CSE 598: Markov Chain and Monte Carlo Methods

Joshua J. Daymude

Fall 2019

1 Introduction

These are self-contained notes on the theory of *Markov Chains and Monte Carlo methods*, emphasizing methods and techniques for (*i*) counting the size of very large sets — e.g., the set of all matchings of a large graph — and (*ii*) randomly sampling from such sets. These two objectives can be approached using cohesive and elegant theory, which this course details. If you need additional references beyond these notes, these texts may be helpful:

1. *Counting, Sampling, and Integrating: Algorithms and Complexity* by Jerrum [11].

2. *Markov Chains and Mixing Times* by Levin-Peres-Wilmer [16].

Author Disclaimers These notes are in continuing development, and as such there may be slight typos and changes in the material. Small amounts of extra credit may be given for catching mistakes or addressing red comments, which indicate places in the standard proofs or other arguments where the logic is not totally clear. These notes should be seen as a supplement but not a replacement to attending lectures and asking questions.

2 Exact Counting

We begin with the study of classical algorithms for *exact counting*, of which there are very few. Most counting algorithms can only hope to guarantee *approximate* counts. In Section 2.1, we present Kasteleyn's algorithm [13, 14] for computing the number of perfect matchings of a planar graph. Virtually all known algorithms for exact counting rely on reductions to the matrix determinant; Kasteleyn's algorithm utilizes a connection to the determinant of the graph's adjacency matrix. Thus, counting perfect matchings of a planar graph is in **P**, since computing the determinant of an adjacency matrix can be done in $\mathcal{O}(n^3)$ time, where n is the number of vertices.

On the negative side, in Section 2.2 we present a result by Valiant [23] that motivates how such computations are, in general, **#P**-complete (the counting equivalent of **NP**-complete for decision problems). To deal with this, we introduce the concept of *fully polynomial randomized approximation schemes* and *fully polynomial almost uniform samplers* in Section 2.3 as methods for getting sufficiently accurate results efficiently.

2.1 An Exact Counting Algorithm for Perfect Matchings in Planar Graphs

Given an undirected graph $G = (V, E)$, where $n = |V|$ is even, we would like to compute $|\mathcal{P}|$, where $\mathcal{P}$ is the set of all perfect matchings of G. Computing $|\mathcal{P}|$ exactly is generally intractable — for example, it is known to be **#P**-complete if G is bipartite, as we will see in Section 2.2 — but is possible to do in polynomial time for some special situations. In particular, we prove:

Theorem 2.1. *Let $G = (V, E)$ be a planar graph with $|V|$ even, and let $\mathcal{P}$ be the set of all perfect matchings of G. Then $|\mathcal{P}|$ can be calculated in polynomial time.*

The general flow of this argument is as follows. We first show that we can orient the edges of any planar graph G such that the orientation is *Pfaffian.* This follows from a constructive argument which builds a Pfaffian orientation from a planar embedding of G. We then show that for any Pfaffian orientation $\vec{G}$ of G with adjacency matrix $\vec{A}$, $\det(\vec{A}) = |\mathcal{P}|^2$. This is a result of Kasteleyn [14]; notably, Temperley and Fisher independently proved a similar result for $\mathbb{Z}^2$ [22].

2.1.1 Obtaining a Pfaffian Orientation

An *orientation* of a graph $G = (V, E)$ replaces each edge $(i, j) \in E$ with either $\vec{ij}$ or $\vec{ji}$. Let $\vec{G} = (V, \vec{E})$ denote the resulting directed graph, and let $\vec{A}$ be its skew-symmetric adjacency matrix, defined as follows:

$$\vec{A}(i,j) = \begin{cases} 1 & \text{if } \vec{ij} \in \vec{E}, \\ -1 & \text{if } \vec{ji} \in \vec{E}, \text{ and} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We now formally define what it means for an orientation to be Pfaffian.

Definition 2.2. *An even length cycle C of an undirected graph G is said to be oddly oriented in an orientation $\vec{G}$ if, given a traversal direction of C, there are an odd number of edges in C oriented in the opposite direction.*

We note that it doesn't matter whether the traversal direction is clockwise or counter-clockwise; since C has even length, there is an odd number of edges in C oriented clockwise if and only if there are also an odd number of edges in C oriented counter-clockwise. Now consider a critical observation about perfect matchings.

Observation 2.3. *For any pair of perfect matchings $P, P' \in \mathcal{P}$, $P \cup P'$ consists of vertex-disjoint cycles of even length and single edges ($= P \cap P'$).*

By Observation 2.3, the following is well-defined.

Definition 2.4. *An orientation $\vec{G}$ of an undirected graph G is said to be Pfaffian if for all pairs of perfect matchings $P, P' \in \mathcal{P}$, all cycles of $P \cup P'$ are oddly oriented.*

We need one last piece of machinery before proving the result of this section.

Formula 2.5 (Euler's). *For a planar embedding of $G = (V, E)$ with faces F, $|V| - |E| + |F| = 2$.*

Lemma 2.6. *For any planar graph G, we can construct a Pfaffian orientation $\vec{G}$ in polynomial time.*

Proof. We will construct an orientation $\vec{G}$ in which the boundary of every face (except possibly the outer one) contains an odd number of clockwise edges (*). This construction builds the orientation inductively on the number of edges (conditioned on the graph being connected). The base case is any spanning tree of G. Since the outer face is the only face, any arbitrary orientation satisfies (*). For a general planar graph, consider an edge e on the outer face. By induction, we can obtain an orientation of $G - e$ satisfying (*). Adding e back in creates at least one new face f. Ignoring e, f must have either an even or odd number of clockwise edges. Thus, we can always orient e such that f contains an odd number of clockwise edges. This concludes the construction.

Note that by removing edges from the outer face first, adding edges back in will only ever split the outer face (and not any internal face) into new faces. This ensures that any internal faces keep their parity of clockwise edges, leaving us only with the newly created face to ensure has an odd number of clockwise edges.

This construction clearly runs in polynomial time; it remains to show that the resulting orientation $\vec{G}$ is Pfaffian. Consider any cycle C in G, and consider the induced subgraph of $\vec{G}$ on C, including all vertices and edges on or inside C. Let $\{f_1, f_2, \ldots, f_{F_{in}}\}$ be the set of internal faces in this subgraph. Let V_{on} (resp., E_{on}) be the number of vertices (resp., edges) on C. Similarly, let V_{in} (resp., E_{in}) be the number of vertices (resp., edges) strictly inside C. Finally, let $E_{on}^{cw}(f_i)$ (resp., $E_{on}^{cw}(C)$) denote the number of clockwise edges on face f_i (resp., C in $\vec{G}$).

By Euler's formula (Formula 2.5),

$$(V_{on} + V_{in}) - (E_{on} + E_{in}) + (F_{in} + 1) = 2$$

But since C is a cycle, $V_{on} = E_{on}$, so

$$V_{in} - E_{in} + F_{in} = 1 \tag{2}$$

Our construction ensured that (*) was satisfied for each face f_i; i.e., $E_{on}^{cw}(f_i) \equiv 1 \bmod 2$. Thus,

$$\sum_{i=1}^{F_{in}} E_{on}^{cw}(f_i) \equiv F_{in} \bmod 2 \tag{3}$$

Since each edge inside the cycle is clockwise to exactly one of the two faces incident to it and each clockwise edge on C is clockwise to the internal face incident to it, we have $\sum_{i=1}^{F_{in}} E_{on}^{cw}(f_i) = E_{in} + E_{on}^{cw}(C)$. Plugging in (3), we have:

$$F_{in} \equiv E_{in} + E_{on}^{cw}(C) \bmod 2$$

Plugging in (2) yields:

$$F_{in} \equiv V_{in} + F_{in} - 1 + E_{on}^{cw}(C) \bmod 2 \Rightarrow V_{in} \not\equiv E_{on}^{cw}(C) \bmod 2 \tag{4}$$

Consider any pair of perfect matchings $P, P' \in \mathcal{P}$, and consider any cycle $C \in P \cup P'$. By Observation 2.3, C must have even length. Furthermore, if any vertex inside C is matched in P with a vertex outside C, then the planarity of G is contradicted; thus the vertices inside C are matched with each other and V_{in} is even. By (4), $E_{on}^{cw}(C)$ must be odd and therefore $\vec{G}$ is Pfaffian. □

Robertson, Seymour, and Thomas [18] gave a polynomial time algorithm for deciding whether a bipartite graph has a Pfaffian orientation (and constructing such an orientation if one exists). However, the problem is still open for general graphs.

2.1.2 Reducing to the Determinant

Let $\overleftrightarrow{G} = (V, \overleftrightarrow{E})$ denote the directed graph formed by replacing each edge $(i, j) \in E$ with a pair of anti-symmetric directed edges $\vec{ij}$ and $\vec{ji}$. We begin with a lemma that reduces the problem of counting perfect matchings of G to counting *even cycle covers* of $\overleftrightarrow{G}$. An even cycle cover is a set of vertex-disjoint directed cycles of even length which cover V.

Lemma 2.7. *Let $G = (V, E)$ be a graph with $|V|$ even, and let $\mathcal{P}$ be the set of all perfect matchings of G. Let $\overleftrightarrow{G}$ be as above, and let $\mathcal{E}$ be the set of all even cycle covers of $\overleftrightarrow{G}$. Then $|\mathcal{P}|^2 = |\mathcal{E}|$.*

Proof. We define a bijection between ordered pairs of perfect matchings and even cycle covers. First consider an ordered pair of perfect matchings $P, P' \in \mathcal{P}$. By Observation 2.3, $P \cup P'$ consists of vertex-disjoint cycles of even length and single edges. Any single edge $(i, j) \in P \cup P'$ (i.e., those in $P \cap P'$) form a directed cycle of length 2 in $\overleftrightarrow{G}$: $(\vec{ij}, \vec{ji})$. For each cycle $C \in P \cup P'$, order the vertices of C arbitrarily and let v be the minimum vertex in C. Orient the edge incident to v in P away from v, and use this orientation for the rest of C to form a directed cycle in $\overleftrightarrow{G}$. The resulting set of directed cycles is vertex-disjoint because the original cycles and single edges were, and since P and P' are perfect matchings, all vertices are covered. This mapping is invertible, so $|\mathcal{P}|^2 \leq |\mathcal{E}|$.

Now consider any even cycle cover $\Sigma \in \mathcal{E}$. For each directed cycle $\overleftrightarrow{C} \in \Sigma$, again define an arbitrary ordering of the vertices and let v be the minimum vertex in $\overleftrightarrow{C}$. Let $\vec{vw}$ be the edge directed out of v; assign its undirected counterpart (v, w) to P and then alternate assigning edges to P and P' around $\overleftrightarrow{C}$. Since $\overleftrightarrow{C}$ has even length, P and P' are both valid matchings; since Σ is an even cycle cover, P and P' both match all the vertices. Thus, $|\mathcal{E}| \leq |\mathcal{P}|^2$. □

Theorem 2.8 (Kasteleyn [14])**.** *Let $G = (V, E)$ be a planar graph with $|V|$ even, and let $\mathcal{P}$ be the set of all perfect matchings of G. For any Pfaffian orientation $\vec{G}$ of G with adjacency matrix $\vec{A}$,* $\det(\vec{A}) = |\mathcal{P}|^2$.

Proof. Consider any Pfaffian orientation $\vec{G}$ of G, and let $\vec{A}$ be its adjacency matrix. Let $n = |V|$, and recall that the *determinant* of $\vec{A}$, an $n \times n$ matrix, is given by:

$$\det(\vec{A}) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i \in [n]} \vec{A}(i, \sigma(i)),$$

where S_n is the set of permutations of $[n] = \{0, 1, \ldots, n-1\}$ and $sgn(\sigma) = (-1)^{N(\sigma)}$ where $N(\sigma) = |\{i, j : i < j \text{ and } \sigma(i) > \sigma(j)\}|$ is the number of inversions under permutation σ.

It will be useful to consider alternate forms of $sgn(\sigma)$. Let $\sigma = \gamma_1 \cdots \gamma_k$ be the cycle decomposition of σ, and let $|\gamma_i|$ denote the length of cycle γ_i. Observe that:

$$sgn(\gamma_i) = \begin{cases} -1 & \text{if } |\gamma_i| \text{ is even, and} \\ 1 & \text{if } |\gamma_i| \text{ is odd.} \end{cases}$$

Equivalently, we have $sgn(\gamma_i) = (-1)^{|\gamma_i|-1}$. Then,

$$sgn(\sigma) = \prod_{i=1}^{k} sgn(\gamma_i) = \prod_{i=1}^{k} (-1)^{|\gamma_i|-1} = (-1)^{\sum_{i=1}^{k}(|\gamma_i|-1)} = (-1)^{n-k} \tag{5}$$

We claim that only permutations composed of even length cycles contribute to $\det(\vec{A})$. Suppose some permutation $\sigma \in S_n$ with cycle decomposition $\sigma = \gamma_1 \cdots \gamma_k$ contains at least one odd length cycle; let γ_i be its first one. Let V_i denote the vertices of γ_i. Reverse the direction of γ_i to obtain $\sigma' = \gamma_1 \cdots \gamma_{i-1}\gamma_i^{-1}\gamma_{i+1}\cdots\gamma_k$. Since $|\gamma_i|$ has odd length, the number of clockwise edges and the number of counter-clockwise edges must have opposite parity. Thus, by (1),

$$\prod_{j\in V_i} \vec{A}(j,\sigma(j)) = -\prod_{j\in V_i} \vec{A}(j,\sigma'(j))$$

Applying this argument to each odd cycle, we have:

$$\prod_{i\in[n]} \vec{A}(i,\sigma(i)) = -\prod_{i\in[n]} \vec{A}(i,\sigma'(i))$$

By (5), $sgn(\sigma) = (-1)^{n-k} = sgn(\sigma')$. So $sgn(\sigma)\prod_{i\in[n]} \vec{A}(i,\sigma(i)) + sgn(\sigma')\prod_{i\in[n]} \vec{A}(i,\sigma'(i)) = 0$, which proves the claim.

So we need only consider permutations composed only of even length cycles. Furthermore, observe that $\prod_{i\in[n]} \vec{A}(i,\sigma(i)) \neq 0$ if and only if for every $\forall i, (i,\sigma(i)) \in E$. Then the permutations yielding non-zero terms are exactly those which correspond to even cycle covers of $\vec{G}$. So,

$$\det(\vec{A}) = \sum_{\sigma\in\mathcal{E}} sgn(\sigma)\prod_{i\in[n]} \vec{A}(i,\sigma(i))$$

Consider any $\sigma \in \mathcal{E}$, and let $k(\sigma)$ denote the number of (even length) cycles in σ. Since $\vec{G}$ is Pfaffian and each cycle $\gamma_i \in \sigma$ has even length, each γ_i must have an odd number of both clockwise and counter-clockwise edges in each cycle, implying:

$$\prod_{i\in[n]} \vec{A}(i,\sigma(i)) = \prod_{i\in[k(\sigma)]}\prod_{j\in V_i} \vec{A}(j,\sigma(j)) = (-1)^{k(\sigma)}$$

Finally, we can compute the determinant, recalling the alternate form for $sgn(\sigma)$ in (5) and the fact that n is even, by supposition.

$$\det(\vec{A}) = \sum_{\sigma\in\mathcal{E}} sgn(\sigma)\prod_{i\in[n]} \vec{A}(i,\sigma(i)) = \sum_{\sigma\in\mathcal{E}}(-1)^{n-k(\sigma)}(-1)^{k(\sigma)} = \sum_{\sigma\in\mathcal{E}}(-1)^n = |\mathcal{E}|$$

Combining this result with Lemma 2.7, we have $\det(\vec{A}) = |\mathcal{E}| = |\mathcal{P}|^2$. □

We conclude this section by wrapping up the results to prove our original theorem.

Proof of Theorem 2.1. Consider any planar graph $G = (V,E)$ with $n = |V|$ even, and let $\mathcal{P}$ denote the set of all perfect matchings of G. By Lemma 2.6, we can construct a Pfaffian orientation $\vec{G}$ for G in polynomial time. Let $\vec{A}$ be the adjacency matrix of $\vec{G}$. Then, by Theorem 2.8, $\det(\vec{A}) = |\mathcal{P}|^2$. This determinant can be computed using standard methods in $\mathcal{O}(n^3)$ time, and thus so can $|\mathcal{P}|$. □

2.2 Hardness of Exact Counting

Although we were able to obtain a polynomial algorithm for computing the number of perfect matchings of a planar graph in Section 2.1, in this section we will motivate that such exact counting problems are generally intractable. Consider the related problem of counting the number of perfect matchings of a bipartite graph. We will prove the following result by Valiant, which considers an equivalent problem called #(0, 1)-Perm (Problem 2.3):

Theorem 2.9 (Valiant [23])**.** *#(0,1)-*Perm *is* ***#P**-complete.*

In Section 2.2.1, we recall some terminology and techniques used in the study of **NP**-complete and **#P**-complete problems. We then introduce the **#P**-complete problems used in Valiant's proof in Section 2.2.2 and give the full proof in Section 2.2.3, establishing a multi-part reduction beginning with #Exact-3-Cover and ending at #(0,1)-Perm.

2.2.1 A Primer on NP- and #P-completeness

Recall that the class **NP** contains decision problems with "yes" or "no" answers, essentially asking whether or not a *witness* w exists for a given instance I of a problem. The important characteristic of problems in **NP** is that they are *polynomially verifiable*, i.e., there exists an **NP**-predicate $\mathcal{X} : \Sigma^* \times \Sigma^* \to \{0,1\}$ such that for any instance I and candidate witness w, there exists a polynomial p such that $\mathcal{X}(I,w)$ can be computed in at most $p(|I|)$ time. Note that this implies that if I has a witness, then there exists a witness w with $|w| \leq p(|I|)$ such that $\mathcal{X}(I,w) = 1$. A more formal and complete definition of **P**, **NP**, and related classes can be found in [6].[1]

Instead of asking whether or not a witness exists for a given instance I, problems in the class **#P** ask how many witnesses there are. Here, we're dealing with function problems of the form $f : \Sigma^* \to \mathbb{N} = \{0,1,2,\ldots\}$ taking an input instance I and outputting the number of witnesses. Informally, a function problem f is in **#P** if, given a set of candidate witnesses, each witness can be checked in polynomial time. More formally, function problems in **#P** compute $f_{\mathcal{X}}(I) = |\{w : \mathcal{X}(I,w) = 1\}|$, where $\mathcal{X}$ is an **NP**-predicate.

Recall that a *polynomial transformation* (also sometimes called a *reduction*) from a decision problem Π_1 with instances $\mathcal{I}_1$ to another decision problem Π_2 with instances $\mathcal{I}_2$ is a function $\Phi : \mathcal{I}_1 \to \mathcal{I}_2$ satisfying: (i) Φ can be computed in polynomial time, and (ii) for all $I \in \mathcal{I}_1$, I has a witness if and only if $\Phi(I)$ does. If such a reduction is possible, we will write $\Pi_1 \leq \Pi_2$.

Reductions are similar for function problems.[2] A reduction Φ from problem $f_{\mathcal{X}}$ to $f_{\mathcal{Y}}$ is said to be *parsimonious* if it preserves the number of solutions, i.e., for all instances I of $f_{\mathcal{X}}$, we have $f_{\mathcal{X}}(I) = f_{\mathcal{Y}}(\Phi(I))$. Here also we require Φ to be computed in polynomial time, and write $f_{\mathcal{X}} \leq f_{\mathcal{Y}}$ if such a reduction is possible. A function problem f is said to be ***#P**-hard* if there exists another function problem $f' \in$ **#P** such that $f' \leq f$. If we also have that $f \in$ **#P**, then f is said to be ***#P**-complete.*

The standard proof by Cook [4] that Sat is **NP**-complete is parsimonious, so #Sat is **#P**-complete. Many of the usual **NP**-completeness reductions are also parsimonious, so we have:

$$\#\textsc{Sat} \leq \#3\textsc{Sat} \leq \#\textsc{Exact-3-Cover}$$

2.2.2 Some #P-complete Problems

Before we show the multi-step reduction to prove Theorem 2.9, it will be useful to be familiar with the problems we'll use. Valiant's theorem involves a variant of the *permanent* of an $n \times n$ matrix

[1] Garey and Johnson use the full notions of nondeterministic Turing machines and recognized languages when defining **NP**, where we use the looser verifier definition for simplicity.

[2] For a reduction $f_{\mathcal{X}} \leq f_{\mathcal{Y}}$, we are somewhat obscuring the line between *Karp reducibility* — which requires two polynomial-time functions ϕ, ψ such that $f_{\mathcal{X}}(x) = \psi(f_{\mathcal{Y}}(\phi(x)))$ — and *Cook/Turing reducibility*, which says $f_{\mathcal{X}}$ can be computed in polynomial time given an oracle for $f_{\mathcal{Y}}$.

A, which is defined as:

$$\operatorname{per}(A) = \sum_{\sigma \in S_n} \prod_{i \in [n]} A(i, \sigma(i)),$$

where S_n is the set of permutations of $[n] = \{0, 1, \ldots, n-1\}$. Note that this is similar to but not the same as the determinant of A, which includes the signature $sgn(\sigma)$ (see, e.g., the proof of Theorem 2.8). For our proof, we'll consider the following three permanent problems:

Problem 2.1 (#PERM)**.** *Given an $n \times n$ matrix A with integer values, what is* $\operatorname{per}(A)$*?*

Problem 2.2 (#$(0,1)$-d-PERM)**.** *Given an $n \times n$ matrix A with at most d distinct integer values other than* 0 *and* 1*, what is* $\operatorname{per}(A)$*?*

Problem 2.3 (#$(0,1)$-PERM)**.** *Given an $n \times n$ matrix A with only* 0 *and* 1 *values, what is* $\operatorname{per}(A)$*?*

As we will see, permanent problems are closely related to problems about matchings of bipartite graphs. In the following, we'll use $\mathcal{M}_G$ (resp., $\mathcal{P}_G$) to denote the set of all (perfect) matchings of a graph G. Given a set of matchings $\mathcal{M} \subseteq \mathcal{M}_G$, we define their total weight to be:

$$w(\mathcal{M}) = \sum_{M \in \mathcal{M}} w(M) = \sum_{M \in \mathcal{M}} \prod_{e \in M} w(e).$$

Note that for the definition of the following **#P** problems, we're using the product of edge weights to define the weight of a matching as opposed to a summation, which may appear more natural.

Problem 2.4 (#W-BI-MATCH)**.** *Given a bipartite graph $G = (A \cup B, E)$ with integer edge weights $w : E \to \mathbb{Z}$, what is $w(\mathcal{M}_G)$?*

Problem 2.5 (#W-BI-PER-MATCH)**.** *Given a bipartite graph $G = (A \cup B, E)$ with integer edge weights $w : E \to \mathbb{Z}$, what is $w(\mathcal{P}_G)$?*

Problem 2.6 (#BI-PER-MATCH)**.** *Given a bipartite graph $G = (A \cup B, E)$, what is $|\mathcal{P}_G|$?*

A useful observation is that #$(0,1)$-PERM is equivalent to #BI-PER-MATCH and, by a similar but generalized argument, #PERM is equivalent to #W-BI-PER-MATCH. We'll show the former informally. Let $R = \{r_1, \ldots, r_n\}$ be the rows of A and let $C = \{c_1, \ldots, c_n\}$ be its columns. Construct a bipartite graph $G = (R \cup C, E)$ where an edge (r_i, c_j) is included in E if and only if $A(i,j) = 1$; note that this is bipartite because we only allow edges between R and C. The permutations of S_n capture every possible way of mapping the row indices to the column indices in a one-to-one fashion. In the graph G, this is the same as considering every possible bipartite perfect matching, ignoring whether or not those edges are actually present. Fixing a particular $\sigma \in S_n$, its contribution to the permanent is 1 if and only if $A(i, \sigma(i)) = 1$ for all i; otherwise, the product (and thus its contribution) is 0. But $A(i, \sigma(i)) = 1$ for all i if and only if the perfect bipartite matching in G corresponding to σ has all its edges present. So the number of perfect matchings is $\operatorname{per}(A)$.

Finally, we consider #EXACT-3-COVER, which is where we'll start our multi-step reduction.

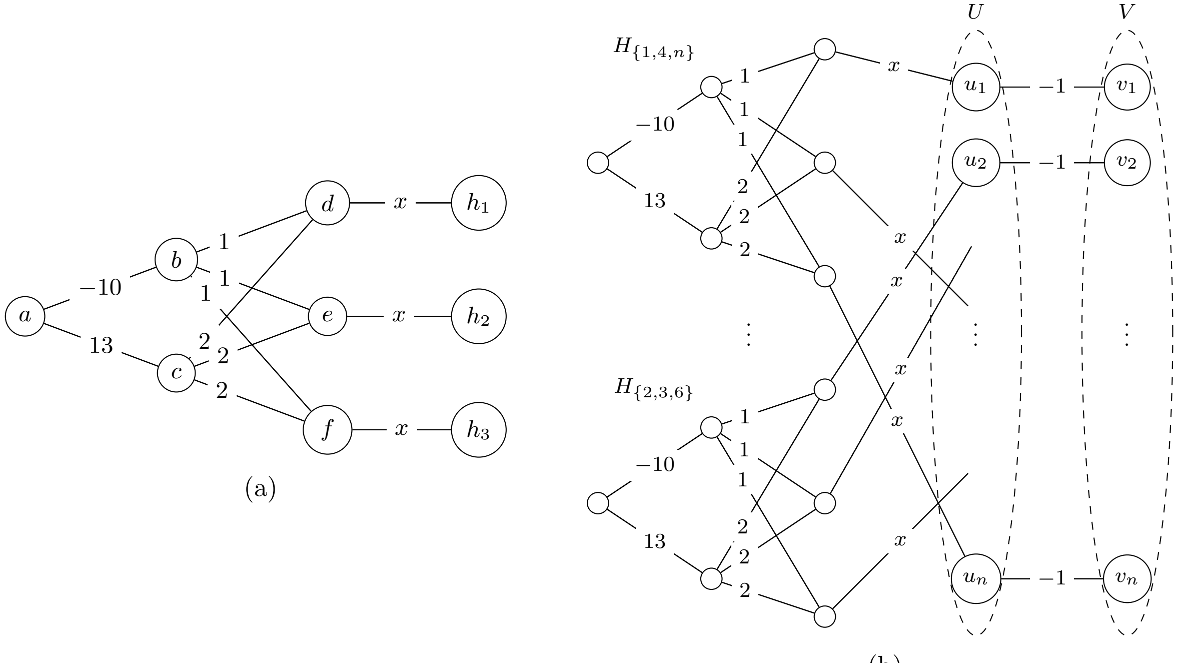
Problem 2.7 (#EXACT-3-COVER)**.** *Given a set $X = \{1, \ldots, n\}$ and a collection $Y \subseteq \binom{X}{3}$, let $\mathcal{Z}$ be the set of subcollections $Z \subseteq Y$ such that each $i \in X$ is in exactly one 3-set of Z. What is $|\mathcal{Z}|$?*

As a simple example, suppose $X = \{1, \ldots, 6\}$ and $Y = \{\{1,2,3\}, \{4,5,6\}, \{1,3,5\}, \{2,4,6\}\}$. Then there are two subcollections containing each element of X uniquely: $Z_1 = \{\{1,2,3\}, \{4,5,6\}\}$ and $Z_2 = \{\{1,3,5\}, \{2,4,6\}\}$.

2.2.3 Valiant's Proof

Recall that we're proving Theorem 2.9, which states that #(0, 1)-Perm is **#P**-complete. To show #(0, 1)-Perm is **#P**-hard, we will show the following multi-step reductions: (i) #Exact-3-Cover ≤ #W-Bi-Match ≤ #Perm, and (ii) #(0, 1)-d-Perm ≤ #(0, 1)-($d-1$)-Perm. The first reduction simply establishes that #Perm is **#P**-hard. The second reduction can then be applied iteratively to reduce #Perm — which allows for any number of non-(0, 1) distinct integer entries — to #(0, 1)-Perm, proving that #(0, 1)-Perm is **#P**-hard, as desired.

It is notable that the decision problem Bi-Per-Match, which asks whether or not an unweighted bipartite graph has a perfect matching, is in **P**. Both the Ford–Fulkerson algorithm [5] and the more efficient Hopcroft–Karp algorithm [8] find a maximal bipartite matching (and thus can answer whether or not a perfect matching exists) in polynomial time. However, changing this to the counting problem #Bi-Per-Match where we must count all perfect matchings is, by Valiant's Theorem, strictly more difficult.



(a)

(b)

Figure 1: (a) Gadget H and (b) the bipartite graph used in the proof of Lemma 2.10.

Lemma 2.10. #Exact-3-Cover *(Problem 2.7)* ≤ #W-Bi-Match *(Problem 2.4)*.

Proof. Consider the gadget H, depicted in Figure 1a. We will first show that the total weight of all matchings of H, denoted $w(\mathcal{M}_H)$, is equal to $4(x^3+1)$. Recall that:

$$w(\mathcal{M}_H) = \sum_{M \in \mathcal{M}_H} w(M) = \sum_{M \in \mathcal{M}_H} \prod_{e \in M} w(e).$$

We can group the matchings into four "cases":

1. Those which have an x^3 term in their weight must use all three x-weighted edges. So the only choice is whether to include either (a,b), (a,c), or neither, resulting in:

$$x^3 - 10x^3 + 13x^3 = 4x^3.$$

2. Those which have an x^2 term in their weight must use exactly two of the x-weighted edges; w.l.o.g. suppose that (d,h_1) and (e,h_2) are used. Then the resulting weight is:

$$x^2 - 10x^2 + 13x^2 + 1x^2 + 2x^2 + (-10\cdot 2)x^2 + (13\cdot 1)x^2 = 0.$$

   This summation would be the same regardless of which two x-weighted edges we picked, so there is no x^2 term in the summed weight $w(\mathcal{M}_H)$.

3. Those which have an x term in their weight must use exactly one of the x-weighted edges; w.l.o.g. suppose (d,h_1) is used. Then the resulting weight is:

$$x - 10x + 13x + 2(1x) + 2(2x) + 2(-10\cdot 2)x + 2(13\cdot 1)x + 2(2\cdot 1)x = 0.$$

4. Those which have a constant term in their weight must use no x-weighted edges, resulting in:

$$1\ - 10 + 13 + 3(1) + 3(2) + 3(-10\cdot 2) + 3(13\cdot 1) + 6(2\cdot 1) = 4.$$

   The 1 in the above summation is necessary to make it equal 4, but I have no idea where it comes from. Could it be the weight of the empty matching?

So we have that $w(\mathcal{M}_H) = 4(x^3+1)$.

Now consider any instance $I = (X,Y)$ of #Exact-3-Cover; recall that $X = \{1,\dots,n\}$ and $Y \subseteq \binom{X}{3}$. Let S be the number of solutions to instance I. We want to transform I into an instance of #W-Bi-Match, which is a bipartite graph $G = (U \cup V, E)$ with integer edge weights. Construct G as follows: for each $i \in X$, add u_i and v_i as vertices to U and V, respectively, and add an edge (u_i, v_i) with weight -1 to E. Then, for each 3-set $A = \{i,j,k\} \in Y$, add a copy H_A of gadget H setting $h_1 = u_i$, $h_2 = u_j$, and $h_3 = u_k$. Note that the resulting G (e.g., Figure 1b) is bipartite.

We claim that if $x = 1$ in each gadget copy H_A, then $w(\mathcal{M}_G) = 4^{|Y|}S$. If this is true, then given an oracle for #W-Bi-Match we can take an instance I of #Exact-3-Cover, transform it into an instance G of #W-Bi-Match with $x = 1$, solve #W-Bi-Match on G using the oracle, and divide the solution by $4^{|Y|}$ to obtain S, thereby also solving #Exact-3-Cover.

So it remains to prove our claim that if $x = 1$ in each gadget copy H_A, then $w(\mathcal{M}_G) = 4^{|Y|}S$. $\mathcal{M}_G$ is the set of all matchings of G; let $\mathcal{M}' \subset \mathcal{M}_G$ be the set of all matchings covering all of the u_i vertices but none of the v_i vertices.

Consider any matching $M \in \mathcal{M}_G \setminus \mathcal{M}'$, and let i be the smallest index such that such that u_i is not (or v_i is) covered in M. Consider the complementary matching $M' = M \oplus (u_i, v_i)$.[3] There are two cases: (i) if u_i was not covered in M, then neither was v_i, so $M' = M \cup \{(u_i,v_i)\}$, or (ii) if v_i was covered in M, then $(u_i,v_i) \in M$ and thus $M' = M \setminus \{(u_i,v_i)\}$. Because $w((u_i,v_i)) = -1$, we have that $w(M) = -w(M')$. Since this mapping is bijective, such pairs of M and M' always cancel each other out, so $w(\mathcal{M}_G \setminus \mathcal{M}') = 0$.

[3] The symmetric difference $\oplus$ means that if $(u_i,v_i) \in M$, then it is dropped in M', and if $(u_i,v_i) \notin M$, then it is added to M'.

Now, consider any $M \in \mathcal{M}'$; M must match every vertex u_i but none of the v_i vertices, implying that $(u_i, v_i) \notin M$ for all i. So M must only include edges from the copies of gadget H, and in particular all the u_i vertices must be matched using the gadget's x-weighted edges. We want to show that if M contributes weight to the total sum, then for each gadget copy H_A, M must either use all three x-weighted edges or none of them. This implies that each matching which contributes weight to the total sum corresponds to an exact 3-cover of X, since each gadget copy was created directly from a 3-set in Y.

In detail, let $G' \subset G$ be the subgraph containing only the x-weighted edges and their endpoints. Any matching $M \in \mathcal{M}'$ must include some partial matching of G' which covers all the u_i vertices, possibly also including some other edges from the gadget copies. Fixing such a partial matching $M' \in \mathcal{M}_{G'}$, we avoid any possibility that the gadget copies conflict at the u_i endpoints of their x-weighted edges and thus can choose what each gadget copy H_A contributes to M independently of the other copies. However, not all the matchings in $\mathcal{M}_H$ are available to us at every gadget copy. For a given gadget copy H_A, we can only pick those $M'' \in \mathcal{M}_{H_A}$ that agree with M'; that is, those in $\mathcal{M}^*_{H_A} = \{M'' \in \mathcal{M}_{H_A} : M' \cap E(H_A) = M'' \cap E(G')\}$. So,

$$w(\mathcal{M}_G) = w(\mathcal{M}_G \setminus \mathcal{M}') + w(\mathcal{M}') = \sum_{M \in \mathcal{M}'} w(M) = \sum_{M' \in \mathcal{M}_{G'}} \prod_{A \in Y} \sum_{M'' \in \mathcal{M}^*_{H_A}} w(M'').$$

For the innermost sum, we know from our initial inspection of H that if $|M'' \cap E(G')| \in \{1, 2\}$, then there is an x or x^2 term in $w(M'')$ and the total weight over all such matchings is 0. Otherwise, there is a constant or x^3 term in $w(M'')$, and the total weight over all such matchings is 4 (in the constant term case) and $4x^3 = 4$ when $x = 1$ (in the x^3 term case). Let $\mathcal{M}^*_{G'} \subset \mathcal{M}_{G'}$ be the set of matchings of G' that, for each gadget copy, either use all three x-weighted edges or none of them. Then,

$$w(\mathcal{M}_G) = \cdots = \sum_{M' \in \mathcal{M}_{G'}} \prod_{A \in Y} \sum_{M'' \in \mathcal{M}^*_{H_A}} w(M'') = \sum_{M' \in \mathcal{M}^*_{G'}} \prod_{A \in Y} 4 = \sum_{M' \in \mathcal{M}^*_{G'}} 4^{|Y|} = 4^{|Y|} S,$$

since $\mathcal{M}^*_{G'}$ corresponds exactly to the set of exact 3-covers of X using the collection Y. □

We now reduce computing the total weight of all matchings of a bipartite graph to computing the permanent, utilizing the fact we proved informally in Section 2.2.2 that #Perm is equivalent to #W-Bi-Per-Match (Problem 2.5).

Lemma 2.11. #W-Bi-Match *(Problem 2.4)* $\leq$ #Perm *(Problem 2.1).*

Proof. Consider any instance of #W-Bi-Match, that is, a bipartite graph $G = (A \cup B, E)$ with integer edge weights. Let $a = |A|$ and $b = |B|$, and let $\mathcal{M}^k_G \subseteq \mathcal{M}_G$ denote the set of matchings of G of size k. We will compute:

$$w(\mathcal{M}_G) = \sum_{k=0}^{\min\{a,b\}} \sum_{M \in \mathcal{M}^k_G} w(M).$$

To do so, fix a value of k and construct a new graph G' by adding $b - k$ new vertices to A (each adjacent only to the original vertices of B) and $a - k$ new vertices to B (each adjacent only to the original vertices of A). Each newly added edge is given weight 1. Thus, every matching

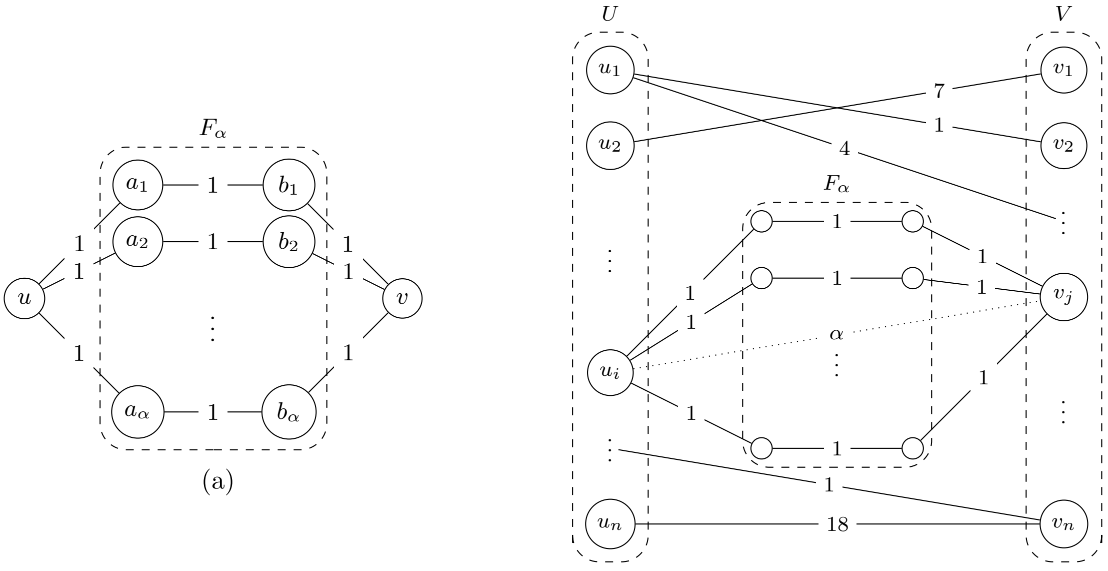


Figure 2: (a) The gadget F_α replacing an edge (u, v), and (b) the bipartite graph with one α-weighted edge (u_i, v_j) replaced by F_α as in Lemma 2.12.

$M \in \mathcal{M}_G^k$ leaves exactly $a - k$ vertices of A unmatched and $b - k$ vertices of B unmatched. So we can choose any one of $(a - k)!$ ways to match these unmatched vertices of A to the vertices of G' that were added to B, and similarly can choose any one of $(b - k)!$ ways to match the unmatched vertices of B to the vertices of G' that were added to A. So each $M \in \mathcal{M}_G^k$ corresponds to exactly $(a-k)!(b-k)!$ perfect matchings of G', and these perfect matchings each have the same weight as M since the added edges have weight 1. Therefore, given an oracle for #W-Bi-Per-Match (which is equivalent to #Perm) we can easily compute $w(\mathcal{M}_G^k)$ by dividing the result by $(a-k)!(b-k)!$. □

For the last piece of our argument, we'll once again rely on the fact we proved informally in Section 2.2.2 that #Perm is equivalent to #W-Bi-Per-Match, this time considering the versions where the matrix (or bipartite graph) only has d distinct non-$(0, 1)$ integer values (or integer edge weights).

Lemma 2.12. *#$(0,1)$-d-Perm (Problem 2.2) $\leq$ #$(0,1)$-$(d-1)$-Perm.*

Proof. Consider an instance $G = (U \cup V, E)$ of the equivalent weighted bipartite matching problem, where G has d distinct integer edge weights that are not 0 or 1. Let α be one of these non-$(0, 1)$ edge weights. We desire an instance G' of the weighted bipartite matching problem with only $d-1$ distinct non-$(0, 1)$ integer edge weights; construct G' by replacing every edge with weight α in G with the gadget F_α, as in Figure 2. The problem is now to compute:

$$w(\mathcal{P}_G) = \sum_{P \in \mathcal{P}_G} w(P) = \sum_{P \in \mathcal{P}_G} \prod_{e \in P} w(e),$$

given an oracle that computes $w(\mathcal{P}_{G'})$.

Consider any perfect matching $P \in \mathcal{P}_G$, and consider any edge (u, v) that had weight α in G. If $(u, v) \notin P$, then P matched u and v using some other edges in G. Thus, P already forms a perfect matching of $G' - F_\alpha$. But the only way to form a perfect matching for F_α without using u and v is to take all the "middle" edges (a_i, b_i). So P corresponds exactly to a perfect matching $P' = P \cup \{(a_i, b_i) : i \in \{1, \ldots, \alpha\}\} \in \mathcal{P}_{G'}$. Moreover, since all the middle edges have weight 1, $w(P) = w(P')$.

Now suppose $(u, v) \in P$. To form a corresponding perfect matching $P' \in \mathcal{P}_{G'}$, we have to construct a perfect matching of F_α which also matches u and v. It is easy to see that the only way to do this is to use a pair of edges (u, a_i) and (b_i, v) for some i, and then take all the middle edges (a_j, b_j) for all $j \neq i$. The weight of the resulting perfect matching P' is $w(P') = w(P)/\alpha$, since P includes the α-weighted edge (u, v) while P' uses many 1-weighted edges in its place. But there are α possible perfect matchings, since there are α choices of pairs of edges to match u and v within F_α. Thus, their contribution to the total weight of perfect matchings is $\alpha w(P') = w(P)$.

Therefore, in either case, $w(\mathcal{P}_G) = w(\mathcal{P}_{G'})$, so given an oracle that computes $w(\mathcal{P}_{G'})$, we immediately solve our original problem as well. □

We can now formally prove Valiant's Theorem.

Proof of Theorem 2.9. We know that #Exact-3-Cover is **#P**-complete because the standard proofs reducing Sat $\leq$ 3Sat $\leq$ Exact-3-Cover are parsimonious. By Lemmas 2.10 and 2.11, we have that #Exact-3-Cover $\leq$ #W-Bi-Match $\leq$ #Perm. By iteratively applying Lemma 2.12, we can reduce #Perm from an problem allowing any number of non-$(0, 1)$ entries in its instances to #$(0, 1)$-Perm, which only allows 0 and 1 entries. Thus, we have that #$(0, 1)$-Perm is **#P**-hard.

Furthermore, given any witness for #$(0, 1)$-Perm — say, a permutation σ of $[n]$ mapping its rows to its columns — it can easily be verified in n operations whether $\prod_{i \in [n]} A(i, \sigma(i))$ is 0 or 1. Thus, #$(0, 1)$-Perm $\in$ **#P**, and so we conclude that #$(0, 1)$-Perm is **#P**-complete. □

2.3 Approximate Counting and Uniform Sampling

We saw in Section 2.2 that many counting problems are **#P**-complete, leaving little hope for efficient algorithms that produce exact outputs. The next best outcome we could aim for are efficient algorithms for *approximate counting.* We will see that there is a tight relationship between approximate counting and *random sampling*, though we will largely ignore the formal proof of their equivalence and focus instead on concrete examples.

One tool we will use often are *Chernoff bounds.* This is a very powerful tool for analyzing a sum of independent random variables; in essence, it can be used to understand how unlikely it is that a random variable is far from its expectation. We will use the following simplified bound, but stronger bounds and proofs can be found in, e.g., [17].

Theorem 2.13 (Chernoff)**.** *Let $X_1, X_2, \ldots, X_n$ be independent, identically distributed random variables in $\{0, 1\}$, and let $p = \mathrm{E}[X_i]$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathrm{E}[X] = np$. Then, for any $\varepsilon \in (0, 1]$,*

$$\Pr[|X - \mu| > \varepsilon\mu] < 2\exp(-\varepsilon^2\mu/3).$$

2.3.1 Approximate Counting (FPRAS)

As a motivating example, we saw that #$(0, 1)$-Perm (Problem 2.3) is **#P**-complete (Theorem 2.9). Consider an instance of an approximate counting version of #$(0, 1)$-Perm that relaxes the requirements of exact counting. Given an $n \times n$ matrix A with only 0 and 1 values and an error parameter

ε, compute a value P such that:

$$(1-\varepsilon)\mathrm{per}(A) \leq P \leq (1+\varepsilon)\mathrm{per}(A).$$

The value P gets us "close" to the exact value of $\mathrm{per}(A)$, where ε controls how accurate "close" really is. But this approximate counting version says nothing about how long we have to find a suitable P, nor does it allow any small probability of failing to find such a P. When all these considerations are taken together, we obtain the following:

Definition 2.14. *Let $f : \Sigma^* \to \mathbb{N}$ be a function (counting) problem, $x \in \Sigma^*$ be an input, $\varepsilon > 0$ be an error parameter, and $0 < \delta < 1$ be a confidence parameter. A fully polynomial randomized approximation scheme (FPRAS) is a randomized algorithm that computes $y \in \mathbb{N}$ in time polynomial in $|x|$, $1/\varepsilon$, and $\ln(1/\delta)$ such that:*

$$\Pr\left[(1-\varepsilon)f(x) \leq y \leq (1+\varepsilon)f(x)\right] \geq 1-\delta.$$

When designing an FPRAS, we can make things easier by first only considering $\delta = 1/4$, meaning that our algorithm only needs to find a suitable output with probability $3/4$. We can then use a technique called *boosting* (Algorithm 1) to amplify this success probability up to $1-\delta$, for arbitrary $\delta \in (0,1)$.

Algorithm 1 Success Probability Boosting

```
Let $f : \Sigma^* \to \mathbb{N}$ be a function (counting) problem, $x \in \Sigma^*$ be an input, $\varepsilon > 0$ be an error parameter, and $0 < \delta < 1$ be a confidence parameter.
Let $\mathcal{A}$ be an algorithm which produces output $y \in \mathbb{N}$ such that $(1-\varepsilon)f(x) \leq y \leq (1+\varepsilon)f(x)$ with probability $3/4$.
for $i \in \{1, 2, \ldots, k = 36\ln(2/\delta)\}$ do
    Run $\mathcal{A}$ and obtain output $y_i$.
return Median$(y_1, y_2, \ldots, y_k)$.
```

To analyze the correctness of this boosting technique, consider the outputs $y_1, y_2, \ldots, y_k$ of algorithm $\mathcal{A}$ and define indicator variables X_i as follows:

$$X_i = \begin{cases} 1 & \text{if } y_i \in [(1-\varepsilon)f(x), (1+\varepsilon)f(x)], \\ 0 & \text{otherwise.} \end{cases}$$

Let $X = \sum_{i=1}^{k} X_i$. Since $\mathcal{A}$ produces an output y_i in the desired range with probability $3/4$ in time polynomial in $|x|$ and $1/\varepsilon$, we have $\mathrm{E}\left[X_i\right] = 3/4$, and thus $\mathrm{E}\left[X\right] = \mathrm{E}\left[\sum_{i=1}^{k} X_i\right] = \sum_{i=1}^{k} \mathrm{E}\left[X_i\right] = 3k/4$. Finally, note that the median of $y_1, y_2, \ldots, y_k$, say z, is in the desired range if and only if at least half of the y_i's are; i.e., if $X \geq k/2$. By this and Chernoff bounds (Theorem 2.13),

$$\begin{aligned}
\Pr\left[\text{boosting fails}\right] &= \Pr\left[z \notin [(1-\varepsilon)f(x), (1+\varepsilon)f(x)]\right] \\
&= \Pr\left[X < k/2\right] \\
&= \Pr\left[\left|X - \frac{3k}{4}\right| > \frac{k}{4} = \frac{1}{3} \cdot \frac{3k}{4}\right] \\
&= \Pr\left[|X - \mathrm{E}\left[X\right]| > (1/3)\mathrm{E}\left[X\right]\right] \\
&< 2\exp\left(-\frac{(1/3)^2\,(3k/4)}{3}\right) \\
&= 2e^{-k/36}
\end{aligned}$$

Recall from Algorithm 1 that $k = 36\ln(2/\delta)$. Plugging this in, we have:

$$\Pr\left[\text{boosting fails}\right] < 2e^{-k/36} = 2e^{-36\ln(2/\delta)/36} = \delta,$$

and thus the probability that boosting successfully outputs a value within the desired error tolerance is at least $1-\delta$, as desired. Algorithm 1 runs in $\mathcal{O}(\ln(1/\delta))$ time, so together with any algorithm $\mathcal{A}$ that obtains an output in the desired range with probability (at least) 3/4, we obtain an FPRAS.

2.3.2 Uniform Sampling (FPAUS)

Unlike in counting problems, whose goal is to output the size of a set or the total number of witnesses to some predicate, *sampling problems* seek to output elements from a set according to some probability distribution. A lottery is a good example. The instance of our sampling problem describes the lottery, and we wish to output a single winner. In order to be a fair lottery, we'd want to choose the winner according to the uniform distribution. Ensuring this adherence to the uniform distribution becomes the responsibility of the sampling algorithm. Other sampling problems may target different distributions (Gaussian, Gibbs, etc.).

As with approximate counting, here we will settle for an approximation to the desired distribution. To measure how similar two distributions are, we'll use the following.

Definition 2.15. *The total variation distance between two probability distributions μ and π on a space Ω is given by:*

$$d_{TV}(\mu, \pi) = \frac{1}{2}\sum_{x \in \Omega} |\mu(x) - \pi(x)| = \max_{A \subseteq \Omega} |\mu(A) - \pi(A)|.$$

Note that total variation distance is half the L_1 (Manhattan) distance. We are now ready to formalize what we mean by an approximate sampler.

Definition 2.16. *Let f be a sampling problem over a space Ω with target distribution π and input $x \in \Sigma^*$. Let $\delta > 0$ be a difference parameter. A fully polynomial almost uniform sampler (FPAUS) is an algorithm that generates solutions from a distribution μ on Ω in time polynomial in $|x|$ and $\ln(1/\delta)$ such that:*

$$d_{TV}(\mu, \pi) \leq \delta.$$

$$\begin{tikzcd}
\boxed{\text{Exact Counting}} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \boxed{\text{Exact Sampling}} \arrow[d, Rightarrow] \\
\boxed{\text{Approx. Counting (FPRAS)}} \arrow[r, Leftrightarrow] & \boxed{\text{Approx. Sampling (FPAUS)}}
\end{tikzcd}$$

Figure 3: Relationships among exact and approximate counting and sampling problems.

As we've previously hinted at, counting and sampling are closely related. In particular, an FPRAS and FPAUS are equivalent for "self-reducible" problems [12]. The full set of relationships are shown in Figure 3, where a directed edge from $\mathcal{P}$ to $\mathcal{Q}$ means that if you can do $\mathcal{P}$, then you can do $\mathcal{Q}$.

Instead of defining "self-reducible" formally and proving these relationships in general, we'll examine a particular self-reducible problem: counting/sampling from the matchings $\mathcal{M}_G$ of a graph G. It is easy to see that if one can do exact counting (resp., exact sampling), one can also do approximate counting (resp., approximate sampling). So we begin our analysis with a straightforward reduction from exact counting to exact sampling.

Lemma 2.17. *Given an algorithm $\mathcal{A}$ that computes $|\mathcal{M}_G|$ for an arbitrary graph $G = (V, E)$ in time polynomial in $|V|$, one can construct an algorithm that samples a matching $M \in \mathcal{M}_G$ uniformly at random in time polynomial in $|V|$.*

Proof. Begin by choosing edge $e_1 = (u_1, v_1) \in E$. Let $G_1 = (V, E \setminus \{e_1\})$ and $G'_1 = G[V \setminus \{u_1, v_1\}]$, i.e., the induced subgraph on vertices other than u_1 and v_1. For any matching $M \in \mathcal{M}_G$, there are two cases: either $e_1 \notin M$ and thus M is also a matching of G_1, or $e_1 \in M$ and thus $M \setminus \{e_1\}$ is a matching of G'_1. Thus, for a matching $M \in \mathcal{M}_G$ chosen uniformly at random:

$$\Pr\left[e_1 \in M\right] = \frac{|\mathcal{M}_{G'_1}|}{|\mathcal{M}_{G_1}| + |\mathcal{M}_{G'_1}|} = \frac{|\mathcal{M}_{G'_1}|}{|\mathcal{M}_G|}.$$

So we have the correct probability of including e_1 in our randomly sampled matching M. Now we want the correct probability of including edge e_2 in M conditioned on whether or not e_1 is in M.

More generally, we would like to know the probability of including edge e_i in M conditioned on whether or not each e_j for $j \in \{1, \ldots, i-1\}$ was included in M. Let b_i be an indicator variable equal to 1 if $e_i \in M$ and 0 otherwise. Let $G_{b_1 \cdots b_i}$ be the subgraph of G generated by iteratively either removing edge e_i if $b_i = 0$ or replacing the current subgraph with $G_{b_1 \cdots b_{i-1}}[V_{i-1} \setminus \{u_i, v_i\}]$ if $b_i = 1$. Then:

$$\Pr\left[e_i \in M | e_1, \ldots, e_{i-1} \in^? M\right] = \Pr\left[b_i = 1 | b_1 \cdots b_{i-1}\right] = \frac{|\mathcal{M}_{G_{b_1 \cdots b_{i-1} 1}}|}{|\mathcal{M}_{G_{b_1 \cdots b_{i-1}}}|}, \text{ and}$$
$$\Pr\left[e_i \notin M | e_1, \ldots, e_{i-1} \in^? M\right] = \Pr\left[b_i = 0 | b_1 \cdots b_{i-1}\right] = \frac{|\mathcal{M}_{G_{b_1 \cdots b_{i-1} 0}}|}{|\mathcal{M}_{G_{b_1 \cdots b_{i-1}}}|}$$

The above probabilities hold for any matching, so consider a specific matching $M \in \mathcal{M}_G$, and let $b_1 \cdots b_{|E|}$ be the indicator variables corresponding to the edges included in M. Then,

$$\Pr\left[M\right] = \Pr\left[b_1 \cdots b_{|E|}\right] = \frac{|\mathcal{M}_{G_{b_1}}|}{|\mathcal{M}_G|} \cdot \frac{|\mathcal{M}_{G_{b_1 b_2}}|}{|\mathcal{M}_{G_{b_1}}|} \cdots \frac{|\mathcal{M}_{G_{b_1 \cdots b_{|E|}}}|}{|\mathcal{M}_{G_{b_1 \cdots b_{|E|-1}}}|} = \frac{|\mathcal{M}_{G_{b_{|E|}}}|}{|\mathcal{M}_G|} = \frac{1}{|\mathcal{M}_G|},$$

and thus M is chosen uniformly at random from among all possible matchings of G.

Each (conditional) probability can be calculated explicitly using two calls to the exact counting algorithm $\mathcal{A}$ (one for the numerator, and one for the denominator). There are $|E|$ such probabilities, so the total time to sample uniformly at random is $\mathcal{O}(|E|) = \mathcal{O}(|V|^2)$, which is polynomial in $|V|$. □

Next, we reduce from approximate counting to almost uniform sampling.

Lemma 2.18. *Given an FPRAS $\mathcal{F}$ that produces a value X such that $(1-\varepsilon)|\mathcal{M}_G| \leq X \leq (1+\varepsilon)|\mathcal{M}_G|$ for an arbitrary graph $G = (V, E)$ with probability at least $1-\delta$ in time polynomial in $|V|$ and $1/\varepsilon$, one can construct an FPAUS that samples a matching $M \in \mathcal{M}_G$ arbitrarily close to uniformly at random.*

Proof. We use the same approach as in Lemma 2.17, but this time must analyze the effects of using the FPRAS $\mathcal{F}$ instead of an exact counter. Each time we compute a conditional probability of whether or not to include an edge e_i in the sampled matching — i.e., when we are computing $\Pr\left[e_i \in M | e_1, \ldots, e_{i-1} \in^? M\right]$ and $\Pr\left[e_i \notin M | e_1, \ldots, e_{i-1} \in^? M\right]$ — we must make three calls to $\mathcal{F}$: one to estimate $|\mathcal{M}_{G_{b_1 \cdots b_{i-1}}}|$, another to estimate $|\mathcal{M}_{G_{b_1 \cdots b_{i-1} 0}}|$, and a third to estimate $|\mathcal{M}_{G_{b_1 \cdots b_{i-1} 1}}|$. Because $\mathcal{F}$ is a FPRAS, each estimate is within a $(1 \pm \varepsilon)$ multiplicative factor of the correct value, with probability at least $1-\delta$. Thus, the error incurred for each (conditional) probability calculation is at most:

$$\frac{1+\varepsilon}{1-\varepsilon} \leq (1+2\varepsilon)^2 \leq e^{4\varepsilon}.$$

We calculate at most $|E|$ conditional probabilities, so the total multiplicative error we could incur is at most $e^{4\varepsilon|E|} \leq e^{4\varepsilon|V|^2}$. TODO: details for success and rejection.

It remains to bound the failure probability. As argued before, we make at most $3|E| \leq 3|V|^2$ calls to $\mathcal{F}$, each of which fail to produce a good approximate count with probability at most δ. Thus, the probability that the entire algorithm fails is at most $3|V|^2\delta$. TODO: details for total variation distance. □

3 Introduction to Markov Chains

The remainder of this course is devoted to the study of Markov chains, techniques for analyzing them, and applications in sampling, counting, and related problems. In this section, we begin with the motivation, terminology, and important results related to discrete Markov chains. In Section 3.1, we give four example Markov chains and examine their properties. Finally, in Section 3.2, we introduce the notion used to understand a Markov chain's running time.

Given a (usually huge) finite, discrete state space Ω, the goal is to sample $x \in \Omega$ proportional to some weight $w(x)$. More formally, we want to sample $x \in \Omega$ with probability $\pi(x) = w(x)/Z$, where $Z = \sum_{y \in \Omega} w(y)$ is a value called the *normalizing constant* or *partition function.*[4] As we will see, this is a versatile and powerful framework to consider many problems in, including:

- *Approximate counting.* We already saw this application in Section 2.3.2.

[4] Essentially, the normalizing constant Z is only used to make the resulting distribution π a valid probability distribution: $\pi(x) \in [0, 1]$ for all $x \in \Omega$, and $\sum_{x \in \Omega} \pi(x) = 1$.

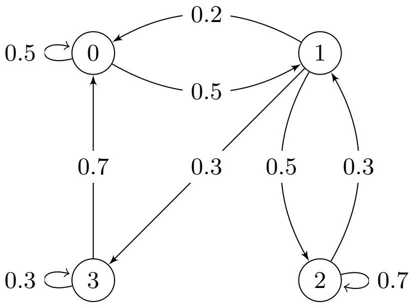


Figure 4: A simple Markov chain. The nodes of the graph represent the chain's states, and the directed edges show the chain's transitions weighted by their probabilities.

- *Sampling representatives.* For example, some researchers once observed that all samples they obtained from the set of 3-regular graphs were Hamiltonian. This inspired them to investigate further, and it is now proven that the proportion of n-vertex 3-regular graphs that contain Hamiltonian cycles tends to 1 as n approaches ∞ [19].
- *Generating data.* Humans are bad at distinguishing what is random from what is not, and are even worse at generating randomness themselves. Markov chains can be used to generate random test data for use in quality assurance applications.
- Many others, including *approximate volume/integration calculations*, *combinatorial optimization*, *statistical inference (machine learning)*, etc.

For our first Markov chain, we'll keep a simple running example and introduce relevant terminology along the way. A Markov chain has the form $\mathcal{M} = (\Omega, P)$, where Ω is a finite, discrete state space and P is an $|\Omega| \times |\Omega|$ *transition matrix*, where $P(x, y)$ is the probability of going from state $x \in \Omega$ to state $y \in \Omega$ in one step. For example, if we had $\Omega = \{0, 1, 2, 3\}$, we may have the following transition matrix:

$$P = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0.2 & 0 & 0.5 & 0.3 \\ 0 & 0.3 & 0.7 & 0 \\ 0.7 & 0 & 0 & 0.3 \end{bmatrix}$$

We could express this chain graphically if we wanted, where the states of Ω are nodes and a transition from x to y is a directed edge from x to y weighted with the probability $P(x, y)$, as in Figure 4. Observe that each row of P (above) sums to 1; more formally, we have that $\sum_{y \in \Omega} P(x, y) = 1$. In the graphical representation, this means that the weights of all edges leaving any given node also sum to 1. This property is called *stochasticity.*

Now, let $X_t \in \Omega$ be the random variable representing the state of our chain $\mathcal{M}$ at time $t \in \mathbb{N}$. We already saw that the transition matrix P gives us the probability of moving from a state $x \in \Omega$ to a neighboring state $y \in \Omega$ in a single step (i.e., P defines the *one-step probabilities*). Formally, we have the following:

$$\Pr\left[X_{t+1} = y | X_t = x\right] = P(x, y).$$

With this notation, we can define what it means when we say a Markov chain is *memoryless.* Informally, this means that a Markov chain only depends on its current state X_t when calculating X_{t+1}; the previous history of states that it visited does not have any bearing on this calculation.

Definition 3.1. *A chain $\mathcal{M} = (\Omega, P)$ is* <u>memoryless</u> *(i.e., it satisfies the* <u>Markovian Property</u>*) if:*

$$\Pr\left[X_{t+1} = y | X_0 = x_0 \wedge X_1 = x_1 \wedge \cdots \wedge X_t = x_t\right] = \Pr\left[X_{t+1} = y | X_t = x_t\right].$$

The probability of going from a state $x \in \Omega$ to a state $y \in \Omega$ in exactly T steps is called the *T-step probability*, given by:

$$P^T(x,y) = \begin{cases} P(x,y) & \text{if } T = 1; \\ \sum_{z\in\Omega} P(x,z)P^{T-1}(z,y) & \text{if } T > 1. \end{cases}$$

Let's derive this for ourselves for better understanding (see the calculation below). We want to calculate the probability that, beginning in a state $x \in \Omega$ at time t, we are in a state $y \in \Omega$ exactly T steps later. We don't know this probability directly, so we break the ways of getting to y in exactly T steps into all the ways of going from x to z in one step, and then from z to y in $T-1$ steps. The former are simply the one-step probabilities from x to z. But the latter are $(T-1)$-step probabilities, which lead to a recursive definition since we don't know these values directly either.

$$\begin{aligned} \Pr\left[X_{t+T} = y | X_t = x\right] &= \sum_{z\in\Omega} \Pr\left[X_{t+1} = z | X_t = x\right] \Pr\left[X_{t+T} = y | X_{t+1} = z\right] \\ &= \sum_{z\in\Omega} P(x,z)P^{T-1}(z,y) \\ &= P^T(x,y) \end{aligned}$$

This recursive construction shows that explicitly calculating the T-step transition matrix can be done by matrix multiplying P by itself T times. Using our example P from above, we can calculate the 2-step and 20-step probabilities using matrix multiplication:

$$P^2 = \begin{bmatrix} 0.35 & 0.25 & 0.25 & 0.15 \\ 0.31 & 0.25 & 0.35 & 0.09 \\ 0.06 & 0.21 & 0.64 & 0.09 \\ 0.56 & 0.35 & 0 & 0.09 \end{bmatrix}$$

$$P^{20} = \begin{bmatrix} 0.24423 & 0.24419 & 0.40693 & 0.10466 \\ 0.24420 & 0.24419 & 0.40696 & 0.10465 \\ 0.24414 & 0.24418 & 0.40704 & 0.10464 \\ 0.24427 & 0.24420 & 0.40687 & 0.10466 \end{bmatrix}$$

Notice that, as T gets larger, all the rows begin to look more and more similar to one another. This means that if this chain is run sufficiently long, the probability that it is in a state $y \in \Omega = \{0,1,2,3\}$ conforms to a distribution π (below), regardless of what the initial state x was.

$$\pi(x) \approx \begin{bmatrix} 0.24420 & 0.24419 & 0.40694 & 0.10465 \end{bmatrix}.$$

We say this Markov chain has converged to a *stationary distribution*, which we now define generally.

Definition 3.2. *Consider a Markov chain $\mathcal{M} = (\Omega, P)$. A distribution π over the state space Ω is said to be the* <u>stationary distribution</u> *of $\mathcal{M}$ if it satisfies $\pi = \pi P$, i.e.,*

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

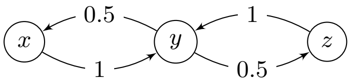


Figure 5: A Markov chain with period 2.

Not all Markov chains are guaranteed to have a unique stationary distribution; some have none at all and some have multiple. However, if we want to achieve our goal of using Markov chains to sample from state spaces Ω according to a desired target distribution π, then it will be advantageous to study chains that converge to a unique stationary distribution. To classify such chains, we need some additional terminology.

Definition 3.3. *A Markov chain $\mathcal{M} = (\Omega, P)$ is irreducible if $\forall x, y \in \Omega,\ \exists t : P^t(x, y) > 0$.*

Irreducibility means that it is possible to go from any one state x to any other state y using however many steps are needed. This particular transition may not be very likely, depending on P, but irreducibility requires that it is possible. In the graph representation of an irreducible chain, the resulting graph forms a single connected component.

Definition 3.4. *A Markov chain $\mathcal{M} = (\Omega, P)$ is aperiodic if $\forall x \in \Omega,\ \gcd\{t : P^t(x, x) > 0\} = 1$.*

An aperiodic chain avoids multiple periodic loops from a state x back to itself. For example, consider the chain in Figure 5. Notice that $P^2(x, x) > 0$ since $x \to y \to x$ is a valid path. But we also have that $P^4(x, x) > 0$ since $x \to y \to z \to y \to x$ is also valid. So this chain has a period of $\gcd\{2, 4\} = 2$. If we wanted to make this chain aperiodic, we could simply add a self-loop on x by changing P slightly so that $P(x, x) > 0$. This would make its period $\gcd\{1, 2, 4\} = 1$. In general, we will often design Markov chains with these self-loops to ensure they are aperiodic.

Definition 3.5. *A Markov chain $\mathcal{M} = (\Omega, P)$ is ergodic if $\exists t : \forall x, y \in \Omega,\ P^t(x, y) > 0$.*

The definition of ergodicity is very similar to that of irreducibility, but with the order of the $\exists, \forall$ quantifiers switched. If a Markov chain is ergodic, then there exists a $t \in \mathbb{N}$ such that the graph resulting from using P^t for the edges (instead of just P) is fully connected. This somewhat cumbersome definition is made easier by the fact that a Markov chain is ergodic if and only if it is both irreducible and aperiodic (see, e.g., [16]). This brings us to the following theorem.

Theorem 3.6 (Fundamental Theorem of Markov Chains)**.** *If a Markov chain $\mathcal{M} = (\Omega, P)$ is finite and ergodic, then it has a unique stationary distribution π such that:*

$$\forall x, y \in \Omega,\ \lim_{t \to \infty} P^t(x, y) = \pi(y).$$

So, by Theorem 3.6, we know that if we design our Markov chains carefully to be both irreducible and aperiodic, they will converge to a unique stationary distribution π. But how do we find π? Generally speaking, we need to perform Gaussian elimination on P, which is a massive $|\Omega| \times |\Omega|$ matrix. This is clearly infeasible for any meaningfully large state space Ω, but we can easily calculate π without doing this general process for some important special cases.

Proposition 3.7. *If the transition matrix P of a Markov chain $\mathcal{M} = (\Omega, P)$ is symmetric (i.e., $P(x, y) = P(y, x)$ for all $x, y \in \Omega$), then the stationary distribution π is uniform over Ω.*

Proof. Let $n = |\Omega|$. We claimed that π is uniform over Ω; i.e., $\pi(x) = 1/n$ for all $x \in \Omega$. To show π is a stationary distribution, we must verify that $\pi = \pi P$:

$$[\pi P](x) = \sum_{y \in \Omega} \pi(y)P(y,x) = \frac{1}{n}\sum_{y \in \Omega} P(y,x) = \frac{1}{n}\sum_{y \in \Omega} P(x,y) = \frac{1}{n} \cdot 1 = \frac{1}{n} = \pi(x).$$

The third equality follows from the assumption that P is symmetric, and the fourth equality follows from the fact that P is stochastic; i.e., the values in a row of P always sum to 1. □

Definition 3.8. *A Markov chain $\mathcal{M} = (\Omega, P)$ is time-reversible with respect to a distribution π if it satisfies the detailed balance condition:*

$$\forall x, y \in \Omega,\ \pi(x)P(x,y) = \pi(y)P(y,x).$$

Proposition 3.9. *If a Markov chain $\mathcal{M} = (\Omega, P)$ is time-reversible with respect to a distribution π, then π is a stationary distribution over Ω.*

Proof. As in Proposition 3.7, we must verify that $\pi = \pi P$.

$$[\pi P](x) = \sum_{y \in \Omega} \pi(y)P(y,x) = \sum_{y \in \Omega} \pi(x)P(x,y) = \pi(x)\sum_{y \in \Omega} P(x,y) = \pi(x) \cdot 1 = \pi(x).$$

The second equality follows from the assumption that $\mathcal{M}$ is time-reversible with respect to π, and the fourth equality follows from the fact that P is stochastic. □

3.1 Example Markov Chains

To make this introduction to Markov chains more concrete, we'll look at a few example chains, using the above techniques and definitions (ergodicity, symmetry, time-reversibility, etc.) to find their stationary distributions.

3.1.1 Random Walks on Regular Graphs

A connected, undirected graph $G = (V, E)$ is *d-regular* if every node in v has exactly d edges incident to it. Let $N(u) \subseteq V$ denote the neighbors of a node $u \in V$. A *random walk* is a very simple process: starting at a given node u, choose a neighboring node v uniformly at random among $N(u)$; then, go to v and repeat. When formulating this as a Markov chain, we have that the state space Ω should be all the nodes in V. The transition matrix P should have the form $P(u,v) = 1/|N(u)|$ for every edge $e = \{u, v\} \in E$, and all other entries of P should be 0.

Now, since the graph G is d-regular, every node has the same number of neighbors d. Thus, we have $P(u,v) = 1/d = P(v,u)$. So P is symmetric, and by Proposition 3.7, we have that the stationary distribution of this random walk is $\pi(u) = 1/|V|$, for all $u \in V$. This intuitively makes sense; since a random walk does not favor any one node over any other, the Markov chain should be equally likely to be at any node at stationarity.

Algorithm 2 Markov Chain for Sampling Random Matchings of Graph $G = (V, E)$

```
   Let $X_t \in \Omega$ denote the matching at time $t$. Repeat:
1: Choose an edge $e \in E$ uniformly at random.
2: Let $X' \leftarrow X_t \oplus e$; i.e., set $X' \leftarrow X_t \cup \{e\}$ if $e \notin X_t$ and set $X' \leftarrow X_t \setminus \{e\}$ otherwise.
3: if $X' \in \Omega$ (i.e., $X'$ is a matching) then
4:     With probability $1/2$, set $X_{t+1} \leftarrow X'$.
5: else Set $X_{t+1} \leftarrow X_t$.
```

3.1.2 Sampling Random Matchings

Recall that a matching of a graph $G = (V, E)$ is a set of edges $M \subseteq E$ such that no two edges in M share an endpoint. Let Ω be the set of all possible matchings of a graph G. We would like to sample random matchings from Ω; the Markov chain described in Algorithm 2 achieves this goal.

Before analyzing this Markov chain, we'll spend some time getting an intuition about it. X_t denotes the state of the chain at time t; in this chain, the states are matchings of G. In Step 1, an edge of G is chosen uniformly at random. This can be any edge in the whole graph; all are equally likely to be chosen. Then, Step 2 has two cases: when e is already part of the matching X_t, and when it is not. In the former case, e is dropped from X_t to form a new set of edges, X'. In the latter, e is added to X_t to form X'. The notation in Step 2 succinctly captures these two cases as $X' \leftarrow X_t \oplus e$, where $\oplus$ is the *symmetric difference.*

Next, Step 3 checks whether or not X' is a matching. In the case that e was not part of the matching X_t but does share one or both endpoints with other edges in X_t, adding e to form X' would not result in a valid matching. So, in this case, $X' \notin \Omega$ and the chain should not transition to it. If X' is a matching, Step 4 flips a coin: if heads, then the chain transitions to X'; otherwise, it stays at X_t. Thus, even if X' is a matching, the chain may be "lazy" and remain in the same state. The reason for this will become clear when we analyze this chain.

Example. Consider the graph $G = (V, E)$ depicted in Figure 6a. There are eleven possible matchings of G, including the empty matching $\emptyset$:

$$\Omega = \{\emptyset, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_1, e_5\}, \{e_2, e_5\}, \{e_2, e_6\}, \{e_4, e_6\}\}.$$

These are the states of our Markov chain, depicted in Figure 6b. As an aside, notice that even for small examples like this one, the corresponding Markov chain becomes very large. To explicitly obtain the transition probabilities P, we'll use Algorithm 2. Suppose we start at the empty matching $\emptyset$, and we want to transition to one of the singleton matchings, say $\{e_i\}$. To make this transition, Step 1 would need to choose e_i from among the six edges; this happens with probability $1/|E| = 1/6$. After setting $X' \leftarrow \emptyset \oplus \{e_i\} = \{e_i\}$ in Step 2, Step 3 finds that $X' = \{e_i\}$ is indeed a valid matching. We transition to X' with probability $1/2$ in Step 4. So, all together, we have:

$$P(\emptyset, \{e_i\}) = \frac{1}{6} \cdot \frac{1}{2} = \frac{1}{12}$$

The remaining transition probabilities depicted in Figure 6b can be calculated using a similar approach. Notably, the self-loops collect the probability that (i) the chosen edge in Step 1 does not yield a valid matching in Step 3, and (ii) Step 4 is "lazy" and stays at the same state.

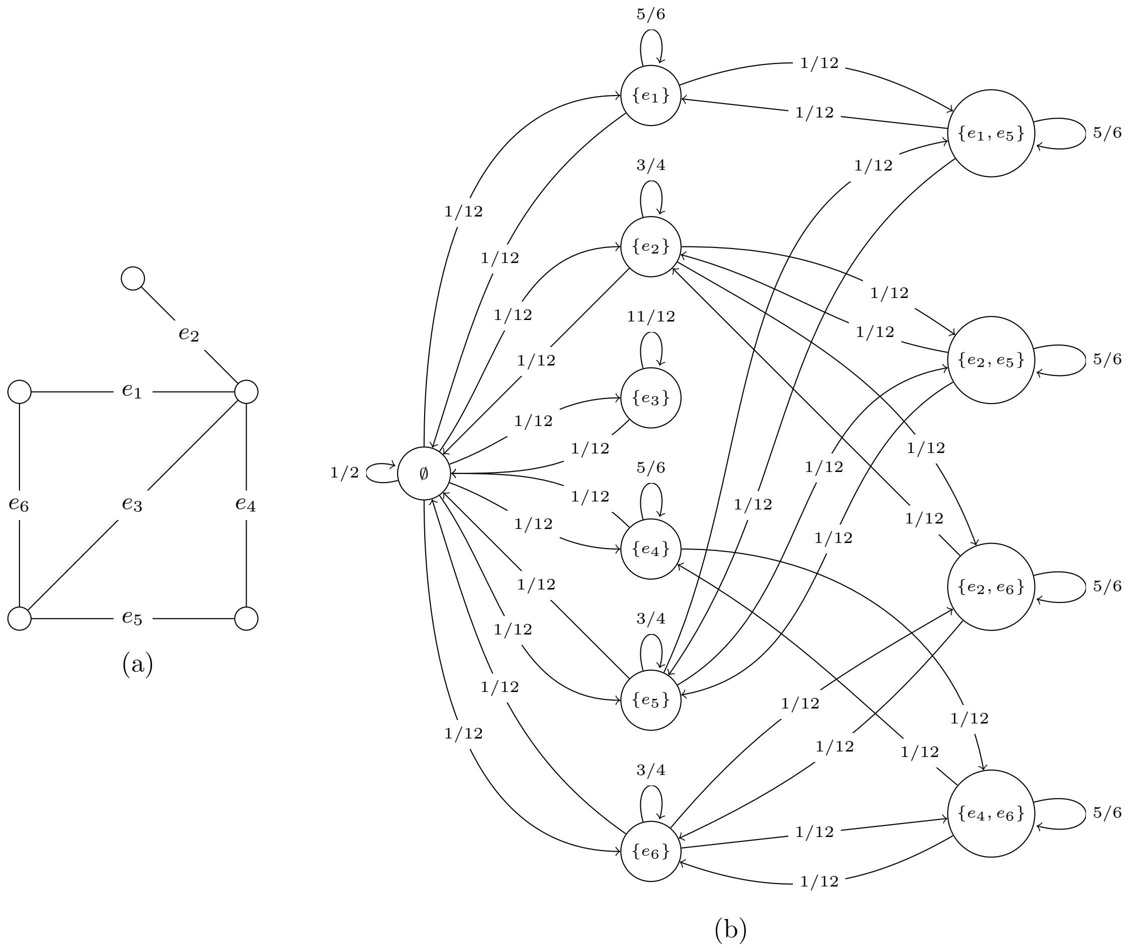


Figure 6: (a) An undirected graph $G = (V, E)$ with labeled edges. (b) The Markov chain over all matchings of G, where the transition probabilities are defined by Algorithm 2.

Stationary Distribution. To find the stationary distribution of the Markov chain defined by Algorithm 2, we first show that it is ergodic (i.e., irreducible and aperiodic). Recall that an irreducible Markov chain can transition from any one state to any other. So consider any two valid matchings of G, say $x = \{e_1, e_2, \ldots, e_k\}$ and $y = \{f_1, f_2, \ldots, f_\ell\}$, where $e_i, f_j \in E$ for all i, j. It is possible to transition from x to $\emptyset$ in k iterations of Algorithm 2, since with nonzero probability Steps 1–2 may successively choose and drop each edge e_i, for $1 \leq i \leq k$, and Step 4 may permit the transition in each iteration. By a similar argument, it is possible to transition from $\emptyset$ to y in ℓ iterations by successively choosing and adding each edge f_j, for $1 \leq j \leq \ell$. Thus, it is possible to transition from any matching x to any other y, and we conclude that the chain is irreducible.

Aperiodicity comes more easily: there is at least a 1/2 probability that the chain remains at its current state in every iteration (Step 4), so $\gcd\{t : P^t(x, x)\} = \gcd\{1, \ldots\} = 1$ for any $x \in \Omega$.

Since the chain is both irreducible and aperiodic, we conclude that it is ergodic. By Theorem 3.6, the chain must have a unique stationary distribution π. Inspection of the transition probabilities P shows that P is a symmetric matrix. Consider any two distinct matchings $x, y \in \Omega$ (note that if $x = y$, then we trivially have $P(x, y) = P(x, x) = P(y, x)$). If it is impossible to transition from x to y in one step, then $P(x, y) = 0 = P(y, x)$. Otherwise, x and y must differ by exactly one edge, say e^*. The probability of Step 1 choosing $e^* \in E$ is $1/|E|$, and the probability of committing to the transition in Step 4 is 1/2. So $P(x, y) = 1/(2|E|) = P(y, x)$. Therefore, P is symmetric and, by Proposition 3.7, $\pi(x) = 1/|\Omega|$, for all $x \in \Omega$.

Bringing it all together, we showed that the Markov chain (Ω, P) defined by Algorithm 2 is ergodic, implying that the stationary distribution $\pi(x) = 1/|\Omega|$ is unique. Thus, Algorithm 2 samples matchings of an input graph G uniformly at random, as desired.

3.1.3 The Metropolis Process

The Metropolis process (from the Metropolis-Hastings algorithm [7]) provides a powerful design framework for Markov chains. Given a state space Ω and a weight function $w : \Omega \to \mathbb{R}^+$, the goal is to sample states $x \in \Omega$ according to a distribution $\pi(x) = w(x)/Z$, where $Z = \sum_{y \in \Omega} w(y)$ is the partition function that makes π a valid probability distribution. The Metropolis process is a general template for assigning a Markov chain's transition probabilities so that its unique stationary distribution is π. This is an important tool because it allows us to choose a desired target distribution and construct a Markov chain that samples according to it.

The Metropolis process requires that the underlying Markov chain $\mathcal{M} = (\Omega, P)$ be both irreducible and *reversible* (not to be confused with *time-reversible*), meaning that for any pair of states $x, y \in \Omega$ such that $P(x, y) > 0$, we must also have $P(y, x) > 0$. Self-loops (transitions from a state to itself) are allowed, and may be desired to force $\mathcal{M}$ to be aperiodic.

For convenience, let $N(x) \subseteq \Omega$ be the set of neighboring states of $x \in \Omega$ (note that $N(x)$ does not include x itself). Every state $x \in \Omega$ has a distribution $\kappa(x, \cdot)$ that defines part of its transition probabilities. This distribution must satisfy two properties: (*i*) $\kappa(x, y) = \kappa(y, x) > 0$ if $y \in N(x)$, and (*ii*) $\sum_{y \in N(x)} \kappa(x, y) \leq 1$ and $\kappa(x, x) = 1 - \sum_{y \in N(x)} \kappa(x, y)$. With this distribution, we can define the Metropolis process (Algorithm 3).

Another name for the probabilities given in Step 2 is the *Metropolis filter.* The resulting Markov chain $\mathcal{M}$ is ergodic, since it was assumed to be irreducible and can be made aperiodic using self-loops. One can easily verify by detailed balance that π is the unique stationary distribution of $\mathcal{M}$,

Algorithm 3 The Metropolis Chain

```
Let $X_t \in \Omega$ denote the state at time $t$. Repeat:
Choose a neighboring state $X' \in N(X_t)$ with probability $\kappa(X_t, X')$.
With probability $\min\{1, \pi(X')/\pi(X_t)\} = \min\{1, w(X')/w(X_t)\}$, set $X_{t+1} \leftarrow X'$.
Otherwise, set $X_{t+1} \leftarrow X_t$.
```

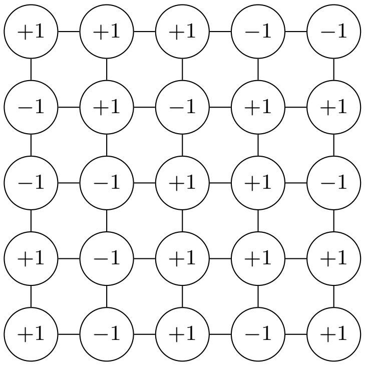


Figure 7: An Ising configuration on a 5 × 5 square grid.

as desired. W.l.o.g., assume that $w(x) \geq w(y)$. Then,

$$\pi(x)P(x,y) = \frac{w(x)}{Z} \cdot \kappa(x,y) \cdot \min\left\{1, \frac{w(y)}{w(x)}\right\} = \frac{w(x)}{Z} \cdot \kappa(x,y) \cdot \frac{w(y)}{w(x)} = \frac{w(y)}{Z} \cdot \kappa(y,x) = \pi(y)P(y,x).$$

The third equality follows from the requirement that $\kappa(x,y) = \kappa(y,x)$. This is true even if x and y aren't neighbors; in that case, $\kappa(x,y) = 0 = \kappa(y,x)$.

Therefore, by defining a weight function $w : \Omega \to \mathbb{R}^+$ and ensuring the underlying chain is ergodic and reversible, we can use the Metropolis process to ensure that it converges to a distribution that samples states proportional to their weight.

3.1.4 The Ising Model of Ferromagnetism

The Ising Model, originally proposed by Ernst Ising in 1925 [9],[5] is a model of ferromagnetism from statistical mechanics. Space is modeled as an $N \times N$ square grid $G = (V, E)$. The state space of this system is $\Omega = \{+1, -1\}^V$, where each configuration $\sigma \in \Omega$ assigns a *spin* $\sigma(v) \in \{+1, -1\}$ to each node $v \in V$ (see, e.g., Figure 7).

Each configuration $\sigma \in \Omega$ has *energy* defined by a *Hamiltonian*:

$$H(\sigma) = -\sum_{(u,v)\in E} \sigma(u)\sigma(v).$$

Each term $\sigma(u)\sigma(v)$ contributes $+1$ to the summation if $\sigma(u) = \sigma(v)$ (i.e., they have the same spin) and contributes -1 to the summation if $\sigma(u) \neq \sigma(v)$ (i.e., they have opposite spins). So another way of expressing a configuration's energy is:

$$H(\sigma) = -[(\#\text{same-spin edges in } \sigma) - (\#\text{opposite-spin edges in } \sigma)].$$

[5] This paper is in German, so not the most helpful reference for English-speaking students, I admit.

In an analogy to physical systems, let $T > 0$ denote *temperature* and $\beta = 1/T$ denote inverse temperature. The *weight* of a configuration $\sigma \in \Omega$ is defined as $w(\sigma) = e^{-\beta \cdot H(\sigma)}$. Using these weights, the probability that a configuration $\sigma \in \Omega$ appears should be:

$$\mu(\sigma) = w(\sigma)/Z = e^{-\beta \cdot H(\sigma)}/Z,$$

where $Z = \sum_{\tau \in \Omega} w(\tau) = \sum_{\tau \in \Omega} e^{-\beta \cdot H(\tau)}$ is the *normalizing constant* or *partition function* that makes μ a valid probability distribution. This distribution μ is known as the *Gibbs distribution.*

To develop our intuition, let's consider the extremes. If $\sigma \in \Omega$ has all same-spin edges, then: $w(\sigma) = e^{-\beta \cdot H(\sigma)} = e^{-\beta(-(|E|-0))} = e^{\beta|E|}$. On the other hand, if σ has all opposite-spin edges: $w(\sigma) = e^{-\beta \cdot H(\sigma)} = e^{-\beta(-(0-|E|))} = e^{-\beta|E|}$. The ratio between these weights is:

$$\frac{w(\text{all same-spin})}{w(\text{all opposite-spin})} = \frac{e^{\beta|E|}}{e^{-\beta|E|}} = e^{2\beta|E|} > 1.$$

Thus, the weight of configurations with mostly same-spin edges will always be higher than those that don't. However, we can still consider the effect of inverse temperature β. As β becomes larger (i.e., temperature T gets "colder"), the weight of configurations with a majority of same-spin edges becomes very large compared to those with a majority of opposite-spin edges. For example, if $\beta = 100$ (i.e., $T = 1/100$), then the all same-spin edges configuration has $e^{200|E|}$ times the weight of the all opposite-spin edges configuration. On the other hand, as β becomes smaller (i.e., temperature T gets "hotter"), the gap between the weights of majority same-spin edge and majority opposite-spin edge configurations also becomes smaller. For example, if $\beta = 1/(2|E|)$ (i.e., $T = 2|E|$), then the weight ratio between the all same-spin edges configuration and the all opposite-spin edges configuration is just e.

We would like to sample configurations from Ω according to the Gibbs distribution. To achieve this, consider the Markov chain defined in Algorithm 4. This Markov chain is designed using the Metropolis process (Section 3.1.3); note that Step 3 uses a Metropolis filter.

Algorithm 4 Metropolis Chain for the Ising Model on Grid $G = (V, E)$

```
Let $X_t \in \Omega$ denote the configuration at time $t$. Repeat:
1: Choose a node $u \in V$ and a spin $s \in \{+1, -1\}$ both uniformly at random.
2: Set $X'(u) = s$ and set $X'(v) = X_t(v)$ for all $v \neq u$.
3: With probability $\min\{1, w(X')/w(X_t)\}$, set $X_{t+1} \leftarrow X'$. Otherwise, set $X_{t+1} \leftarrow X_t$.
```

In Step 1, a node u on the grid is chosen uniformly at random, as is a spin s. Note that there is a 1/2 probability that the node u already has spin s, in which case nothing will change in this iteration. In Step 2, an intermediate configuration X' is constructed that updates the spin of u to match s and keeps all other nodes' spins the same. Step 3 uses a Metropolis filter to define the probability of making the transition to X'. Because X_t and X' only differ by a local change (the spin of node u), many of the terms cancel out when calculating $w(X')/w(X_t)$:

$$\begin{aligned}\frac{w(X')}{w(X_t)} = \frac{e^{-\beta\cdot H(X')}}{e^{-\beta\cdot H(X_t)}} &= \exp\left\{\beta \sum_{(i,j)\in E} X'(i)X'(j) - \beta \sum_{(i,j)\in E} X_t(i)X_t(j)\right\} \\ &= \exp\left\{\beta\left(\sum_{(i,j)\in E} X'(i)X'(j) - X_t(i)X_t(j)\right)\right\} \\ &= \exp\left\{\beta\left(\sum_{(u,j)\in E} X'(u)X'(j) - X_t(u)X_t(j)\right)\right\} \\ &= \exp\left\{\beta\left(\sum_{(u,j)\in E} X_t(j)\left(X'(u) - X_t(u)\right)\right)\right\}\end{aligned}$$

In more detail, the only change from X_t to X' is that $X'(u) \leftarrow s$, a possibly new spin. For all nodes $j \neq u$, $X_t(j) = X'(j)$. Thus, we obtain the third line from the second by observing that $X'(i)X'(j) = X_t(i)X_t(j)$ when $u \neq i \neq j$, and we obtain the fourth line from the third similarly. There are several cases to consider. First, if $X_t(u) = X'(u)$ (i.e., the spin s matches the spin that u already had), then nothing changes and the probability of transitioning to $X' = X_t$ is:

$$\min\left\{1, \frac{w(X')}{w(X_t)}\right\} = \min\left\{1, e^{\beta(\sum_{(u,j)\in E} X_t(j)(X'(u)-X_t(j)))}\right\} = \min\left\{1, e^{\beta(0)}\right\} = \min\{1,1\} = 1.$$

Otherwise, if $X_t(u) = +1$ and $X'(u) = -1$,

$$\frac{w(X')}{w(X_t)} = \exp\left\{-2\beta\left(\sum_{(u,j)\in E} X_t(j)\right)\right\} = \exp\left\{-2\beta((\#\text{ of }+1\text{ nbrs of }u) - (\#\text{ of }-1\text{ nbrs of }u))\right\}.$$

This probability is ≥ 1 when u has at least as many -1-spin neighbors and $+1$-spin neighbors, and is < 1 otherwise. An analogous but opposite conclusion can be drawn for when u changes from a -1 spin to a $+1$ spin. Thus, independent of inverse temperature β, the Markov chain always makes transitions that cause the spin of u to agree with at least as many neighbors as before. Transitions that cause the spin of u to disagree with more neighbors than before can still occur, but they become more unlikely as the number of disagreeing neighbors increases.

This probability calculation shows the mathematics behind the intuition that this Markov chain prefers configurations with more same-spin edges. This is further supported by the fact that, since this chain is based on the Metropolis process, it will converge to the desired Gibbs distribution at stationarity. We can verify this by detailed balance. Consider any two configurations $\sigma, \tau \in \Omega$ that differ by exactly one node, and suppose w.l.o.g. that $w(\sigma) \geq w(\tau)$. Then,

$$\begin{aligned}\pi(\sigma)P(\sigma,\tau) &= \frac{w(\sigma)}{Z}\cdot\frac{1}{|V|}\cdot\frac{1}{2}\cdot\min\left\{1, \frac{w(\tau)}{w(\sigma)}\right\} \\ &= \frac{w(\sigma)}{Z}\cdot\frac{1}{|V|}\cdot\frac{1}{2}\cdot\frac{w(\tau)}{w(\sigma)} \\ &= \frac{w(\tau)}{Z}\cdot\frac{1}{|V|}\cdot\frac{1}{2}\cdot 1 \\ &= \pi(\tau)P(\tau,\sigma)\end{aligned}$$

3.2 Mixing Time

We've seen several techniques for determining a Markov chain's stationary distribution, which characterizes a chain's long run behavior. However, we have not yet analyzed how long it would take a Markov chain to reach its stationary distribution. This running time is formalized by the *mixing time.* Recall that the total variation distance between two distributions μ and π is $d_{TV}(\mu, \pi) = (1/2)\sum_{x\in\Omega} |\mu(x) - \pi(x)|$.

Definition 3.10. *The time required for a Markov chain $\mathcal{M} = (\Omega, P)$ to "reach" stationarity from given an initial state $X_0 \in \Omega$, given an error tolerance $\varepsilon > 0$, is defined as:*

$$T_{mix}^{X_0}(\varepsilon) = \min\{t : d_{TV}(P^t(X_0, \cdot), \pi) \leq \varepsilon\}.$$

The *mixing time* *of $\mathcal{M}$ is the longest time to "reach" stationarity over all possible starting states:*

$$T_{mix}(\varepsilon) = \max_{X_0\in\Omega} T_{mix}^{X_0}(\varepsilon).$$

For our purposes, we can define $T_{\text{mix}} := T_{\text{mix}}(1/4)$. Using probability boosting (Algorithm 1), we can find the mixing time for an arbitrary $\varepsilon > 0$ by running our chain slightly longer: $T_{\text{mix}}(\varepsilon) = T_{\text{mix}}(1/4)\ln(1/\varepsilon)$. In the next section, we'll see our first general technique for bounding T_{mix}.

4 Coupling

Coupling is one technique for bounding the mixing time (Definition 3.10) of a Markov chain. Recall that when investigating a Markov chain's mixing time, we are interested in the amount of time it takes for the variation distance between a worst case arbitrary initial distribution and the chain's stationary distribution to come within some small error. As it turns out, we can use a coupling of these two distributions to exactly describe their variation distance. This, in turn, will help us bound the mixing time of the chain.

Suppose we have a finite state space Ω and two distributions μ, ν on Ω. Informally, a *coupling* ω on $\Omega \times \Omega$ is a distribution such that sampling $(\sigma, \tau) \sim \omega$ is like independently sampling $\sigma \sim \mu$ and $\tau \sim \nu$. In other words, each "row" of ω behaves like an element of μ and each "column" of ω behaves like an element of ν. More formally:

Definition 4.1. *Given a finite state space Ω and two distributions μ, ν on Ω, a distribution ω on $\Omega \times \Omega$ is a* *coupling* *if for all $\sigma \in \Omega$, $\sum_{\tau\in\Omega} w(\sigma, \tau) = \mu(\sigma)$ and for all $\tau \in \Omega$, $\sum_{\sigma\in\Omega} w(\sigma, \tau) = \nu(\tau)$.*

For example, let $\Omega = \{1, 2, 3, 4\}$, and suppose $\mu = (1/2, 1/4, 0, 1/4)$ and $\nu = (1/3, 1/3, 1/3, 0)$. Then both of the following distributions are example couplings of μ and ν:

$$\omega = \begin{bmatrix} 1/3 & 1/12 & 1/12 & 0 \\ 0 & 1/8 & 1/8 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/8 & 1/8 & 0 \end{bmatrix} \qquad \omega' = \begin{bmatrix} 1/3 & 1/12 & 1/12 & 0 \\ 0 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \end{bmatrix}$$

A sample from ω (or ω') is a pair $(\sigma, \tau) \in \Omega^2$. Observing only the choice of σ appears as if $\sigma \sim \mu$, and observing only the choice of τ appears as if $\tau \sim \nu$.

The following lemma is largely concerned with the probability that, given a sample (σ, τ) from a coupling, $\sigma \neq \tau$. When considering the matrix forms of a coupling ω (as in the examples above), this probability is the sum of all the off-diagonal elements:

$$\Pr[\sigma \neq \tau] = 1 - \Pr[\sigma = \tau] = 1 - \sum_{\eta \in \Omega} \omega(\eta, \eta).$$

Lemma 4.2 (Coupling Lemma [1])**.** *Let μ and ν be distributions on a finite state space Ω. Then:*

1. *For any coupling ω of μ and ν, a sample $(\sigma, \tau) \sim \omega$ satisfies $d_{TV}(\mu, \nu) \leq \Pr[\sigma \neq \tau]$.*
2. *There exists a coupling ω' of μ and ν such that for every sample $(\sigma, \tau) \sim \omega'$, $d_{TV}(\mu, \nu) = \Pr[\sigma \neq \tau]$.*

Proof. Observe that for any state $\eta \in \Omega$, we have $\omega(\eta, \eta) \leq \mu(\eta)$ and $\omega(\eta, \eta) \leq \nu(\eta)$. Otherwise, if say $\omega(\eta, \eta) > \mu(\eta)$, we would have:

$$\sum_{\eta' \in \Omega} \omega(\eta, \eta') = \omega(\eta, \eta) + \sum_{\eta' \neq \eta} \omega(\eta, \eta') > \mu(\eta),$$

violating the assumption that ω is a coupling. So we know that $\omega(\eta, \eta) \leq \min\{\mu(\eta), \nu(\eta)\}$.

To prove the first claim of the lemma, we can simply calculate the probability that $\sigma \neq \tau$ in a sample $(\sigma, \tau) \sim \omega$:

$$\begin{aligned}
\Pr[\sigma \neq \tau] &= 1 - \sum_{\eta \in \Omega} \omega(\eta, \eta) \\
&\geq 1 - \sum_{\eta \in \Omega} \min\{\mu(\eta), \nu(\eta)\} \\
&= \sum_{\eta \in \Omega} \mu(\eta) - \min\{\mu(\eta), \nu(\eta)\} \\
&= \sum_{\eta : \mu(\eta) < \nu(\eta)} (\mu(\eta) - \mu(\eta)) + \sum_{\eta : \mu(\eta) \geq \nu(\eta)} (\mu(\eta) - \nu(\eta)) \\
&= \max_{A \subseteq \Omega} \mu(A) - \nu(A) \\
&= d_{TV}(\mu, \nu)
\end{aligned}$$

The last equality follows from Definition 2.15. So we have $\Pr[\sigma \neq \tau] \geq d_{TV}(\mu, \nu)$, as desired.

To prove the second claim, we construct a new coupling ω' as follows. For every $\eta \in \Omega$, set $\omega'(\eta, \eta) = \min\{\mu(\eta), \nu(\eta)\}$. For distinct elements $\eta \neq \eta'$, set:

$$\omega'(\eta, \eta') = \frac{(\mu(\eta) - \omega(\eta, \eta))(\nu(\eta') - \omega(\eta', \eta'))}{1 - \sum_{\upsilon \in \Omega} \omega(\upsilon, \upsilon)}$$

Proving that this resulting distribution ω' is a valid coupling is a simple exercise in validating the sums given in Definition 4.1. Moreover, it is easy to see that when ω' is constructed this way, we have $\Pr[\sigma \neq \tau] = d_{TV}(\mu, \nu)$. The argument follows from the same equations as for the first claim above, except the inequality ($\geq$) in the second line is replaced by an equality, since all elements $\omega'(\eta, \eta) = \min\{\mu(\eta), \nu(\eta)\}$. □

We now give the general application of couplings to Markov chains. Suppose we have two copies of a Markov chain $\mathcal{M} = (\Omega, P)$ called (X_t) and (Y_t). We would like to define a joint evolution (X'_t, Y'_t) where the two chains can coordinate transitions, but both (X'_t) and (Y'_t) viewed in isolation behave according to P. Thus, for all $\sigma, \tau, \eta \in \Omega$ we need:

$$\Pr\left[X'_{t+1} = \eta | X'_t = \sigma, Y'_t = \tau\right] = P(\sigma, \eta), \text{ and}$$
$$\Pr\left[Y'_{t+1} = \eta | X'_t = \sigma, Y'_t = \tau\right] = P(\tau, \eta).$$

We will also specify that if $X'_t = Y'_t$, then $X'_{t+1} = Y'_{t+1}$; that is, once the two chains converge, they follow the same transitions going forward. These *Markovian couplings* are more restrictive than general couplings, since we want the joint evolution to also be a Markov chain (but on $\Omega \times \Omega$ instead of just Ω). For any starting state $(x, y) \in \Omega \times \Omega$ of the Markovian coupling (X'_t, Y'_t), let $T_c^{x,y}$ be the first time that the probability of the coupling continuing to diverge is small; i.e.:

$$T_c^{x,y} = \min\{t : \Pr\left[X'_t \neq Y'_t | X'_0 = x, Y'_0 = y\right] \leq 1/4\}.$$

By Lemma 4.2,

$$d_{TV}\left(P^{T_c^{x,y}}(x, \cdot), P^{T_c^{x,y}}(y, \cdot)\right) \leq \Pr\left[X'_{T_c^{x,y}} \neq Y'_{T_c^{x,y}} | X'_0 = x, Y'_0 = y\right] \leq 1/4.$$

If $y \in \Omega$ is chosen according to the stationary distribution π of $\mathcal{M}$, then for any $x \in \Omega$ we have:

$$d_{TV}\left(P^{T_c^{x,y}}(x, \cdot), \pi\right) \leq \Pr\left[X'_{T_c^{x,y}} \neq Y'_{T_c^{x,y}} | X'_0 = x, Y'_0 \sim \pi\right] \leq 1/4.$$

The *coupling time* of the Markovian coupling (X'_t, Y'_t) is defined as $T_c = \max_{x,y \in \Omega} T_c^{x,y}$. So, using Definition 3.10, we can conclude the following relationship between the coupling time of the Markovian coupling (X'_t, Y'_t) and the mixing time of $\mathcal{M}$:

$$T_{\text{mix}} = \max_{x \in \Omega} T^x_{\text{mix}}(1/4) = \max_{x \in \Omega} \left\{\min\{t : d_{TV}\left(P^t(x, \cdot), \pi\right) \leq 1/4\}\right\} \leq \max_{x \in \Omega} \left\{T_c^{x,y} : y \sim \pi\right\} \leq T_c.$$

We summarize these results in the following lemma.

Lemma 4.3. *The mixing time of a Markov chain $\mathcal{M} = (\Omega, P)$ is at most the coupling time of a Markovian coupling (X'_t, Y'_t) of two copies of $\mathcal{M}$, (X_t) and (Y_t).*

4.1 Example Applications of Couplings

We now turn to several examples of bounding a Markov chain's mixing time using the coupling technique.

4.1.1 Random Walks on the Hypercube

A *hypercube* in n dimensions is a graph $G = (V, E)$. The vertices of the hypercube are $V = \{0, 1\}^n$, all n-bit strings (so $|V| = 2^n$). Two vertices $u, v \in V$ have an edge $(u, v) \in E$ between them if their bit strings differ in exactly one position. For example, a 3-dimensional hypercube is depicted in Fig. 8. There is an edge between 000 and 001, for example, but not between 000 and 101.

Consider the Markov chain $\mathcal{M} = (V, P)$ defined in Algorithm 5. The state space of this chain is the vertex set of the hypercube. For each state X_t, we use $X_t(i)$ to refer to the i-th bit in the bit string of X_t.

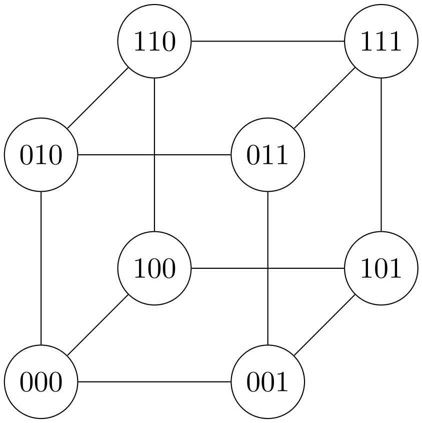


Figure 8: The hypercube in 3 dimensions.

Algorithm 5 Random Walk on the Hypercube

```
Let $X_t \in V$ denote the state at time $t$. Repeat:
Choose an index $i \in \{1, 2, \dots, n\}$ and a bit $b \in \{0, 1\}$ each uniformly at random.
Set $X_{t+1}(i) \leftarrow b$ and, for all $j \neq i$, set $X_{t+1}(j) \leftarrow X_t(j)$.
```

Essentially, each iteration of this chain assigns a random bit value b to a random index i in the current string. With probability $1/2$, the bit value matches the bit already in the string, so nothing changes. With the remaining probability, the bit string changes in one position, i.

Consider two copies of this Markov chain, (X_t) and (Y_t). We need to define a coupling $(X_t, Y_t) \to (X_{t+1}, Y_{t+1})$ so that when viewed independently, $X_t \to X_{t+1}$ and $Y_t \to Y_{t+1}$ look like the original transitions of Algorithm 5. The first and most obvious coupling is the following: if $X_t = Y_t$, then choose the same index i and bit b for both; otherwise, if $X_t \neq Y_t$, choose i_x, b_x for (X_t) and i_y, b_y for (Y_t) independently. Although this coupling definitely makes the individual chains behave as desired, the two copies don't necessarily converge quickly. For example, if we had $X_t = 0^n$ and $Y_t = 0^{n-1}1$, then the chain converges at time $t+1$ only if (i) each chain chooses the last index for both i_x and i_y (which occurs with probability $1/n^2$), and (ii) the bits are chosen such that $b_x = b_y$ (which occurs with probability $1/2$). Thus, even when X_t and Y_t are as close as possible, differing by only one bit, the probability they converge is very small: $1/(2n^2)$.

Instead, let's consider the following coupling: choose an index i and a bit b each uniformly at random, and use i, b for both (X_t) and (Y_t). To analyze this coupling, let $D_t = \{i : X_t(i) \neq Y_t(i)\}$ be the set of indices on which the chains disagree at time t, and similarly let $A_t = \{i : X_t(i) = Y_t(i)\}$ be the set of indices on which the chains agree at time t. In this coupling, there are two cases. If the chosen i is in D_t, then we have $X_t(i) \neq Y_t(i)$, but in the next step $X_{t+1}(i) = b = Y_{t+1}(i)$; thus, $|D_{t+1}| = |D_t| - 1$. Otherwise, if $i \in A_t$, then the chains already agreed at index i and both are assigned to b, so nothing changes (i.e., $|D_{t+1}| = |D_t|$).

We would like to show that the chains converge (i.e., $|D_t| = 0$) fairly quickly. Thus, consider the expectation of $d_t = |D_t|$:

$$\mathrm{E}\left[d_{t+1}\right] = (d_t - 1) \cdot \frac{d_t}{n} + d_t \cdot \frac{n - d_t}{n} = \frac{d_t^2}{n} - \frac{d_t}{n} + \frac{d_t n}{n} - \frac{d_t^2}{n} = d_t \left(1 - \frac{1}{n}\right).$$

By induction, we have that:

$$\mathrm{E}\left[d_t\right] \leq d_0 \left(1 - \frac{1}{n}\right)^t \leq ne^{-t/n}.$$

We also know that $\Pr\left[X_t \neq Y_t\right] \leq \mathrm{E}\left[d_t\right]$, since either the chains have converged and this probability — and therefore the number of disagreements — are both 0, or the chains have not yet converged and $\Pr\left[X_t \neq Y_t\right] = 1 \leq \mathrm{E}\left[d_t\right]$. Thus, setting $t = n\ln(4n)$, we have:

$$\Pr\left[X_t \neq Y_t\right] \leq \mathrm{E}\left[d_t\right] \leq ne^{-t/n} = 1/4.$$

Therefore, by Lemma 4.3, we conclude $T_{\text{mix}} \leq T_c = \mathcal{O}(n \log n)$.

4.1.2 Card Shuffling

Consider a deck of n distinct cards $\{c_1, c_2, \ldots, c_n\}$. The state space Ω in this situation is all the different ways the deck can be ordered, which is the set of permutations on $\{c_1, c_2, \ldots, c_n\}$. So $|\Omega| = n!$ (an example of a very large state space). Now consider the following "top-to-random" shuffling strategy (formalized in Algorithm 6): repeatedly take the card at the top of the deck and insert it at a position chosen uniformly at random from the n possible positions.

Algorithm 6 Top-to-Random Shuffling

```
Let $X_t \in \Omega$ be the order of the deck at time $t$. For $1 \leq i \leq n$, let $X_t(i)$ denote the card at the
$i$-th position. Repeat:
Choose a position $p \in \{1, \ldots, n\}$ uniformly at random, and set $X_{t+1}(p) \leftarrow X_t(1)$.
for $i \in \{1, \ldots, p-1\}$ do $X_{t+1}(i) \leftarrow X_t(i+1)$.
```

The Markov chain for top-to-random shuffling (Algorithm 6) is ergodic because it is both irreducible and aperiodic. To show irreducibility, consider any two states $\sigma, \tau \in \Omega$. To transition from σ to τ, start at σ and repeatedly put the top card at the bottom of the deck until the bottom card agrees with τ; then do the same for the next-lowest card, and so on until τ is reached. Aperiodicity follows more obviously: there is a $1/n$ probability of remaining in the same state at every iteration since the top card can be put back on top. However, this chain is not symmetric and its moves are not reversible, so we don't know the stationary distribution π of this chain immediately using, for example, Proposition 3.7.

The mixing time of top-to-random shuffling is difficult to analyze directly, but its inverse "random-to-top" shuffling isn't! Consider the Markov chain given in Algorithm 7.

Algorithm 7 Random-to-Top Shuffling

```
Let $X_t \in \Omega$ be the order of the deck at time $t$. For $1 \leq i \leq n$, let $X_t(i)$ denote the card at the
$i$-th position. Repeat:
Choose a card $c \in \{c_1, \ldots, c_n\}$ uniformly at random; let $p$ be the position of $c$ in $X_t$.
Set $X_{t+1}(1) \leftarrow X_t(p) = c$.
for $i \in \{1, \ldots, p-1\}$ do $X_{t+1}(i+1) \leftarrow X_t(i)$.
```

To bound the mixing time of random-to-top shuffling, we use the coupling technique. Consider two copies of the random-to-top Markov chain, say (X_t) and (Y_t). The coupling we use is to always

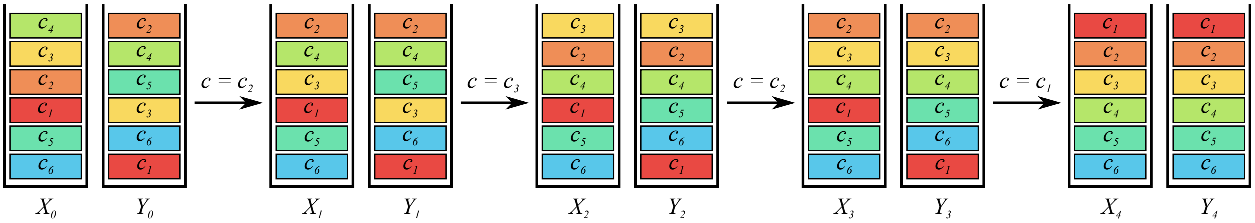

c_4 c_3 c_2 c_1 c_5 c_6
X_0
c_2 c_4 c_5 c_3 c_6 c_1
Y_0
$c = c_2$
c_2 c_4 c_3 c_1 c_5 c_6
X_1
c_2 c_4 c_5 c_3 c_6 c_1
Y_1
$c = c_3$
c_3 c_2 c_4 c_1 c_5 c_6
X_2
c_3 c_2 c_4 c_5 c_6 c_1
Y_2
$c = c_2$
c_2 c_3 c_4 c_1 c_5 c_6
X_3
c_2 c_3 c_4 c_5 c_6 c_1
Y_3
$c = c_1$
c_1 c_2 c_3 c_4 c_5 c_6
X_4
c_1 c_2 c_3 c_4 c_5 c_6
Y_4

Figure 9: Example of the identity coupling for Random-to-Top Shuffling (Algorithm 7). By moving the same card to move to the top in both chains, the chains will couple once each card is chosen at least once (or potentially before, as in this example).

choose the same random card c to move to the top for both chains, even though c may be in a different position for (X_t) and (Y_t). Note that this is a valid coupling since, from the individual perspective of either chain, c still functions as a card chosen uniformly at random. After the first transition $(X_0, Y_0) \to (X_1, Y_1)$, both chains agree on the top card, say c. After choosing another card $c' \neq c$ in a later step and putting it on top, both chains will agree on the positions of c and c'. After choosing yet another card $c'' \notin \{c, c'\}$ and putting it on top, the chains agree on the positions of c, c', and c''. This goes on until each card has been chosen at least once, at which point the chains have coupled (i.e., they agree on the positions of all n cards). An example is shown in Figure 9.

So it remains to analyze the expected time required to select all n cards. This is an instance of the classical *Coupon Collector* problem, in which there are n distinct "coupons" and in each step a coupon chosen uniformly at random is "collected". The goal is to understand how many steps are needed in expectation to collect all n coupons. It is well known that the number of steps required is $\mathcal{O}(n \log n)$ in expectation; for completeness, we give the argument here.

Lemma 4.4. *The expected number of steps to collect all n coupons in the Coupon Collector problem is $\mathcal{O}(n \log n)$.*

Proof. Let T be the time required to collect all n coupons, and let t_i denote the time to get the $(i+1)$-th unique coupon after collecting i unique coupons. It is easy to see that the probability of choosing a unique coupon after collecting i is:

$$p_i = 1 - \frac{i}{n} = \frac{n-i}{n}.$$

Thus, we have the following:

$$\mathrm{E}\left[T\right] = \sum_{i=0}^{n-1} \mathrm{E}\left[t_i\right] = \sum_{i=0}^{n-1} \frac{1}{p_i} = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=0}^{n-1} \frac{1}{n-i} = n \left(\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{2} + 1 \right)$$

But observing the following,

$$\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{2} = \sum_{i=2}^{n} \frac{1}{i} \leq \int_{x=1}^{n} \frac{1}{x} dx = \ln(n) - \ln(1) = \ln n,$$

we have $\mathrm{E}\left[T\right] \leq n(1 + \ln n) = \mathcal{O}(n \log n)$. □

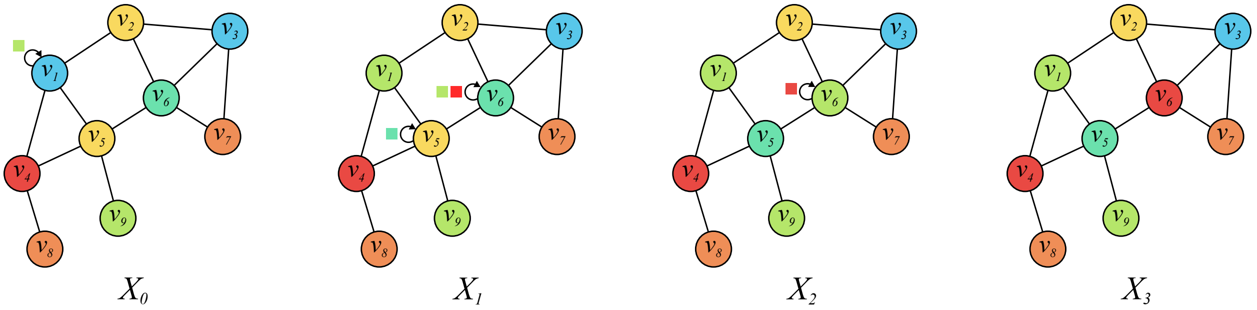


Figure 10: Illustration of the irreducibility argument for recoloring a graph according to Glauber dynamics (Algorithm 8). We wish to transform X_0 into X_3 using $\Delta + 2 = 6$ colors. Node $X_0(v_1)$ is blue, but $X_3(v_1)$ is green. Since v_1 has no green neighbors in X_0, it can simply recolor with green to obtain X_1. Nodes $v_2, \dots, v_4$ are all colored the same in X_1 and X_3, so we skip to v_5. Node v_5 needs to be teal as in X_3, but has a teal neighbor v_6 in X_1. So v_5 recolors with teal after v_6 recolors with a free color (one of green or red) to obtain X_2. Node v_6 can recolor to red as needed because it has no red neighbors, and all the remaining nodes agree with X_3.

Borrowing notation from the above proof, let T be the number of steps before the random-to-top shuffling coupling has coupled; i.e., they have chosen each of the n cards at least once. Then $\mathrm{E}[T] \leq n(1 + \ln n)$, by the above proof. By Markov's inequality, $\Pr[T > 4\mathrm{E}[T]] \leq 1/4$; thus, $\Pr\left[X_{4\mathrm{E}[T]} \neq Y_{4\mathrm{E}[T]}\right] \leq 1/4$. Therefore, by Lemma 4.3, we conclude $T_{\text{mix}} \leq T_c = \mathcal{O}(n \log n)$.

4.1.3 Graph Coloring

Consider an undirected graph $G = (V, E)$ on n vertices. The neighborhood of a vertex $u \in V$ is the set $N(u) = \{v \in V : (u, v) \in E\}$. The degree of a vertex u is the number of neighbors it has (i.e., $|N(u)|$), and the maximum degree of G is $\Delta = \max_{u \in V} |N(u)|$.

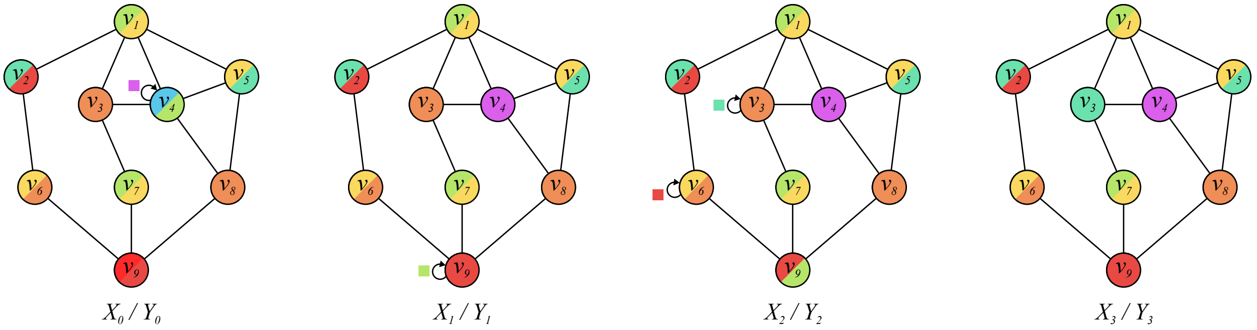
Definition 4.5. *Given $k \geq 2$ colors, a proper k-coloring of a graph $G = (V, E)$ is an assignment $c : V \to [k] = \{1, 2, \dots, k\}$ such that if $(u, v) \in E$, then $c(u) \neq c(v)$.*

When $k > \Delta$, there is always a way to properly k-color the graph; however, this is not always the case when $k \leq \Delta$. Let Ω be the set of all proper k-colorings of a graph G. Using Markov chain and Monte Carlo methods, we want to either estimate $|\Omega|$ or sample from Ω according to a target distribution. Consider the following Markov chain.

Algorithm 8 Graph Coloring (Glauber Dynamics)

```
    Let $X_t \in \Omega$ be the proper $k$-coloring of $G$ at time $t$. Repeat:
 1: Choose a vertex $u \in V$ and a color $i \in [k]$ each uniformly at random.
 2: for $v \neq u$ do $X_{t+1}(v) \leftarrow X_t(v)$.
 3: if $u$ has no neighbors with color $i$ then $X_{t+1}(u) \leftarrow i$.
 4: else $X_{t+1}(u) \leftarrow X_t(u)$.
```

In order to analyze this chain, we need to be sure it's ergodic (i.e., aperiodic and irreducible). The chain is clearly aperiodic, since for any vertex $u \in V$ chosen in Step 1 there is a $1/k$ probability of choosing the color it already has: $i = X_t(u)$. Moreover, we can show this chain is irreducible



X_0 / Y_0 X_1 / Y_1 X_2 / Y_2 X_3 / Y_3

Figure 11: Example of the identity coupling for Glauber dynamics (Lemma 4.6). Initially, the agreeing nodes are $A_0 = \{v_3, v_8, v_9\}$, and the rest compose the disagreeing nodes D_0. If a disagreeing node and a color not in its neighborhood is chosen (like v_4 and pink at $t = 0$), the node successfully recolors and the number of disagreeing nodes decreases by one. If an agreeing node and a color in one neighborhood but not the other is chosen (like v_9 and green at $t = 1$), the node recolors in one chain but not in the other, and the number of disagreeing nodes increase by one. All other recolorings (like the ones shown at $t = 2$) either succeed or fail in both chains and thus do not change the number of disagreements.

whenever there are at least $k \geq \Delta + 2$ colors. We need to show that we can transform any one proper k-coloring, say c_1, into any other proper k-coloring, say c_2, using only the vertex-by-vertex recoloring mechanism used by the chain (an example of this mechanism is shown in Figure 10). Fix some ordering of the vertices, say $V = \{v_1, v_2, \ldots, v_n\}$. Starting with $j = 1$, examine v_j. If $c_1(v_j) = c_2(v_j)$, nothing needs to be done since the colorings already agree at v_j. Otherwise, we need to recolor v_j with $c_2(v_j)$, but there are two cases. The easy case is when v_j has no neighbors that with color $c_2(v_j)$; in this case, we can simply recolor v_j with $c_2(v_j)$. However, if one or more of its neighbors does have color $c_2(v_j)$, then each such neighbor u must be recolored using a "free" color at u; i.e., one that does not conflict with any of the neighbors of u. ote that this "free" color must exist because $k \geq \Delta + 2$. Repeating this process iteratively (for $j = 2, \ldots, n$) will successfully transform coloring c_1 into c_2, so this chain is irreducible.

Now, since this chain is ergodic and its transition matrix is symmetric, we know that it converges to its unique stationary distribution, which is uniform (Proposition 3.7). So we can use the coupling technique to argue about its mixing time.

Lemma 4.6. *If $k > 3\Delta$, then $T_{mix} = \mathcal{O}(nk \log n)$, where $n = |V|$ and k is the number of colors.*

Proof. Consider two copies of the chain, (X_t) and (Y_t). Define an identity coupling (X_t, Y_t) that chooses the same vertex $u \in V$ and color $i \in [k]$ for both copies of the chain. Similar to the approach we took for random walks on the hypercube (Section 4.1.1), let $A_t = \{u \in V : X_t(u) = Y_t(u)\}$ be the set of vertices to which the colorings X_t and Y_t assign the same color, and let $D_t = \{u \in V : X_t(u) \neq Y_t(u)\}$ be the set of vertices to which the colorings X_t and Y_t assign different colors.

Given a vertex $u \in V$, we will be particularly interested in how the sets A_t and D_t intersect with its neighborhood; that is, we are interested in which neighbors of u are assigned the same colors by X_t and Y_t and which are not. For convenience, let $a_t(u) = |A_t \cap N(u)|$ and let $d_t(u) = |D_t \cap N(u)|$. Note that A_t and D_t form a partition of the vertex set V, and thus $A_t \cap N(u)$ and $D_t \cap N(u)$ form a partition of $N(u)$. Then we have the following:

$$\sum_{u \in A_t} d_t(u) = \sum_{u \in D_t} a_t(u).$$

We can establish this equality through a simple counting argument. Consider any vertex $v \in D_t$. The number of times v is counted on the left-hand side — in $\sum_{u \in A_t} d_t(u)$ — is exactly the number of times v is in the neighborhood of a vertex $u \in A_t$. Counted a different way, this is the number of neighbors of v that are in A_t, or $a_t(v) = |A_t \cap N(v)|$. An analogous argument shows that for any vertex $w \in A_t$, the number of times w is counted on the right-hand side is the number of neighbors of w that are in D_t, or $d_t(w) = |D_t \cap N(w)|$. So the equality holds.

The two copies of the chain have coupled once $|D_t| = 0$; i.e., the two colorings assign the same colors to all the vertices. As with the previous coupling arguments, we have that $\Pr[X_t \neq Y_t] \leq \mathrm{E}[|D_t|]$. So we need to analyze how $|D_{t+1}|$ may change with respect to $|D_t|$, hoping that it is likely to become smaller.

There are three cases (see Figure 11 for examples). The first is $|D_{t+1}| = |D_t| - 1$, which happens when a disagreeing vertex $u \in D_t$ and a color $i \in [k]$ are chosen such that no neighbor of u has color i in X_t or Y_t. This means that u can be recolored to $X_{t+1}(u) = Y_{t+1}(u) \leftarrow i$ in both chains, making u an agreeing vertex. To count the number of such colors, note that there can be at most $2|N(u)| - a_t(u)$ distinct colors in the neighborhood of u since u has $|N(u)|$ neighbors and $a_t(u)$ of them agree. Thus, there are at least $k - 2|N(u)| + a_t(u)$ colors that do not appear in the neighborhood of u in either chain.

The second case is $|D_{t+1}| = |D_t| + 1$, which happens when an agreeing vertex $u \in A_t$ and a color $i \in [k]$ are chosen such that u can be recolored to i in one chain but not the other (e.g., if a neighbor of u had color i in X_t but no neighbor of u had color i in Y_t). Colors that appear in the neighborhood of u in one chain but not the other cannot include any vertices in $A_t \cap N(u)$, since the chains agree on the colors of such vertices. However, such colors can be any of those used to color vertices in $D_t \cap N(u)$, of which there are at most $2d_t(u)$.

The last case is $|D_{t+1}| = |D_t|$, which occurs whenever the above two cases do not. Therefore,

$$\begin{aligned}
\mathrm{E}[|D_{t+1}|] &= (|D_t| - 1)\Pr[|D_{t+1}| = |D_t| - 1] + (|D_t| + 1)\Pr[|D_{t+1}| = |D_t| + 1] \\
&\quad + |D_t|(1 - \Pr[|D_{t+1}| = |D_t| - 1] - \Pr[|D_{t+1}| = |D_t| + 1]) \\
&= |D_t| - \Pr[|D_{t+1}| = |D_t| - 1] + \Pr[|D_{t+1}| = |D_t| + 1] \\
&\leq |D_t| - \sum_{u \in D_t} \left(\frac{k - 2|N(u)| + a_t(u)}{nk} \right) + \sum_{u \in A_t} \left(\frac{2d_t(u)}{nk} \right) \\
&= |D_t| - \frac{1}{nk} \left(\sum_{u \in D_t} (k - 2|N(u)| + a_t(u)) - \sum_{u \in D_t} 2a_t(u) \right) \\
&= |D_t| - \frac{1}{nk} \sum_{u \in D_t} (k - 2|N(u)| - a_t(u)) \\
&\leq |D_t| - \frac{|D_t|(k - 3\Delta)}{nk} \\
&= |D_t| \left(1 - \frac{k - 3\Delta}{nk} \right)
\end{aligned}$$

Therefore, given the starting states X_0 and Y_0 for the chains, we have:

$$\Pr\left[X_t \neq Y_t | X_0, Y_0\right] \leq \mathrm{E}\left[|D_t| \mid X_0, Y_0\right] \leq |D_0| \left(1 - \frac{k - 3\Delta}{nk}\right)^t \leq n \left(1 - \frac{1}{nk}\right)^t,$$

where the last inequality follows from our assumption that $k > 3\Delta$. But $n(1 - 1/nk)^t \leq ne^{-t/nk} \leq 1/4$ whenever $t \geq nk \ln(4n)$. So we conclude that $T_{\mathrm{mix}} = \mathcal{O}(nk \log n)$. □

It is worth noting that the mixing time should not, in general, grow with the number of colors as asserted by Lemma 4.6. More refined techniques, like the one outlined in the next section, will obtain a more reasonable mixing time of $\mathcal{O}(n \log n)$.

5 Path Coupling

Path coupling, like coupling, is a technique for bounding the mixing time of a Markov chain. However, when defining a path coupling between two copies of a chain (X_t) and (Y_t), one only needs to specify what the coupling does with pairs of adjacent states (X_t, Y_t). This is generally much easier than defining a coupling for all pairs of arbitrary states. For example, in the graph coloring application we saw in Section 4.1.3, we had to define a coupling that handled any pair of proper k-colorings (X_t, Y_t). In this section, we will define a path coupling that only has to handle pairs of colorings that disagree at a single vertex.

We first present the path coupling lemma as our main tool for these arguments. We will then re-prove Lemma 4.6 using path coupling to show that we can obtain the same bounds with simpler analysis. Finally, we will improve on the bounds of Lemma 4.6, showing that we can improve the mixing time to $\mathcal{O}(n \log n)$ with even less colors: $k > 2\Delta$.

5.1 The Path Coupling Lemma

The path coupling lemma was introduced by Bubley and Dyer [3]. Informally, this tool is used as follows. Given a state space Ω, we first need to decide which pairs $(X, Y) \in \Omega \times \Omega$ are considered "adjacent" in a way that these adjacencies connect Ω. Let $S \subseteq \Omega \times \Omega$ be the set of adjacencies. We then define a metric $d : \Omega \times \Omega \to \mathbb{R}_{\geq 1}$ as follows. Let $P = (X = Z_0, Z_1, \ldots, Z_\ell = Y)$ be a shortest path from X to Y such that $(Z_i, Z_{i+1}) \in S$ for each i, and set $d(X, Y) = \ell$. Thus, if $(X, Y) \in S$ — i.e., X and Y are adjacent — then $d(X, Y) = 1$.

Once we have a set S of pairs of adjacent states and the resulting metric d, we need to define a path coupling *only on the pairs of adjacent states in S*. Note that this is different (and simpler) than the usual coupling arguments we studied in Section 4, that required a coupling to be defined on all pairs of states in $\Omega \times \Omega$. The path coupling $(X_t, Y_t) \to (X_{t+1}, Y_{t+1})$ we define on pairs $(X_t, Y_t) \in S$ must have the property that, in expectation, the distance between X_{t+1} and Y_{t+1} is some fraction of the distance between X_t and Y_t, biasing the processes towards coupling.

If all of this is set up correctly, the path coupling lemma states that the path coupling we defined on pairs of adjacent states in S can be generalized to a coupling on all pairs of states in $\Omega \times \Omega$, yielding a bound on the mixing time. Formally, we have the following lemma.

Lemma 5.1 (Bubley-Dyer [3])**.** *Given a state space Ω, let $S \subseteq \Omega \times \Omega$ be a set of pairs of states such that (Ω, S) is connected. Let $d : \Omega \times \Omega \to \mathbb{R}_{\geq 1}$ be a distance metric where $d(X, Y)$ equals* *the length of the shortest path from X to Y in (Ω, S). If there exists $0 < \alpha < 1$ and a coupling $(X_t, Y_t) \to (X_{t+1}, Y_{t+1})$ such that for every $(X_t, Y_t) \in S$ we have*

$$\mathrm{E}\left[d(X_{t+1}, Y_{t+1}) | X_t, Y_t\right] \leq (1-\alpha) d(X_t, Y_t),$$

then:

$$T_{mix}(\varepsilon) \leq \frac{\ln(d_{max}/\varepsilon)}{\alpha},$$

where $d_{max} = \max_{X,Y \in \Omega \times \Omega} d(X,Y)$ is the diameter of (Ω, S).

Proof. It suffices to construct a coupling $(X_t, Y_t) \to (X_{t+1}, Y_{t+1})$ for all pairs of states $(X_t, Y_t) \in \Omega \times \Omega$ such that $\mathrm{E}\left[d(X_{t+1}, Y_{t+1}) | X_t, Y_t\right] \leq (1-\alpha) d(X_t, Y_t)$ for some $0 < \alpha < 1$. Consider any $(X_t, Y_t) \in \Omega \times \Omega$, and let $P = (X_t = Z_t^0, Z_t^1, \ldots, Z_t^{d(X_t, Y_t)} = Y_t)$ be a shortest path from X_t to Y_t in (Ω, S). Note that such a path must exist since we assumed S connects Ω. Since P is a shortest path in (Ω, S), each pair (Z_t^i, Z_t^{i+1}) must be in S for each $i \in \{0, 1, \ldots, d(X_t, Y_t) - 1\}$. By assumption, there exists $0 < \alpha < 1$ such that:

$$\mathrm{E}\left[d(Z_{t+1}^i, Z_{t+1}^{i+1}) | Z_t^i, Z_t^{i+1}\right] \leq (1-\alpha) d(Z_t^i, Z_t^{i+1}),$$

for each $i \in \{0, 1, \ldots, d(X_t, Y_t) - 1\}$. Thus,

$$\begin{aligned}
\mathrm{E}\left[d(X_{t+1}, Y_{t+1}) | X_t, Y_t\right] &\leq \sum_{i=0}^{d(X_t,Y_t)-1} \mathrm{E}\left[d(Z_{t+1}^i, Z_{t+1}^{i+1}) | Z_t^i, Z_t^{i+1}\right] \\
&\leq \sum_{i=0}^{d(X_t,Y_t)-1} (1-\alpha) d(Z_t^i, Z_t^{i+1}) \\
&= (1-\alpha) d(X_t, Y_t)
\end{aligned}$$

This establishes that our generalized coupling also satisfies the conditions of the lemma. Now,

$$\Pr\left[X_t \neq Y_t | X_0, Y_0\right] \leq \mathrm{E}\left[d(X_t, Y_t) | X_0, Y_0\right] \leq d(X_0, Y_0)(1-\alpha)^t \leq d_{max}(1-\alpha)^t.$$

Since $d_{max}(1-\alpha)^t \leq d_{max} e^{-t\alpha} \leq \varepsilon$ whenever $t \geq \ln(d_{max}/\varepsilon)/\alpha$, we conclude by Lemma 4.3 that $T_{\mathrm{mix}}(\varepsilon) \leq \ln(d_{max}/\varepsilon)/\alpha$. □

Note that while Lemma 5.1 is sufficient for our purposes, it is a truncated version of the full path coupling lemma in [3].

5.2 Applications to Graph Coloring

Recall from Section 4.1.3 that we defined a Markov chain that samples proper k-colorings of a graph $G = (V, E)$ uniformly (Algorithm 8). We would like to now analyze this chain using path coupling; that is, we would like to focus our attention on defining a coupling on pairs of k-colorings (states) that disagree on the color of exactly one vertex. More generally, let our metric be defined as follows: $d_t = d(X_t, Y_t) = |\{u \in V : X_t(u) \neq Y_t(u)\}|$.

Lemma 5.2. *If $k > 3\Delta$, then $T_{mix} = \mathcal{O}(nk \log n)$, where $n = |V|$ and k is the number of colors.*

Proof. Consider two copies of the chain defined in Algorithm 8, (X_t) and (Y_t). We will define a path coupling $(X_t, Y_t) \mapsto (X_{t+1}, Y_{t+1})$ for pairs of states (X_t, Y_t) that disagree at exactly one vertex, say $X_t(z) = c_x \neq c_y = Y_t(z)$. We can use the identity coupling: choose the same vertex $u \in V$ and color $i \in [k]$ for both copies of the chain.

Since X_t and Y_t agree on all vertices except z, the only updates that may succeed in one chain but fail in the other are when the vertex to update is $u \in N(z)$ and the color chosen is $i \in \{c_x, c_y\}$, which yields $d_{t+1} = d_t + 1 = 2$. The probability that this occurs is $(|N(z)|/n) \cdot (2/k)$.

All other updates will either fail in both chains (leaving the distance unchanged) or succeed in both chains. If the chosen vertex to update is z and the color $i \in [k]$ is not already coloring any of the neighbors of z in either chain, then z is successfully recolored in both chains. Since X_t and Y_t only disagree at z, they agree on the colors of all neighbors in $N(z)$. Thus, there are at least $k - |N(z)|$ colors that could successfully recolor z, and thus the probability that $d_{t+1} = d_t - 1 = 0$ is at least $(1/n) \cdot (k - |N(z)|)/k$. Combining these results, for any pairs of states disagreeing at exactly one vertex z, we have:

$$\begin{aligned}
\mathrm{E}\left[d_{t+1} | d_t = 1\right] &= 2 \cdot \Pr\left[d_{t+1} = 2\right] + 1 \cdot \Pr\left[d_{t+1} = 1\right] + 0 \cdot \Pr\left[d_{t+1} = 0\right] \\
&= 2 \cdot \Pr\left[d_{t+1} = 2\right] + \left(1 - \Pr\left[d_{t+1} = 2\right] - \Pr\left[d_{t+1} = 0\right]\right) \\
&= 1 + \Pr\left[d_{t+1} = 2\right] - \Pr\left[d_{t+1} = 0\right] \\
&\leq 1 + \frac{2|N(z)|}{nk} - \frac{k - |N(z)|}{nk} \\
&= 1 + \frac{3|N(z)| - k}{nk} \\
&\leq 1 + \frac{3\Delta - k}{nk} \\
&\leq 1 - 1/nk
\end{aligned}$$

whenever $k > 3\Delta$. Setting $\alpha = 1/nk$, we conclude $T_{\text{mix}}(1/4) \leq nk \ln(4 d_{max}) \leq nk \ln(4n) = \mathcal{O}(nk \log n)$ by Lemma 5.1. □

The following theorem uses a slightly different path coupling to establish fast mixing when the number of colors is $k > 2\Delta$. This was independently proven by Jerrum [10] and Salas and Sokal [20] using a direct coupling argument; our proof uses path coupling, which simplifies the analysis. Note that the coupling used in the proof below modifies the original Markov chain (Algorithm 8) by including all possible k-colorings of G in the state space Ω, whether they are proper or not.

Theorem 5.3. *If $k > 2\Delta$, then $T_{mix} = \mathcal{O}(nk \log n)$, where $n = |V|$ and k is the number of colors.*

Proof. Consider two copies of the chain defined in Algorithm 8, (X_t) and (Y_t). Once again, we will define a path coupling $(X_t, Y_t) \to (X_{t+1}, Y_{t+1})$ for pairs of states (X_t, Y_t) that disagree at exactly one vertex, say $X_t(z) = c_x \neq c_y = Y_t(z)$. We use a different coupling than that for Lemma 5.2, however. Choose the same vertex $u \in V$ for both chains. If $u \notin N(z)$ — that is, u is either z itself or not one of the neighbors of z — choose the same color $i \in [k]$ for both chains. Otherwise, if $u \in N(z)$, choose a color $i_x \in [k]$ uniformly at random for X_t and set i_y for Y_t as follows:

$$i_y = \begin{cases} c_x & \text{if } i_x = c_y; \\ c_y & \text{if } i_x = c_x; \\ i_x & \text{otherwise.} \end{cases}$$

For z to become an agreeing vertex, it needs to be chosen and recolored with a color not already in its neighborhood. The number of colors represented in $N(z)$ is at most $|N(z)|$, so the probability of successfully recoloring z is at least $(1/n)\cdot(k-|N(z)|)/k$. No other moves can increase the number of agreements, since z is the only disagreeing vertex.

For a new disagreement to occur, a vertex $u \in N(z)$ must be chosen and color i_x must be chosen to be c_y. In this case, $i_x = c_y$ and $i_y = c_x$ (by the coupling), so if both chains are successful in their recoloring, $X_{t+1}(u) = c_y$ and $Y_{t+1}(u) = c_x$, a new disagreement. It is possible, however, that one or both of these attempted recolorings fail (because u may have some other $i_{\{x,y\}}$-colored neighbor), so the probability that a new disagreement occurs is at most $(|N(z)|/n)\cdot(1/k)$. Any other recoloring will either succeed or fail in both chains, leaving the number of (dis)agreements unchanged. Therefore,

$$\begin{aligned}
\mathrm{E}\left[d_{t+1}|d_t = 1\right] &= 2\cdot \Pr\left[d_{t+1}=2\right] + 1\cdot \Pr\left[d_{t+1}=1\right] + 0\cdot \Pr\left[d_{t+1}=0\right] \\
&= 2\cdot \Pr\left[d_{t+1}=2\right] + (1 - \Pr\left[d_{t+1}=2\right] - \Pr\left[d_{t+1}=0\right]) \\
&= 1 + \Pr\left[d_{t+1}=2\right] - \Pr\left[d_{t+1}=0\right] \\
&\le 1 + \frac{|N(z)|}{nk} - \frac{k-|N(z)|}{nk} \\
&= 1 - \frac{k-2|N(z)|}{nk} \\
&\le 1 - \frac{k-2\Delta}{nk} \\
&\le 1 - 1/nk
\end{aligned}$$

whenever $k > 2\Delta$. Setting $\alpha = 1/nk$, we conclude $T_{\text{mix}}(1/4) \le nk\ln(4d_{max}) \le nk\ln(4n) = \mathcal{O}(nk\log n)$ by Lemma 5.1. □

6 Conductance

Let $\mathcal{M} = (\Omega, P)$ be a Markov chain with stationary distribution π. For any $x, y \in \Omega$, let $Q(x,y) = \pi(x)P(x,y)$. For subsets $A, B \subset \Omega$, let $Q(A,B) = \sum_{x\in A}\sum_{y\in B} Q(x,y)$ be the probability that, at stationarity, the next move goes from a state in A to a state in B. The *conductance* of a subset $S \subset \Omega$ is:

$$\Phi(S) = \frac{Q(S,\overline{S})}{\pi(S)},$$

where $\pi(S) = \sum_{x\in S}\pi(x)$. This captures the probability that, starting at some state of S at stationarity, the chain escapes S in the next step. Intuitively, if $\Phi(S)$ is large, then S is unlikely to "trap" the chain.

Definition 6.1. *The conductance of a Markov chain (Ω, P) with stationary distribution π is:*

$$\Phi_* = \min_{S\subset\Omega:\pi(S)\le 1/2} \Phi(S).$$

The conductance of a Markov chain measures its ability to escape "small" subsets. Large conductance means the chain is likely to travel among its state space quickly, while small conductance indicates that the chain is likely to be slow mixing. The following result formally relating the mixing time of a Markov chain to its conductance was proven independently by Sinclair and Jerrum [21] and Lawler and Sokal [15] in the late 1980s.

Lemma 6.2. *Given a Markov chain $\mathcal{M} = (\Omega, P)$ with stationary distribution π, we have:*

$$\frac{1}{4\Phi_*} \leq T_{mix}(1/4) \leq \frac{2}{\Phi_*^2} \log\left(\frac{1}{4\pi_{min}}\right),$$

where $\pi_{min} = \min_{x\in\Omega} \pi(x)$.

Proof. We only prove the lower bound, i.e., that $T_{mix}(1/4) \geq 1/(4\Phi_*)$. Given any $S \subset \Omega$, let:

$$\mu_S(x) = \begin{cases} \frac{\pi(x)}{\pi(S)} & \text{if } x \in S; \\ 0 & \text{if } x \in \overline{S}. \end{cases}$$

We first prove the claim that $\Phi(S) = d_{TV}(\mu_S P, \mu_S)$. By the definition of conductance (Definition 6.1), we have:

$$\begin{aligned}
\Phi(S) &= \sum_{x\in S}\sum_{y\notin S} \frac{\pi(x)}{\pi(S)} \cdot P(x,y) \\
&= \sum_{x\in S}\sum_{y\notin S} \mu_S(x)P(x,y) \\
&= \sum_{y\notin S}\sum_{x\in \Omega} \mu_S(x)P(x,y) \\
&= \sum_{y\notin S} [\mu_S P](y) \\
&= \sum_{y\notin S} [\mu_S P](y) - \mu_S(y)
\end{aligned}$$

The third inequality (where $x \in S$ is replaced with $x \in \Omega$) follows from the fact that $\mu_S(x) = 0$ whenever $x \notin S$, and thus the sum does not change. We get the final equality for the same reason: $\mu_S(y) = 0$ because $y \notin S$. Now, by the definition of total variation distance (Definition 2.15):

$$d_{TV}(\mu_S P, \mu_S) = \max_{A\subseteq\Omega} |[\mu_S P](A) - \mu_S(A)| = \sum_{y\in\Omega:[\mu_S P](y)\geq\mu_S(y)} [\mu_S P](y) - \mu_S(y)$$

Thus, if we show that summing over $\overline{S}$ and $A = \{y \in \Omega : [\mu_S P](y) \geq \mu_S(y)\}$ are equivalent, then $\Phi(S) = d_{TV}(\mu_S P, \mu_S)$ holds. If $y \notin S$, then $\mu_S(y) = 0 \leq [\mu_S P](y)$, and thus $y \in A$. If $y \in S$, then:

$$[\mu_S P](y) = \sum_{x\in\Omega} \mu_S(x)P(x,y) = \sum_{x\in S} \frac{\pi(x)}{\pi(S)} P(x,y) < \frac{1}{\pi(S)} \sum_{x\in\Omega} \pi(x)P(x,y) = \frac{\pi(y)}{\pi(S)} = \mu_S(y)$$

and thus $y \notin A$. So we have $\Phi(S) = d_{TV}(\mu_S P, \mu_S)$, as claimed. We use this in conjunction with the following fact (which we will not prove) to obtain the desired lower bound.

Observation 6.3. *Given a Markov chain $\mathcal{M} = (\Omega, P)$ and any two distributions μ and ν over Ω, we have:*

$$d_{TV}(\mu P, \nu P) \leq d_{TV}(\mu, \nu).$$

Combining this observation with the claim, we have, for all time $t \geq 0$:

$$d_{TV}(\mu_S P^{t+1}, \mu_S P^t) \leq d_{TV}(\mu_S P, \mu_S) = \Phi(S).$$

Thus,

$$d_{TV}(\mu_S P^T, \mu_S) \leq \sum_{t=0}^{T-1} d_{TV}(\mu_S P^{t+1}, \mu_S P^t) \leq \sum_{t=0}^{T-1} d_{TV}(\mu_S P, \mu_S) = T \cdot \Phi(S).$$

But,

$$\pi(\overline{S}) = |\mu_S(\overline{S}) - \pi(\overline{S})| \leq d_{TV}(\mu_S, \pi) \leq d_{TV}(\mu_S P^T, \mu_S) + d_{TV}(\mu_S P^T, \pi),$$

where the first equality comes from the fact that $\mu_S(\overline{S}) = 0$ by definition of μ_S, the first inequality comes from the definition of total variation distance with $A = \overline{S}$, and the final inequality is a triangle inequality for distributions. Combining the previous two results,

$$\pi(\overline{S}) \leq d_{TV}(\mu_S P^T, \mu_S) + d_{TV}(\mu_S P^T, \pi) \leq T \cdot \Phi(S) + d_{TV}(\mu_S P^T, \pi).$$

Recall that $\Phi_* = \min_{S:\pi(S)\leq 1/2} \Phi(S)$. So if $\pi(S) \leq 1/2$, then $\pi(\overline{S}) \geq 1/2$. Taking $T = T_{mix}(1/4)$,

$$\begin{aligned} \pi(\overline{S}) &\leq T \cdot \Phi_* + d_{TV}(\mu_S P^T, \pi) \\ \frac{1}{2} &\leq T_{mix}(1/4) \cdot \Phi_* + \frac{1}{4} \\ \frac{1}{4\Phi_*} &\leq T_{mix}(1/4) \end{aligned}$$

This proves the desired lower bound. □

6.1 Applications to Lazy Random Walks

Consider a random walk on a connected, undirected graph $G = (V, E)$ with state space $\Omega = V$ and transition matrix

$$P(u,v) = \begin{cases} 1/2 & \text{if } u = v; \\ 1/2d(u) & \text{if } (u,v) \in E; \\ 0 & \text{otherwise,} \end{cases}$$

where $d(u)$ is the degree of node u. Its stationary distribution is $\pi(u) = d(u)/2|E|$. Consider any subset of nodes $S \subset V$. Then,

$$\Phi(S) = \frac{Q(S,\overline{S})}{\pi(S)} = \frac{\sum_{u\in S}\sum_{v\notin S} \pi(u)P(u,v)}{\frac{1}{2|E|}\sum_{u\in S} d(u)} = \frac{\sum_{u\in S}\sum_{v\notin S:(u,v)\in E} \frac{d(u)}{2|E|} \cdot \frac{1}{2d(u)}}{\frac{1}{2|E|}\sum_{u\in S} d(u)} = \frac{|\partial S|}{2\sum_{u\in S} d(u)},$$

where $\partial S = \{(u,v) \in E : u \in S, v \notin S\}$.

6.1.1 The Complete Graph

Consider the complete graph $G = (V, E) = K_n$. Then the conductance of any subset $S \subset V$ is:

$$\Phi(S) = \frac{|\partial S|}{2\sum_{u\in S} d(u)} = \frac{|S|(n-|S|)}{2|S|(n-1)} = \frac{n-|S|}{2(n-1)}.$$

Thus, the conductance of the random walk on K_n is:

$$\Phi_* = \min_{S \subset V : \pi(S) \leq 1/2} \Phi(S) = \frac{n}{4(n-1)} \approx \frac{1}{4}.$$

By Lemma 6.2, we have that:

$$1 \leq T_{mix} \leq 32 \log(n/4) = \mathcal{O}(\log n).$$

In fact, by an easy coupling argument, we can show that $T_{mix} \leq 2$. Suppose two copies of the random walk (X_t) and (Y_t) start at nodes $x \neq y$, respectively. Use the identity coupling: choose the same vertex $w \in V$ uniformly at random and have both chains move to w with probability $1/2$. Since this is a complete graph, x and y are both neighbors of any such w, so the only way the walks do not couple in the next step is if laziness keeps one or both chains from moving to w. This occurs with probability at most $3/4$, so $\Pr[X_5 \neq Y_5] \leq (3/4)^5 < 1/4$ and thus $T_{mix} \leq T_c \leq 5$. So we see that Lemma 6.2 gives a fairly tight lower bound but a loose upper bound in the case of random walks on K_n.

6.1.2 The Barbell Graph

Consider the graph $G = (V, E)$ on n nodes composed of two components $A = K_{n/3}$ and $B = K_{2n/3}$ connected by a single edge. Then,

$$\pi(A) = \sum_{u \in A} \pi(u) = \sum_{u \in A} \frac{d(u)}{2|E|} \approx |A| \cdot \frac{n/3}{2|E|} = \frac{n^2}{18|E|}.$$

Now,

$$|E| = \binom{n/3}{2} + \binom{2n/3}{2} + 1 \approx \frac{n^2}{18} + \frac{4n^2}{18} = \frac{5n^2}{18}.$$

So $\pi(A) \approx 1/5 < 1/2$. Thus,

$$\Phi_* \leq \Phi(A) = \frac{|\partial A|}{2\sum_{u \in A} d(u)} \approx \frac{1}{2(n/3)^2} = \frac{9}{2n^2}.$$

So $1/\Phi_* \geq 2n^2/9$, and thus $T_{mix} \geq n^2/18$, by Lemma 6.2.

6.2 Coloring the Star Graph

Let $G = (V, E)$ be the star graph on n nodes; i.e., G is a graph with node $u \in V$ at its root and all $n-1$ other nodes have u as their only neighbor. We analyze the mixing time of Glauber dynamics for graph coloring (Algorithm 8) on this star graph. Consider the subset $S \subseteq \Omega$ of proper k-colorings that color the root node u with color 1, i.e., $S = \{X \in \Omega : X(u) = 1\}$. In order to analyze the conductance of S, we need to know what pairs of states $X \in S$ and $Y \notin S$ have $Q(X, Y) = \pi(X)P(X, Y) \neq 0$. But $P(X, Y) \neq 0$ if and only if:

1. $X(v) = Y(v)$ for all $v \neq u$, and
2. $X(v) \notin \{1, Y(u)\}$ for all $v \neq u$.

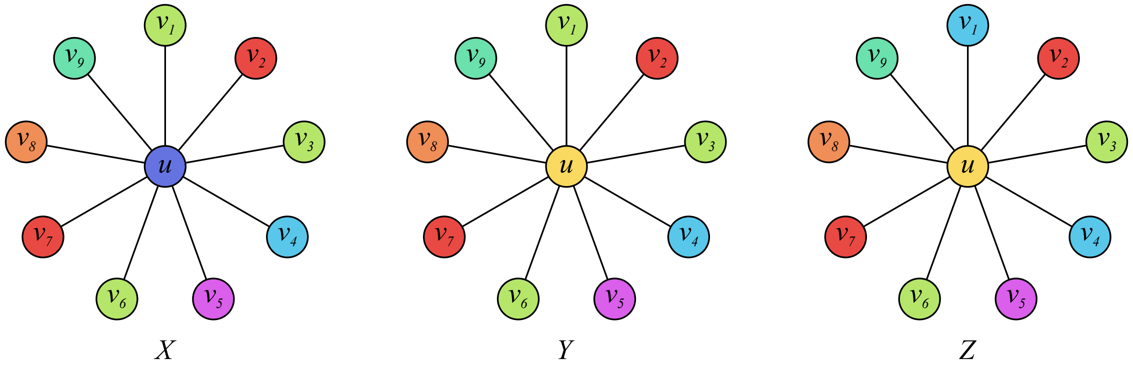


Figure 12: Example proper colorings of the star graph on 10 nodes. Under Glauber dynamics, it is possible to transition from X to Y in one step (by recoloring u with yellow), but is impossible to do so from X to Z because there is more than one disagreement (u and v_1).

See Figure 12 for an example. There are $(k-1)(k-2)^{n-1}$ such pairs, since there are $k-1$ possible colors for Y to color u with and $k-2$ colors for X and Y to color each $v \neq u$. Any such pair has $P(X,Y) \leq 1/n$ and — recalling from Section 4.1.3 — we have $\pi(X) = 1/|\Omega|$ for all $X \in \Omega$. So,

$$\Phi(S) = \frac{Q(S,\overline{S})}{\pi(S)} = \frac{1}{\pi(S)} \sum_{X \in S} \sum_{Y \notin S} Q(X,Y) \leq \frac{|\Omega|}{|S|} \sum_{X \in S} \sum_{Y \notin S} \frac{1}{|\Omega| n} = \frac{(k-1)(k-2)^{n-1}}{|S| n}.$$

But $X \in S$ if and only if $X(u) = 1$ and thus $X(v) \neq 1$ for all $v \neq u$. So $|S| = (k-1)^{n-1}$, and:

$$\Phi(S) \leq \frac{(k-1)(k-2)^{n-1}}{n(k-1)^{n-1}} = \frac{(k-1)^2}{n(k-2)} \left(1 - \frac{1}{k-1}\right)^n \leq \frac{(k-1)^2}{n(k-2)} \cdot e^{-n/(k-1)}.$$

Since $\Phi_* \leq \Phi(S)$, we have by Lemma 6.2 that:

$$T_{mix} \geq \frac{1}{4\Phi_*} \geq \frac{n(k-2)}{4(k-1)^2} \cdot e^{n/(k-1)}.$$

This result shows that if n is asymptotically larger than $\mathcal{O}(k \log k)$, then the mixing time is super-polynomial in n.

7 Spectral Gap

Like conductance (Section 6), the *spectral gap* of a Markov chain can be used to obtain both lower and upper bounds on the chain's mixing time. Consider a Markov chain $\mathcal{M} = (\Omega, P)$. Spectral gap methods use tools from linear algebra to analyze properties of the transition matrix P. We will primarily be using spectral gap methods as a tool to derive mixing time bounds; all relevant proofs for linear algebraic properties can be found in Chapter 12 of [16].

We first recall the relevant linear algebra. For convenience, let $n = |\Omega|$. An $n \times n$ matrix P has *eigenvector* $v = [v_1, v_2, \ldots, v_n]$ and *eigenvalue* λ if:

$$(P - \lambda I)v = 0 \quad \Rightarrow \quad Pv = \lambda v,$$

where I is the $n \times n$ identity matrix. This equation has a nonzero solution for v (and thus also for λ) if and only if $\det(P - \lambda I) = 0$. By the Fundamental Theorem of Algebra, this determinant can be factored as:

$$\det(P - \lambda I) = (\lambda_1 - \lambda)(\lambda_2 - \lambda) \cdots (\lambda_n - \lambda),$$

where $\{\lambda_1, \ldots, \lambda_n\}$ are the eigenvalues of P. Each λ_i may be real or complex, and may not be distinct. The corresponding $\{v_1, \ldots, v_n\}$ satisfying $Pv_i = \lambda_i v_i$ are the eigenvectors of P.

7.1 Linear Algebraic Properties of the Transition Matrix

The following properties hold in general.

Lemma 7.1. *For any finite Markov chain $\mathcal{M} = (\Omega, P)$,*

1. *If λ is an eigenvalue of P, then $|\lambda| \leq 1$. Moreover, the largest eigenvalue of P is $\lambda_1 = 1$.*
2. *If $\mathcal{M}$ is irreducible, the eigenvectors corresponding to the eigenvalue 1 are all generated by the $n \times 1$ vector $\vec{1}$.*
3. *If $\mathcal{M}$ is ergodic, then -1 is not an eigenvalue of P.*

Let $\langle v_1, v_2 \rangle = \sum_{x \in \Omega} v_1(x) v_2(x)$ be the usual inner product on $\mathbb{R}^{|\Omega|}$. Additionally define

$$\langle v_1, v_2 \rangle_\pi = \sum_{x \in \Omega} v_1(x) v_2(x) \pi(x)$$

as an inner product where the terms are "weighted" by their stationary probability $\pi(x)$. For ergodic Markov chains that are time-reversible with respect to a distribution π, we additionally have the following properties.

Lemma 7.2. *For any finite, ergodic Markov chain $\mathcal{M} = (\Omega, P)$ that is time-reversible with respect to distribution π,*

1. *The inner product space $(\mathbb{R}^n, \langle \cdot, \cdot \rangle_\pi)$ has an orthonormal basis of real-valued eigenvectors $\{v_1, \ldots, v_n\}$ corresponding to real eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$.*
2. *The transition matrix can be decomposed as:*

$$P^t(x, y) = \pi(y) \sum_{i=1}^{n} v_i(x) v_i(y) \lambda_i^t$$

Consider any function $f : \Omega \to \mathbb{R}$. Imagining f as a vector, it follows from Lemma 7.2 that:

$$P^t f = \sum_{i=1}^{n} \langle f, v_i \rangle_\pi v_i \lambda_i^t$$

7.2 Bounding the Mixing Time by the Relaxation Time

Suppose $\mathcal{M} = (\Omega, P)$ is a finite, ergodic Markov chain that is time-reversible with respect to distribution π. Let $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > -1$ be the eigenvalues of P, and define:

$$\lambda_* = \max\{|\lambda_i| : \lambda_i \neq 1\} = \max\{|\lambda_2|, |\lambda_n|\}.$$

The *spectral gap* of $\mathcal{M}$ is $1 - \lambda_2$, and the *absolute spectral gap* of $\mathcal{M}$ is $1 - \lambda_*$. By Lemma 7.1, we have that $0 < \lambda_* < 1$ and thus the absolute spectral gap is also strictly between 0 and 1.

Definition 7.3. *The relaxation time of a finite, ergodic, time-reversible Markov chain with absolute spectral gap $1 - \lambda_*$ is given by:*

$$T_r = \frac{1}{1 - \lambda_*}.$$

It is useful to understand the relationship between the relaxation time and the mixing time conceptually. Consider any function $f : \Omega \to \mathbb{R}$, and consider a state $x \in \Omega$ drawn according to the stationary distribution π. Then both $[P^t f](x)$ and $f(x)$ are random variables, and one can prove using the conclusion following Lemma 7.2 that:

$$\mathrm{Var}\left[[P^t f](x)\right] \leq \lambda_*^{2t} \mathrm{Var}\left[f(x)\right].$$

Noting that $0 \leq \lambda_* < 1$ and $\lambda_* = (1 - (1 - \lambda_*))$, we have that if $t \geq T_r$, then:

$$\lambda_*^{2t} \leq \lambda_*^{2/(1-\lambda_*)} = (1 - (1 - \lambda_*))^{2/(1-\lambda_*)} \leq e^{-2(1-\lambda_*)/(1-\lambda_*)} = 1/e^2.$$

This implies that:

$$\sqrt{\mathrm{Var}\left[[P^t f](x)\right]} \leq \sqrt{\lambda_*^{2t} \mathrm{Var}\left[f(x)\right]} \leq (1/e)\sqrt{\mathrm{Var}\left[f(x)\right]}$$

when $t \geq T_r$. Said in words, the standard deviation of $P^t f$ is at most $1/e$ times the standard deviation of f after the relaxation time. We can now state the formal relationship between a Markov chain's relaxation time and mixing time.

Theorem 7.4. *Let $\mathcal{M} = (\Omega, P)$ be a finite, ergodic Markov chain that is time-reversible with respect to distribution π. Then:*

$$(T_r - 1) \log\left(\frac{1}{2\varepsilon}\right) \leq T_{mix}(\varepsilon) \leq T_r \log\left(\frac{1}{\varepsilon \pi_{min}}\right),$$

where $\pi_{min} = \min_{x \in \Omega} \pi(x)$.

7.3 Applications to Lazy Random Walks

We begin with a useful result about the (absolute) spectral gap of lazy chains.

Proposition 7.5. *Consider a finite, ergodic Markov chain $\mathcal{M} = (\Omega, P)$. Let Q be a lazy version of P, given by:*

$$Q(x, y) = \begin{cases} 1/2 + P(x, y)/2 & \text{if } x = y; \\ P(x, y)/2 & \text{otherwise.} \end{cases}$$

Then the absolute spectral gap of Q is equal to its spectral gap; i.e., $\lambda_* = \lambda_2$.

Proof. Recall that we name the eigenvalues of Q according to their non-increasing order $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > -1$, and that $\lambda_* = \max\{|\lambda_2|, |\lambda_n|\}$. If all eigenvalues were non-negative, then it is easy to see that $\lambda_* = \lambda_2$.

So it suffices to show that the eigenvalues of Q are all non-negative. Observe that $Q = (I+P)/2$. Consider any eigenvalue γ_i of P, and let f_i be its corresponding eigenfunction. Then,

$$Qf_i = \frac{f_i + Pf_i}{2} = \frac{f_i + \gamma_i f_i}{2} = \left(\frac{1+\gamma_i}{2}\right) f_i.$$

By Lemma 7.1, $-1 \leq \gamma_i \leq 1$. Thus, letting $\lambda_i = (1+\gamma_i)/2$, we have that λ_i is an eigenvalue of Q and $0 \leq \lambda_i \leq 1$. Since this is true for all possible eigenvalues of Q, the lemma holds. □

7.3.1 The Hypercube

Consider the lazy version of the random walk on the hypercube (Algorithm 5). The states are $\Omega = \{0,1\}^n$, and the transition probabilities are given by:

$$P(x,y) = \begin{cases} 1/2 & \text{if } x = y; \\ 1/2n & \text{if } x \oplus y = 1; \\ 0 & \text{otherwise,} \end{cases}$$

where $x \oplus y$ is the number of bits x and y differ in.

It will be easier for our analysis to think of the states as $\Omega = \{-1,1\}^n$, but otherwise the chain remains unchanged. For a subset of indices $S \subset [n] = \{1, 2, \ldots, n\}$ and a state $x \in \Omega$, let $v_S(x) = \prod_{i \in S} x(i)$. We first show that $\{v_S\}_{S \subset [n]}$ are the eigenvectors of P with corresponding eigenvalues $\lambda_S = (n - |S|)/n$. We do so by verifying that $Pv_S = \lambda_S v_S$, for arbitrary $S \subset [n]$.

Consider any $S \subset [n]$ and any state $x \in \Omega$. Then,

$$[Pv_S](x) = \sum_{y \in \Omega} P(x,y) v_S(y) = \frac{1}{2} v_S(x) + \frac{1}{2n} \sum_{y \in \Omega : x \oplus y = 1} v_S(y),$$

by the definition of P. Now, the remaining y differ from x in exactly one bit. If that bit is in S, then $v_S(y) = -1 \cdot v_S(x)$; otherwise, $v_S(y) = v_S(x)$ as they are products over the same values. So,

$$\begin{aligned}
[Pv_S](x) &= \frac{1}{2} v_S(x) + \frac{1}{2n} \left(\sum_{i \in S} (-1) \cdot v_S(x) + \sum_{i \notin S} v_S(x) \right) \\
&= \frac{1}{2} v_S(x) + \frac{1}{2n} \left(-|S| \cdot v_S(x) + (n - |S|) v_S(x) \right) \\
&= \left(\frac{1}{2} - \frac{|S|}{2n} + \frac{n - |S|}{2n} \right) v_S(x) \\
&= \left(\frac{n - |S|}{n} \right) v_S(x)
\end{aligned}$$

Thus, letting $\lambda_S = (n - |S|)/n$, we have verified the claim. The (absolute) spectral gap is $1 - \lambda_* = 1 - (n-1)/n = 1/n$, and thus by Theorem 7.4,

$$\Omega(n) = (n-1)\log 2 \leq T_{mix}(1/4) \leq n \log(4/2^n) = n^2 \log(1/4) = \mathcal{O}(n^2).$$

As a comparison to conductance, consider the set $S = \{x : x(1) = 1\}$. Then,

$$\Phi(S) = \frac{|\partial S|}{2\sum_{x\in S} d(x)} = \frac{2^{n-1}}{2 \cdot 2^{n-1} \cdot n} = \frac{1}{2n}.$$

So $1/\Phi_* \geq 2n$, and thus $T_{mix} \geq 8n = \Omega(n)$, obtaining the same lower bound by Lemma 6.2.

7.3.2 The Cycle

First consider the (non-lazy) random walk on the cycle of n nodes. This can be thought of as a walk over the cyclic group $\Omega = \{0, 1, \ldots, n-1\}$, where:

$$P(x,y) = \begin{cases} 1/2 & \text{if } y \equiv (x \pm 1) \bmod n; \\ 0 & \text{otherwise.} \end{cases}$$

We claim that $v_k(x) = \cos(2\pi kx/n)$ are the eigenvectors of P, and $\lambda_k = \cos(2\pi k/n)$ are the corresponding eigenvalues. For convenience, let $\omega = e^{2\pi i/n}$, where $i = \sqrt{-1}$ is the imaginary unit. Then, by Euler's formula,

$$v_k(x) = \cos\left(\frac{2\pi kx}{n}\right) = \frac{\omega^{kx} + \omega^{-kx}}{2} \quad \text{and} \quad \lambda_k = \cos\left(\frac{2\pi k}{n}\right) = \frac{\omega^{k} + \omega^{-k}}{2}$$

We verify these eigenvectors and eigenvalues by showing $Pv_k = \lambda_k v_k$, for any $k \in \Omega$:

$$\begin{aligned}
[Pv_k](x) &= \sum_{y\in\Omega} P(x,y) v_k(y) \\
&= \frac{1}{2}(v_k(x-1) + v_k(x+1)) \\
&= \frac{1}{2}\left(\frac{\omega^{k(x-1)} + \omega^{-k(x-1)}}{2} + \frac{\omega^{k(x+1)} + \omega^{-k(x+1)}}{2}\right) \\
&= \frac{\omega^{kx}\cdot\omega^{-k} + \omega^{-kx}\cdot\omega^{k} + \omega^{kx}\cdot\omega^{k} + \omega^{-kx}\cdot\omega^{-k}}{4} \\
&= \frac{\omega^{kx}(\omega^{k} + \omega^{-k}) + \omega^{-kx}(\omega^{k} + \omega^{-k})}{4} \\
&= \frac{\omega^{k} + \omega^{-k}}{2} \cdot \frac{\omega^{kx} + \omega^{-kx}}{2} \\
&= \lambda_k v_k(x)
\end{aligned}$$

Now, as shown in Proposition 7.5, if we were to consider the lazy version of this random walk, then the eigenvalues would instead be:

$$\lambda'_k = \frac{1 + \lambda_k}{2} = \frac{1 + \cos(2\pi k/n)}{2}.$$

Thus, using the Taylor series for $\cos(x)$, the (absolute) spectral gap is:

$$1 - \lambda'_* = 1 - \frac{1 + \cos(2\pi/n)}{2} = \frac{1}{2}\left(1 - \left(1 - \frac{4\pi^2/n^2}{2!} + \frac{16\pi^4/n^4}{4!} - \cdots\right)\right) = \mathcal{O}(n^{-2})$$

By Theorem 7.4, we conclude:

$$\Omega(n^2) \leq T_{mix}(1/4) \leq \mathcal{O}(n^2)\log(4n) = \mathcal{O}(n^2 \log n).$$

8 Canonical Paths

The Max-Flow/Min-Cut Theorem states that the maximum amount of flow that can be sent over a network from a source node to a sink node is equal to the weight of the minimum cut in the network. Informally, it equates the maximum flow a network can handle to the throughput of its most restrictive bottleneck. This serves as a useful analogy for understanding canonical paths. In Section 6, we studied conductance as a way of finding bottlenecks in Markov chains. We showed that when there are cuts $S \subset \Omega$ in the state space with small conductance $\Phi(S)$, it was likely to become trapped in S, yielding slow mixing. In the analogy to Max-Flow/Min-Cut, analyzing conductance is like finding minimum cuts. Designing canonical paths, on the other hand, is like building maximum flows.

Consider a Markov chain $\mathcal{M} = (\Omega, P)$ with stationary distribution π. For every pair of states $x, y \in \Omega$, suppose we want to route $\pi(x)\pi(y)$ flow from x to y using the transitions of $\mathcal{M}$ as "pipes." The path $\gamma_{xy} = (x = z_0, z_1, \ldots, z_\ell = y)$ along which we send this flow from x to y is this pair's *canonical path.* Let $\Gamma = \{\gamma_{xy} : x, y \in \Omega\}$ be the set of all canonical paths. We want to construct Γ with small *congestion*, which we now define formally.

For a transition $t = (u, v)$ such that $P(u, v) > 0$, let $\Gamma_t = \{\delta_{xy} \in \Gamma : (u, v) \in \delta_{x,y}\}$ be the canonical paths that go through transition t. The congestion of Γ is:

$$\rho = \max_{t=(u,v)} \rho_t = \max_{t=(u,v)} \left\{ \frac{1}{\pi(u)P(u,v)} \cdot \sum_{\gamma_{xy} \in \Gamma_t} \pi(x)\pi(y)|\gamma_{xy}| \right\},$$

where the first term captures the "capacity" of transition $t = (u, v)$ and the summation captures the total flow going through t. We have the following upper bound on the mixing time in terms of congestion:

Theorem 8.1. *Given a Markov chain $\mathcal{M} = (\Omega, P)$ with stationary distribution π and a set of canonical paths Γ with congestion ρ, we have:*

$$T_{mix}(\varepsilon) \leq \rho \left(2 \ln \left(\frac{1}{\varepsilon} \right) + \ln \left(\frac{1}{\pi_{min}} \right) \right),$$

where $\pi_{min} = \min_{x \in \Omega} \pi(x)$. Asymptotically, $T_{mix}(1/4) = \mathcal{O}(\rho \log(1/\pi_{min}))$.

8.1 Lazy Random Walk on the Hypercube

We once again analyze the lazy version of the random walk on the hypercube (Algorithm 5). The states are $\Omega = \{0, 1\}^n$, and the transition probabilities are given by:

$$P(x, y) = \begin{cases} 1/2 & \text{if } x = y; \\ 1/2n & \text{if } x \oplus y = 1; \\ 0 & \text{otherwise,} \end{cases}$$

where $x \oplus y$ is the number of bits x and y differ in. Define the canonical path $\gamma_{xy} = (x = z_0, z_1, \ldots, z_\ell = y)$ as follows.

- Set $z_0 = x$.

- For $i \in \{1, 2, \ldots, n\}$, set $z_i(j) = z_{i-1}(j)$ for all $j \neq i$, and set $z_i(i) = y(i)$.
- Skip any steps for which $z_i = z_{i+1}$.

Thus, γ_{xy} is a valid x, y-path in the state space as each (z_i, z_{i+1}) differs by at most one bit, so $P(z_i, z_{i+1}) = 1/2n > 0$. Moreover, γ_{xy} has length at most n since there are n bits in each state.

We know that all non-self-loop transitions have probability $1/2n$, and thus the transition matrix is symmetric and the stationary distribution is uniform. So consider any non-self-loop transition $t = (u, v)$. Then,

$$\rho_t = \frac{1}{\pi(u)P(u,v)} \cdot \sum_{\gamma_{xy} \in \Gamma_t} \pi(x)\pi(y)|\gamma_{xy}| = \frac{(1/2^n)^2}{(1/2^n)(1/2n)} \cdot \sum_{\gamma_{xy} \in \Gamma_t} |\gamma_{xy}| \leq \frac{2n}{2^n} \cdot |\Gamma_t| \cdot n = \frac{2n^2}{2^n} \cdot |\Gamma_t|.$$

So it remains to analyze $|\Gamma_t|$, the number of canonical paths that use transition $t = (u, v)$. In general, this can be done by applying counting arguments that leverage problem-specific information about how the canonical paths were defined. Here, we use a more general method utilizing injective (one-to-one) mappings. If we can prove the existence of an injective mapping $\eta_t : \Gamma_t \to \Omega$, then we can conclude that $|\Gamma_t| \leq |\Omega|$ and thus:

$$\rho_t \leq \frac{2n^2}{2^n} \cdot |\Gamma_t| \leq \frac{2n^2}{2^n} \cdot 2^n = 2n^2.$$

Define such an injective mapping η_t as follows. Suppose transition $t = (u, v)$ flips bit i, and consider any $\gamma_{xy} \in \Gamma_t$. Then $\eta_t(\gamma_{xy}) = z$, where:

$$z(j) = \begin{cases} x(j) & \text{if } j \leq i; \\ y(j) & \text{otherwise.} \end{cases}$$

In words, z agrees with x on all bits up to and including i, and agrees with y on all bits after i. To show η_t in injective, we show that we can uniquely recover x and y using only z and $t = (u, v)$. We already know that x agrees with z on its first i bits. But x agrees with u on all bits after i because the canonical path flips bits one at a time, and transition t has just flipped bit i. Similarly, y agrees with v on its first i bits because those are the bits that have already been flipped along the canonical path, and we already know that y agrees with z on all bits after i. An example is shown below.

x	0	1	1	0	0	1
$\ldots$						
u	1	1	0	0	0	1
v	1	1	0	1	0	1
$\ldots$						
y	1	1	0	1	1	0
—	—	—	—	—	—	—
z	0	1	1	0	1	0

Thus, we conclude that η_t is injective, $|\Gamma_t| \leq |\Omega|$, and thus $\rho_t \leq 2n^2$. Because π is uniform and all non-self-loop transitions have the same probability, we have $\rho = \rho_t$. Therefore, by Theorem 8.1 and adding an extra factor of 2 to account for the lazy property of this chain, we conclude:

$$T_{mix}(\varepsilon) \leq 2\rho \left(2 \ln\left(\frac{1}{\varepsilon}\right) + \ln\left(\frac{1}{\pi_{min}}\right)\right) \leq 4n^2 \left(2 \ln\left(\frac{1}{\varepsilon}\right) + \ln(2^n)\right) = 4n^2(2\ln(1/\varepsilon) + n),$$

or, asymptotically, $T_{mix}(1/4) = \mathcal{O}(n^3)$. This is looser than the $\mathcal{O}(n^2)$ upper bound obtained using spectral gap methods (Section 7.3.1), but as we will see in the next section, canonical paths can be a useful tool where other methods can't obtain any nontrivial upper bounds at all.

8.2 Sampling Random Matchings

We gave a simple Markov chain for sampling matchings of a graph uniformly at random in Section 3.1.2. Here, we give a more general method for sampling random matchings of a graph $G = (V, E)$ according to a bias parameter $\lambda > 0$, according to distribution:

$$\pi(M) = \frac{\lambda^{|M|}}{Z},$$

where $Z = \sum_{M \in \Omega} \lambda^{|M|}$ is the *partition function* that normalizes the weights $\lambda^{|M|}$ as probabilities. Observe that if $\lambda = 1$, we seek to sample matchings uniformly at random; if $\lambda > 1$, we favor larger matchings in our sampling; and if $0 < \lambda < 1$, we favor smaller matchings in our sampling.

We use the Metropolis process (Section 3.1.3) to design a Markov chain for this problem. In particular, recall that the Metropolis probability for accepting a transition from a matching M to another matching M' is $\min\{1, \pi(M')/\pi(M)\}$. But,

$$\frac{\pi(M')}{\pi(M)} = \frac{\lambda^{|M'|}/Z}{\lambda^{|M|}/Z} = \lambda^{|M'|-|M|}.$$

So the Metropolis probabilities depend only on the number of edges gained or lost.

Algorithm 9 Sampling Random Matchings (Weighted)

```
   Let $X_t \in \Omega$ be the matching of $G = (V, E)$ at time $t$. Repeat:
1: Choose an edge $e = (u, v) \in E$ uniformly at random.
2: **if** $u$ and $v$ are both unmatched by $X_t$ **then** set $X' \leftarrow X_t \cup \{e\}$.
3: **else if** $e \in X_t$ **then** set $X' \leftarrow X_t \setminus \{e\}$.
4: **else if** $u$ is unmatched by $X_t$ but $v$ is matched by an edge $e' \in X_t$ (or vice versa) **then** set
   $X' \leftarrow X_t \cup \{e\} \setminus \{e'\}$.
5: **else** set $X' \leftarrow X_t$.
6: With probability $\min\{1, \lambda^{|X'|-|X_t|}\}$ set $X_{t+1} \leftarrow X'$; otherwise, set $X_{t+1} \leftarrow X_t$.
```

Algorithm 9 takes the standard form for a Metropolis chain: it constructs a proposed matching X' in Steps 1–5, and then transitions to X' with the Metropolis probability. For convenience, call the update in Step 2 an "add" move, the update in Step 3 a "remove" move, and the update in Step 4 a "shift" move. These moves can change the number of edges in the matching by at most one, so $\lambda^{|X'|-|X_t|} \in \{\lambda, 1, 1/\lambda\}$.

It is easy to see that this chain is irreducible: to transition from some matching M to another matching M', one could repeatedly remove all edges of M and then repeatedly add all edges of M'. Why do we have aperiodicity? One can also verify that π is the stationary distribution of this chain using detailed balance, though this is somewhat obvious because we proved that this holds for all Metropolis chains in Section 3.1.3. Consider a transition from M to M', and suppose w.l.o.g. that $\pi(M) \leq \pi(M')$.

$$\begin{aligned}
\pi(M)P(M,M') &= \pi(M)\cdot\frac{1}{|E|}\cdot\min\left\{1,\frac{\pi(M')}{\pi(M)}\right\}\\
&= \pi(M)\cdot\frac{1}{|E|}\cdot 1\\
&= \pi(M)\cdot\frac{1}{|E|}\cdot\frac{\pi(M')}{\pi(M')}\\
&= \pi(M')\cdot\frac{1}{|E|}\cdot\frac{\pi(M)}{\pi(M')}\\
&= \pi(M')P(M',M)
\end{aligned}$$

Thus, by the Fundamental Theorem of Markov Chains (Theorem 3.6), we know that this chain eventually converges to the distribution we want to sample from, as desired. However, when analyzing the mixing time of this chain, it was shown that any coupling argument will yield an exponential time upper bound [2]. So we use canonical paths for our analysis instead.

Consider any two matchings $x, y \in \Omega$ and note that their symmetric difference $x \oplus y$ (i.e., the edges strictly in x or in y, but not in both) are the edges we must remove from x and then add to the resulting intermediate graph to obtain y. In fact, we have that the graph $(V, x \oplus y)$ induced by the edges in the symmetric difference consist of vertex-disjoint paths and even-length cycles that alternate between edges of x and edges of y. We can use this observation to define canonical path γ_{xy}. Fix any ordering of the vertices, say $V = (v_1, v_2, \ldots, v_n)$. Now consider any component C of $(V, x \oplus y)$. Define the *start vertex* of C as the vertex of smallest label in C if C is a cycle, and as the endpoint of smaller label in C if C is a path. Order the components of $(V, x \oplus y)$ by increasing start vertex: $C_1, C_2, \ldots, C_k$, and let s_i be the start vertex of C_i. The canonical path γ_{xy} will process these components in this order, as follows:

- If C_i is a cycle (see Figure 13), remove the edge from x that is incident to s_i. Then, repeatedly shift edges of x to become edges of y. Finally, add the edge from y that is incident to s_i.

- If C_i is a path, there are two cases:

  - If s_i is incident to an edge of x (see Figure 14a), then remove this edge, repeatedly shift edges of x towards s_i to become edges of y, and finally add the last edge of y if the path has even length.
  - If s_i is incident to an edge of y (see Figure 14b), then repeatedly shift edges of x towards s_i and then add the last edge of y if the path has odd length.

Now that we've defined our canonical paths $\Gamma = \{\gamma_{xy} : x, y \in \Omega\}$, we need to bound congestion. Recall that the congestion of Γ is:

$$\rho = \max_{t=(M,M')}\left\{\frac{1}{\pi(M)P(M,M')}\cdot\sum_{\gamma_{xy}\in\Gamma_t}\pi(x)\pi(y)|\gamma_{xy}|\right\}$$

To bound ρ, we once again seek an injective mapping $\eta_t : \Gamma_t \to \Omega$. In particular, for a transition $t = (M, M')$, we want an injective mapping η_t that satisfies:

$$\frac{\pi(x)\pi(y)}{\pi(M)P(M,M')} \le m\overline{\lambda}^{2}\pi(\eta_t(\gamma_{xy})),$$

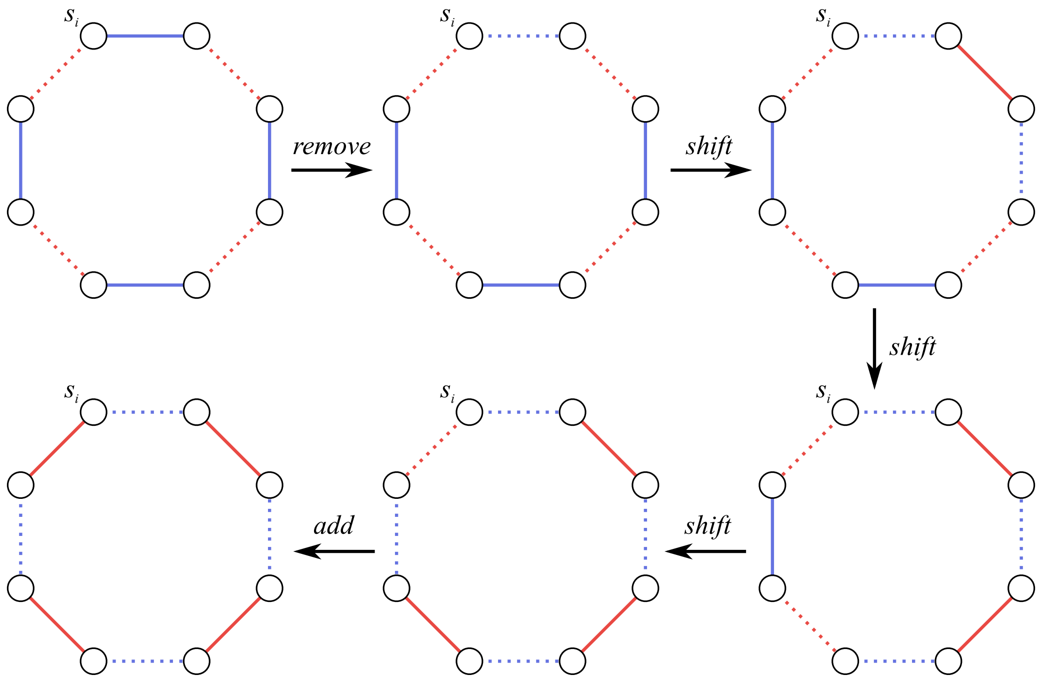


Figure 13: Processing a cycle in canonical path γ_{xy}. Edges of x are shown in blue and edges of y are shown in red. Solid edges are in the current matching, while dotted edges are not.

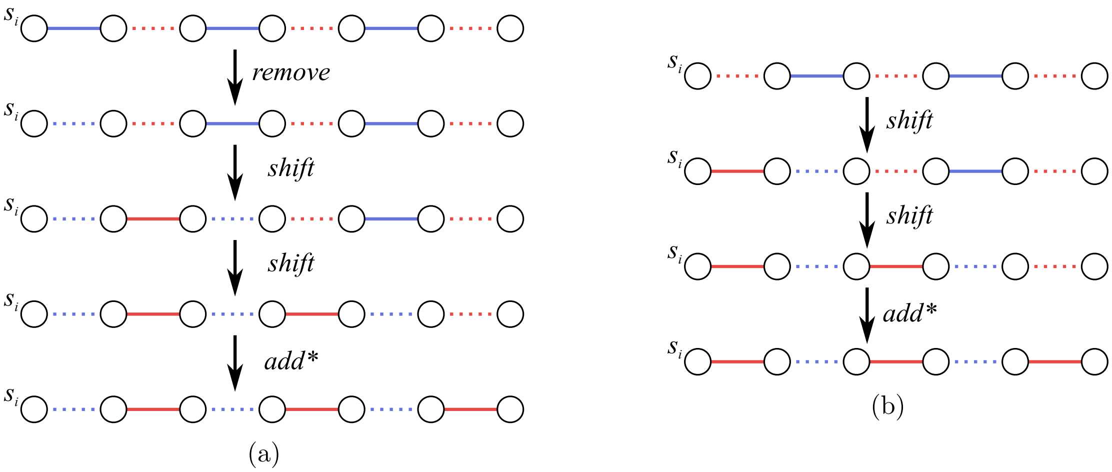


Figure 14: Processing a path in canonical path γ_{xy}, where start vertex s_i is incident to an edge of (a) x and (b) y. The final add moves in both cases are only needed if the path has (a) even or (b) odd length. Edges of x are shown in blue and edges of y are shown in red. Solid edges are in the current matching, while dotted edges are not.

where $m = |E|$ is the number of edges in the graph and $\overline{\lambda} = \max\{1, \lambda\}$. Assuming such a mapping exists, and observing that $|\gamma_{xy}| \leq n = |V|$, we have:

$$\begin{aligned}
\rho_t &= \frac{1}{\pi(M)P(M, M')} \cdot \sum_{\gamma_{xy} \in \Gamma_t} \pi(x)\pi(y)|\gamma_{xy}| \\
&\leq m\overline{\lambda}^2 \sum_{\gamma_{xy} \in \Gamma_t} \pi(\eta_t(\gamma_{xy}))|\gamma_{xy}| \\
&\leq nm\overline{\lambda}^2 \sum_{\gamma_{xy} \in \Gamma_t} \pi(\eta_t(\gamma_{xy})) \\
&\leq nm\overline{\lambda}^2.
\end{aligned}$$

The final inequality follows from the fact that $\sum_{\gamma_{xy} \in \Gamma_t} \pi(\eta_t(\gamma_{xy})) \leq 1$, which needs some explaining. Because η_t is injective, we know that each path $\gamma_{xy} \in \Gamma_t$ is mapped to a distinct state in Ω. Let $\Omega_t \subseteq \Omega$ be the range of η_t. Then,

$$\sum_{\gamma_{xy} \in \Gamma_t} \pi(\eta_t(\gamma_{xy})) = \sum_{z \in \Omega_t} \pi(z) = \pi(\Omega_t) \leq \pi(\Omega) = 1.$$

Since the $\rho_t \leq nm\overline{\lambda}^2$ bound holds for all transitions t, we have $\rho \leq nm\overline{\lambda}^2$. Thus, to obtain an upper bound on the mixing time using Theorem 8.1, it remains to bound π_{min}. The challenge here is in bounding the partition function Z, the denominator of $\pi(M)$. To calculate $Z = \sum_{M \in \Omega} \lambda^{|M|}$ exactly, one would need to enumerate all the matchings of G and their sizes, which is **#P**-hard (recall Section 2.2). Instead, we can obtain a crude bound as follows. The smallest matching is $M = \emptyset$, which has stationary weight $\pi(\emptyset) = \lambda^0/Z = 1/Z$. So,

$$\ln\left(\frac{1}{\pi_{min}}\right) \leq \ln(Z) = \ln\left(\sum_{M \in \Omega} \lambda^{|M|}\right) \leq \ln(|\Omega|\lambda^{n/2}) = \ln(|\Omega|) + (n/2)|\ln \lambda|.$$

As a crude upper bound on the number of matchings, we know that every matching is a subset of edges and there are 2^m distinct subsets of E (not all of which may be matchings), so $|\Omega| \leq 2^m$. Thus, $\ln(1/\pi_{min}) \leq m \ln 2 + (n/2)|\ln \lambda|$, so by Theorem 8.1,

$$T_{mix}(\varepsilon) \leq \rho\left(2\ln\left(\frac{1}{\varepsilon}\right) + \ln\left(\frac{1}{\pi_{min}}\right)\right) \leq nm\overline{\lambda}^2\left(2\ln\left(\frac{1}{\varepsilon}\right) + m\ln 2 + \frac{n|\ln \lambda|}{2}\right).$$

This concludes the main argument, but we need to show that the injective mapping η_t exists and has all the properties we needed. Consider any transition $t = (M, M')$, and call t a *cycle shift of* γ_{xy} if it is a shift move in a cycle C_i of $x \oplus y$. If t is a cycle shift of γ_{xy}, let e_t be the unique edge of x incident to the start vertex s_i of C_i. Define $\eta_t : \Gamma_t \to \Omega$ as follows:

$$\eta_t(\gamma_{xy}) = \begin{cases} (x \oplus y \oplus (M \cup M')) \setminus \{e_t\} & \text{if } t \text{ is a cycle shift of } \gamma_{xy}; \\ x \oplus y \oplus (M \cup M') & \text{otherwise.} \end{cases}$$

In words, $\eta_t(\gamma_{xy})$ agrees with x on the components that have already been processed, with y on the components that have yet to be processed, and with the edges common to both x and y. We prove three claims about η_t: that its range is Ω, that it is injective, and that it has the relationship needed to bound congestion ρ.

Claim 8.2. *For all transitions $t = (M, M')$ and canonical paths $\gamma_{xy} \in \Gamma_t$, $\eta_t(\gamma_{xy})$ is a matching.*

Proof. Consider edges $F = x \oplus y \oplus (M \cup M')$, and suppose there is a vertex $u \in V$ with degree two in the graph (V, F). So F contains two distinct edges (u, v_1) and (u, v_2), and since these are both incident to u one must belong to x and the other to y. Thus, both of these edges must be in $x \oplus y$ and — by the definition of F — neither can belong to $M \cup M'$. The only time a vertex has both its edges from $x \oplus y$ missing in two consecutive matchings along γ_{xy} is when the transition is a shift in cycle C_i and the vertex is s_i. But this is exactly how we defined a cycle shift of γ_{xy}, and in this case η_t drops the edge e_t in x that is incident to $u = s_i$. So in all cases, all vertices in graph $(V, \eta_t(\gamma_{xy}))$ have degree at most one, so $\eta_t(\gamma_{xy})$ is a matching. □

Claim 8.3. *For all transitions $t = (M, M')$, $\eta_t : \Gamma_t \to \Omega$ is injective.*

Proof. To show η_t is injective, we show how to uniquely recover matchings x and y using only $z = \eta_t(\gamma_{xy})$ and $t = (M, M')$. First, by the construction of η_t, we can recover $x \oplus y$ as:

$$x \oplus y = \begin{cases} (z \oplus (M \cup M')) \cup \{e_t\} & \text{if } t \text{ is a cycle shift of } \gamma_{xy}; \\ z \oplus (M \cup M') & \text{otherwise.} \end{cases}$$

Identifying e_t is a bit subtle. Edges $z \oplus (M \cup M')$ induce a set of paths and/or cycles, and we know which component C_i was being processed by transition t by inspecting M and M'. So, if C_i is a path, e_t is the unique edge that makes it a cycle. But how do we know whether or not to include e_t in $x \oplus y$; i.e., how do we know if t is a cycle shift of γ_{xy} if we don't know x and y? Because e_t is an edge of x, the ambiguous case is when C_i appears as an odd-length path that starts and ends with an edge of y. But these cases can be distinguished by the "direction" the shifts move in cycles versus paths of this kind (e.g., compare shifts in Figures 13 and 14b).

Given $x \oplus y$, we know the components $C_1, C_2, \ldots, C_k$ that are processed along the canonical path γ_{xy} and, given t, we know the component C_i that is currently being processed. So, as we observed earlier, we know that x agrees with z on components $C_1, \ldots, C_{i-1}$ while y agrees with z on components $C_{i+1}, \ldots, C_k$. For the component C_i being processed by t, x agrees with z on the already processed part and with M on the rest, while y agrees with z on the not yet processed part and with M' on the rest. This partitions $x \oplus y$ into the x and y portions; it remains to recover $x \cap y$ from z. This is easily done: $x \cap y = M \setminus (x \oplus y)$. So we conclude that η_t is injective. □

Claim 8.4. *For all transitions $t = (M, M')$ and canonical paths $\gamma_{xy} \in \Gamma_t$,*

$$\frac{\pi(x)\pi(y)}{\pi(M)P(M, M')} \le m\overline{\lambda}^2 \pi(\eta_t(\gamma_{xy})),$$

where $\overline{\lambda} = \max\{1, \lambda\}$.

Proof. Let $z = \eta_t(\gamma_{xy})$. Consider the quantities $\lambda^{|x|}\lambda^{|y|}$ and $\lambda^{|M \cup M'|}\lambda^{|z|}$, which are closely related to $\pi(x)\pi(y)$ and $\pi(M)P(M, M')\pi(z)$, respectively. We want to know what each edge $e \in E$ contributes to each of these quantities:

- If $e \notin x$ and $x \notin y$, then we also have $e \notin M \cup M'$ because all matchings on canonical path γ_{xy} are composed of edges from x and y. By the definition of η_t, we conclude $x \notin z$ as well. So e contributes $\lambda^0 = 1$ to both $\lambda^{|x|}\lambda^{|y|}$ and $\lambda^{|M \cup M'|}\lambda^{|z|}$.

- If $e \in x \oplus y$, then by the definition of η_t we know $e \in (M \cup M') \oplus z$ and the contribution to both $\lambda^{|x|}\lambda^{|y|}$ and $\lambda^{|M \cup M'|}\lambda^{|z|}$ is λ, with one exception. If t is a cycle shift of γ_{xy} and $e = e_t$, then $e \notin z$, so it contributes λ to $\lambda^{|x|}\lambda^{|y|}$ but only 1 to $\lambda^{|M \cup M'|}\lambda^{|z|}$.
- If $e \in x$ and $e \in y$, then we also have $e \in M \cup M'$ (again because all matchings along γ_{xy} are composed of edges from x and y). By the definition of η_t, we conclude $x \in z$ as well, and thus e contributes λ^2 to both $\lambda^{|x|}\lambda^{|y|}$ and $\lambda^{|M \cup M'|}\lambda^{|z|}$.

So we conclude that all edges except e_t contribute the same value to both quantities, and thus:

$$\lambda^{|x|}\lambda^{|y|} \leq \max\{1, \lambda\}\lambda^{|M \cup M'|}\lambda^{|z|} \leq \overline{\lambda}\lambda^{|M \cup M'|}\lambda^{|z|}.$$

Observe that $|M \cup M'|$ can have at most one more edge than $|M|$ or $|M'|$, so we have:

$$\begin{aligned} \lambda^{|x|}\lambda^{|y|} &\leq \overline{\lambda}\lambda^{|M \cup M'|}\lambda^{|z|} & \qquad \lambda^{|x|}\lambda^{|y|} &\leq \overline{\lambda}\lambda^{|M \cup M'|}\lambda^{|z|} \\ \lambda^{|x|}\lambda^{|y|} &\leq \overline{\lambda}^2\lambda^{|M|}\lambda^{|z|} & \qquad \lambda^{|x|}\lambda^{|y|} &\leq \overline{\lambda}^2\lambda^{|M'|}\lambda^{|z|} \\ \frac{\lambda^{|x|}\lambda^{|y|}}{Z^2} &\leq \frac{\overline{\lambda}^2\lambda^{|M|}\lambda^{|z|}}{Z^2} & \qquad \frac{\lambda^{|x|}\lambda^{|y|}}{Z^2} &\leq \frac{\overline{\lambda}^2\lambda^{|M'|}\lambda^{|z|}}{Z^2} \\ \pi(x)\pi(y) &\leq \overline{\lambda}^2\pi(M)\pi(z) & \qquad \pi(x)\pi(y) &\leq \overline{\lambda}^2\pi(M')\pi(z) \end{aligned}$$

So, we conclude that:

$$\begin{aligned} \pi(x)\pi(y) &\leq \overline{\lambda}^2 \min\{\pi(M), \pi(M')\}\pi(z) \\ &= \overline{\lambda}^2\pi(M) \cdot \frac{m}{m} \cdot \min\left\{1, \frac{\pi(M')}{\pi(M)}\right\}\pi(z) \\ &= m\overline{\lambda}^2\pi(M)P(M, M')\pi(z), \end{aligned}$$

and the claim holds. □

References

[1] D. J. Aldous. Random walks on finite groups and rapidly mixing Markov chains. *Séminaire de probabilités de Strasbourg*, 17:243–297, 1983.

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

[3] R. Bubley and M. Dyer. Path coupling: A technique for proving rapid mixing in Markov chains. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, SFCS '97, pages 223–231, 1997.

[4] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[5] L. R. Ford Jr. and D. Fulkerson. Maximum flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1st edition, January 1979.

[7] W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[8] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[9] E. Ising. Beitrag zur theorie des ferromagnetismus [contribution to the theory of ferromagnetism]. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[10] M. Jerrum. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Struct. Algorithms*, 7(2):157–165, 1995.

[11] M. Jerrum. *Counting, Sampling, and Integrating: Algorithms and Complexity.* Lectures in Mathematics. ETH Zürich. Birkhäuser, April 2003.

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

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

[14] P. W. Kasteleyn. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1st edition, 1967.

[15] G. F. Lawler and A. D. Sokal. Bounds on the L^2 spectrum for Markov chains and Markov processes: A generalization of Cheeger's inequality. *Transactions of the American Mathematical Society*, 309(2):557–580, 1988.

[16] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times.* American Mathematical Society, 2nd edition, October 2017. Available online at `http://pages.uoregon.edu/dlevin/MARKOV/markovmixing.pdf`.

[17] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1st edition, August 1995.

[18] N. Robertson, P. D. Seymour, and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Annals of Mathematics*, 150:929–975, 1999.

[19] R. W. Robinson and N. C. Wormald. Almost all regular graphs are Hamiltonian. *Random Structures & Algorithms*, 5(2):363–374, 1994.

[20] J. Salas and A. D. Sokal. Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem. *Journal of Statistical Physics*, 86(3):551–579, 1997.

[21] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, 1989.

[22] H. N. V. Temperley and M. E. Fisher. Dimer problem in statistical mechanics - an exact result. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, 6(68):1061–1063, 1961.

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