complex instances. The objective of this thesis was to utilize another approach, the Markov chain Monte Carlo technique, as a possibility of dealing with such instances. Our MCMC algorithm is more general than dynamic programming as it can be used on any instance. However, at present, we have polynomial bounds on the mixing time only for the case of graphs with bounded treewidth. On the other side, we have no indication of slow mixing for other input instances. We believe that it in fact mixes rapidly for all inputs.

Bibliography

[AD86] David Aldous and Persi Diaconis. Shuffling cards and stopping times. *American Mathematical Monthly*, pages 333–348, 1986.

[Ald83] David Aldous. Random walks on finite groups and rapidly mixing Markov chains. In *Séminaire de Probabilités XVII 1981/82*, pages 243–297. Springer, 1983.

[BDGJ99] Russ Bubley, Martin Dyer, Catherine Greenhill, and Mark Jerrum. On approximately counting colorings of small degree graphs. *SIAM Journal on Computing*, 29(2):387–400, 1999.

[BDK08] Magnus Bordewich, Martin Dyer, and Marek Karpinski. Path coupling using stopping times and counting independent sets and colorings in hypergraphs. *Random Structures & Algorithms*, 32(3):375–399, 2008.

[BGK+07] Mohsen Bayati, David Gamarnik, Dimitriy Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, pages 122–127. ACM, 2007.

[CMSS13] Pietro Caputo, Fabio Martinelli, Alistair Sinclair, and Alexandre Stauffer. Random lattice triangulations: structure and algorithms. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, pages 615–624. ACM, 2013.

[CMSS15] Pietro Caputo, Fabio Martinelli, Alistair Sinclair, and Alexandre Stauffer. Dynamics of lattice triangulations on thin rectangles. *preprint arXiv:1505.06161*, 2015.

[Cou90] Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[DFJ02] Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. *SIAM Journal on Computing*, 31(5):1527–1541, 2002.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993.

[Jer03] Mark Jerrum. *Counting, sampling and integrating: algorithms and complexity*. Springer Science & Business Media, 2003.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

[JSV04] 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(4):671–697, 2004.

[JVV86] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[Kas63] Pieter W Kasteleyn. Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4(2):287–293, 1963.

[KKMS09] František Kardoš, Daniel Král', Jozef Miškuf, and Jean-Sébastien Sereni. Fullerene graphs have exponentially many perfect matchings. *Journal of Mathematical Chemistry*, 46(2):443–447, 2009.

[KR01] V.S. Anil Kumar and H. Ramesh. Coupling vs. conductance for the Jerrum-Sinclair chain. *Random Structures and Algorithms*, 18(1):1–17, 2001.

[Sin92] Alistair Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1(04):351–370, 1992.

[Sin09] Alistair Sinclair. CS294 Markov Chain Monte Carlo: Foundations & Applications, Fall 2009.

[SW06] Micha Sharir and Emo Welzl. On the number of crossing-free matchings, cycles, and partitions. *SIAM Journal on Computing*, 36(3):695–720, 2006.

[Val79a] Leslie G Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Val79b] Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[Vig01] Eric Vigoda. A note on the Glauber dynamics for sampling independent sets. *Electronic Journal of Combinatorics*, 8(1):1–8, 2001.

[Wet14] Manuel Wettstein. Counting and enumerating crossing-free geometric graphs. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*. ACM, 2014.