

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

2009

### Sampling edge covers

William August Rummler

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Rummler, William August, "Sampling edge covers" (2009). Thesis. Rochester Institute of Technology.  
Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

# Sampling edge covers

by

William August Rummler

May 26, 2009

A thesis submitted in partial fulfillment of  
the requirements for the degree of  
Master of Science  
(Computer Science)

Department of Computer Science  
B. Thomas Golisano College of Computing and Information Sciences  
Rochester Institute of Technology

---

Ivona Bezáková, Chair

---

Stanisław P. Radziszowski, Reader

---

Edith Hemaspaandra, Observer

*to Elizabeth*

## **Abstract**

The random generation or sampling of combinatorial objects has important applications to statistical analyses in various domains, including physics and evolutionary biology. Some objects whose sampling has been well-studied are contingency tables, matchings, and independent sets. Edge covers are related to matchings in graph theory; however, their association does not appear to be of the sort where results for sampling matchings imply results for sampling edge covers. Consequently, we investigate the sampling of edge covers and prove that efficiently sampling so-called  $(1,2)$ -edge covers is possible under reasonable restrictions.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Related work . . . . .	2
1.2 Preliminaries . . . . .	4
1.2.1 Edge covers . . . . .	4
1.2.2 Markov chains . . . . .	5
1.2.3 Markov chain Monte Carlo . . . . .	7
<b>2 Efficiently sampling (1,2)-edge covers</b>	<b>10</b>
2.1 Defining the canonical flow . . . . .	11
2.1.1 Processing the components of $\psi$ . . . . .	13
2.1.2 The flow, congestion, and connectedness of the state space . . . . .	14
2.2 Analyzing the canonical flow . . . . .	19
2.2.1 The encoding function . . . . .	20
2.2.2 Bounding the congestion and mixing time . . . . .	29
2.3 Graphs for which $\frac{ \mathcal{N} }{ \mathcal{P} } \in O(n^k)$ . . . . .	30
2.4 Initializing the Markov chain . . . . .	32
2.5 Discussion . . . . .	34

<b>A</b>	<b>Edge covers in context</b>	<b>35</b>
A.1	An application . . . . .	35
A.2	Related structures . . . . .	36
A.2.1	Matchings and edge covers . . . . .	36
A.2.2	Independent sets and vertex covers . . . . .	38
A.3	Related properties . . . . .	38
A.3.1	Gallai identities . . . . .	38
A.3.2	Relations via line graphs . . . . .	39
	<b>Bibliography</b>	<b>42</b>

# List of Figures

2.1	Markov chain over $(1, 2)$ -edge covers and near- $(1, 2)$ -edge covers . . . . .	11
2.2	Graph family for which $\frac{ \mathcal{N} }{ \mathcal{P} }$ is exponential . . . . .	12
2.3	Types of $(1, 2)$ -edge-coverage with respect to $(I, F) \in \Omega \times \mathcal{P}$ . . . . .	17
2.4	Possible signatures and coverages in $I \oplus F$ during processing . . . . .	23
2.5	Troublesome scenario for $z : \mathcal{N} \rightarrow \mathcal{P} \times V^8$ injection . . . . .	32

# Acknowledgments

This thesis work would have been trivially impossible without the inspiration and guidance of my advisor and committee chairperson, Ivona Bezáková. Her insight into the topic was essential, her ideas for attacking the problem were effective, and her patience with my absorption of the material never went unappreciated.

I also thank my reader Staszek Radziszowski and my observer Edith Hemaspaandra. Staszek's editing advice was invaluable in my attempts to write with suitable clarity, structure, and progression, while Edith's easy confidence, sharp wit, and critical eye have respectively encouraged, cheered, and educated me.

There are many others to thank for conceptual, technical, administrative, and moral support, including but not limited to the faculty and staff community of the RIT Department of Computer Science; the organizers, attendees, and presenters of the THEORY CANAL Rochester Theory Seminar; and the many researchers whose work provided the foundation and scaffolding for my own investigation.

To my siblings and closest friends, I offer thanks for encouraging me, listening to my rants, encouraging me some more, and then telling me to shut up and get on with it. In an undertaking of this magnitude or greater, it is impossible to underestimate the value of being emotionally grounded.

Finally, a gratitude that cannot be adequately expressed goes to my mother, father, and grandmother. They raised me. In these three words, I feel so much.



# Chapter 1

## Introduction

The random generation or sampling of combinatorial objects has important applications to statistical analyses in various domains, including physics and evolutionary biology. In addition, the sampling of a particular object can be used to count the number of those objects in the space from which one is sampling. Some objects whose sampling has been well-studied are contingency tables [2], matchings [11, 12, 14], and independent sets [6, 7, 18]. Edge covers are a type of graph-theoretic structure related to matchings.

An edge cover in a graph  $G = (V, E)$  is a subset  $W \subseteq E$  of edges such that each vertex  $v \in V$  is the endpoint of at least one edge in  $W$ . Edge covers and matchings are related in graph theory via the bijective equivalence of their analogous structures on vertices, respectively vertex covers and independent sets. They are also related by a certain property given by Tibor Gallai in 1959. This property, sometimes called the Gallai identity, simply states that the order  $n = |V|$  of a graph  $G = (V, E)$  is equal to the sum of two things: the cardinality of a maximum matching in  $G$ , and the cardinality of a minimum edge cover in  $G$ . After some preliminary work on this thesis (work which is presented in Appendix A), it appears that the associations between matchings and edge covers are not of the sort where results for sampling matchings imply results for sampling edge covers.

In this thesis work, we begin an investigation into the sampling of edge covers and prove

that efficiently sampling so-called  $(1, 2)$ -edge covers is possible under reasonable restrictions. Specifically, we show that it is possible to fully implement an efficient<sup>1</sup> Markov chain Monte Carlo sampler for  $(1, 2)$ -edge covers in graphs with minimum degree strictly greater than  $\frac{n-1}{3}$ . More generally, we show that this is also possible for any graph  $G$  where the ratio of  $|\mathcal{N}|$  (the number of so-called near- $(1, 2)$ -edge covers in  $G$ ) to  $|\mathcal{P}|$  (the number of  $(1, 2)$ -edge covers in  $G$ ) is upper-bounded by a polynomial in the size of  $G$ , as long as one can also efficiently generate a  $(1, 2)$ -edge cover in  $G$ .

Above we have spoken of sampling, but this thesis paper and the results of this thesis work are actually referring specifically to what is known as *approximately uniform sampling*. Uniform sampling is when objects are sampled in such a way so that every object that can possibly be generated has the same chance of being sampled. For example, if there are 100 objects in the space from which we are sampling, then uniform sampling implies that each object has a  $\frac{1}{100}$  chance of being generated by the sampler. Sampling is approximately uniform when each object’s chance of being sampled is “close” to the uniform value. This closeness is usually determined by the *sampling error*  $\epsilon$  given as a parameter to the sampler. Again usually, a lower sampling error will require the sampler to run for a longer amount of time.

Throughout this paper, all references to “sampling” should be interpreted as references to “approximately uniform sampling” unless otherwise stated.

## 1.1 Related work

Matchings, independent sets, and vertex covers are three types of structure loosely related to edge covers in graph theoretic terminology.<sup>2</sup> Independent sets and vertex covers are essentially the same structure due to their exact duality via edge-complementation of the graph in which they are defined. Matchings, however, are not dual to edge covers in general,

---

<sup>1</sup>“Efficient” is used in the common sense of computer science; that is, an algorithm is efficient if its running time is upper-bounded by a polynomial in the size of its input.

<sup>2</sup>Their definitions, properties, and some structural relationships are discussed extensively in Appendix A.

nor do they appear to be directly related, although there is a proven relationship between the cardinalities of maximum matchings and minimum edge covers.<sup>3</sup>

Thus there are these three similar structures, but only independent sets and matchings have a history of wide application in areas outside of graph theory. There is prior work on sampling independent sets [6, 7, 18], sampling matchings [14], and sampling perfect matchings [11, 12], but none on sampling edge covers so far as I have been able to determine through literature review.

The ability to efficiently sample independent sets varies over a range of small constant values for maximum degree  $\Delta$  in a graph. It is possible when  $\Delta = 4$ , unlikely to be possible when  $\Delta \geq 6$ , and strictly impossible when  $\Delta \geq 25$  unless  $RP = NP$  [6]. It is an open problem as to whether it is possible when  $\Delta = 5$ .

For all graphs, efficient sampling is possible for matchings and perfect matchings. The latter is particularly well-studied, since approximate counting reduces to approximately uniform sampling and since counting the number of perfect matchings in a graph is equivalent to computing the permanent of a zero-one matrix (one whose entries are each either zero or one), which is an important #P-complete problem in linear algebra.

There appears to be no relationship between edge covers and either matchings or independent sets that can be exploited to derive results on sampling edge covers from results on sampling matchings or sampling independent sets. Therefore, this thesis develops a direct result on sampling a certain kind of edge cover called a  $(1, 2)$ -edge cover.  $(1, 2)$ -edge covers are loosely related to perfect matchings<sup>4</sup> which, as mentioned above, are an important object in sampling research.

---

<sup>3</sup>As mentioned above, the Gallai identity states that the number of vertices in a graph is equal to the sum of the cardinalities of a maximum matching and a minimum edge cover in the graph.

<sup>4</sup>The  $(1, 2)$ -edge cover is almost as closely related to a perfect matching as any edge cover can be, since perfect matchings can also be called 1-edge covers.

## 1.2 Preliminaries

We lay the foundation for the development of our main result by going over the definition of an edge cover; by reviewing basic Markov chain notation, terms, and facts; and by discussing the Markov chain Monte Carlo sampling method and how our result enables its usage for sampling a special type of edge cover in graphs with certain properties.

Before continuing, you should be familiar with computer science theory, basic graph theory, and some probability theory.<sup>5</sup>

### 1.2.1 Edge covers

For completeness, here we summarize and re-state some basic terminology related to edge covers.

An *edge cover* in a finite simple graph  $G = (V, E)$  is a subset of  $E$  containing edges sufficient to *cover* (i.e., be incident on) all vertices in  $V$ . (Note that  $G$  has an edge cover if and only if  $G$  has no isolated vertices.) A *minimal edge cover* in  $G$  is an edge cover that is not a proper superset of any other edge cover. A *minimum edge cover* in  $G$  is an edge cover with cardinality less than or equal to the cardinality of any other edge cover. This minimum cardinality is called the *edge covering number* of  $G$ .

See Appendix A for more information related to edge covers, including presentation of an application of edge covers in weighted graphs and discussion of relationships between edge covers, matchings, independent sets, and vertex covers.

---

<sup>5</sup>I recommend the textbooks of Martin [16] and of Kleinberg and Tardos [15] as suitable introductions to computer science theory and to algorithm design, respectively. Chartrand and Zhang [3] provide an accessible introduction to graph theory, and the *Handbook of Graph Theory* [9] is a thorough reference work. Häggström ([10], Chapter 1) gives a concise summary of essential concepts from probability theory.

### 1.2.2 Markov chains

Please note that the following sequence of definitions and facts is a condensation of Chapters 2–6 in Häggström’s text [10] on finite Markov chains.<sup>6</sup> Good examples and thoughtful discussion of basic Markov chain theory are found therein.

#### Basics

Let  $\mathcal{X} = (X_t)_{t=0}^\infty$  be a discrete-time stochastic process with finite state space  $\Omega = \{\omega_i : 1 \leq i \leq k\}$  for some fixed  $k \geq 1$ . If the conditional distribution of  $X_{t+1}$  given  $(X_0, \dots, X_t)$  depends only on  $X_t$ , then  $\mathcal{X}$  is said to be *Markovian*. If, for fixed  $\omega \in \Omega$ , the conditional distribution of  $X_{t+1}$  given  $X_t = \omega$  is the same for all  $t$ , then  $\mathcal{X}$  is said to be *homogeneous*.

Assume  $\mathcal{X}$  is both Markovian and homogeneous.  $\mathcal{X}$  is now said to be a *Markov chain with transition matrix*  $P$ , where  $P$  is  $k \times k$  and  $P_{i,j} = \Pr(X_{t+1} = \omega_j \mid X_t = \omega_i)$  is the *transition probability* from.

The probability distribution of  $\mathcal{X}$  at time  $t$  is denoted as a row vector

$$\begin{aligned} \mu^{(t)} &= (\mu_i^{(t)})_{i=1}^k \\ &= (\Pr(X_t = \omega_i)_{i=1}^k). \end{aligned}$$

The *initial distribution* of  $\mathcal{X}$  is  $\mu^{(0)}$ . For any  $t \geq 0$ ,  $\mu^{(t)} = \mu^{(0)}P^t$ . For any  $s, t \geq 0$ ,  $\Pr(X_{s+t} = \omega_j \mid X_s = \omega_i) = (P^t)_{i,j}$ .

The transition matrix  $P$  may be indexed by any  $\omega_i, \omega_j \in \Omega$ , denoted  $P(\omega_i, \omega_j) = P_{i,j}$ . A distribution  $\mu$  on  $\Omega$  may also be indexed by any  $\omega_i \in \Omega$ , similarly denoted  $\mu^{(t)}(\omega_i) = \mu_i^{(t)}$ .

A useful way to examine  $\mathcal{X}$  is its *transition graph*, which is a directed graph  $\mathcal{K} = (\Omega, A)$  where  $A = \{(\omega, \omega') \in \Omega \times \Omega : P(\omega, \omega') > 0\}$ . Each directed edge corresponds to the transition from one state to another and is usually labeled with its transition probability.

---

<sup>6</sup>Some notation and vocabulary have been adjusted for consistency.

## Ergodicity

For two states  $\omega_i, \omega_j \in \Omega$ , if there exists  $t \geq 0$  such that  $\Pr(X_{s+t} = \omega_j \mid X_s = \omega_i) > 0$ , then  $\omega_i$  is said to *access*  $\omega_j$ . If  $\omega_i$  and  $\omega_j$  access each other, then they are said to *communicate*. If  $\omega_i$  and  $\omega_j$  communicate for all  $\omega_i, \omega_j \in \Omega$ , then  $\mathcal{X}$  is said to be *irreducible*; otherwise,  $\mathcal{X}$  is said to be *reducible*.

The *period* of a state  $\omega_i \in \Omega$  is defined as  $d(\omega_i) = \gcd\{t \geq 1 : (P^t)_{i,i} > 0\}$ . If  $d(\omega_i) = 1$ , then  $\omega_i$  is said to be *aperiodic*. If  $\omega$  is aperiodic for all  $\omega \in \Omega$ , then  $\mathcal{X}$  is said to be aperiodic; otherwise,  $\mathcal{X}$  is said to be *periodic*.

$\mathcal{X}$  is said to be *ergodic* if it is both irreducible and aperiodic.

## Stationarity

A row vector  $\pi = (\pi_i)_{i=1}^k$  is said to be a *stationary distribution* for  $\mathcal{X}$  if and only if the following two properties hold:

1.  $\pi$  is a probability distribution;  
i.e.,  $\pi_i \geq 0$  for all  $1 \leq i \leq k$ , and  $\sum_{i=1}^k \pi_i = 1$ .
2.  $\pi$  is stationary for the transition matrix  $P$ ;  
i.e.,  $\pi P = \pi$  (meaning  $\sum_{i=1}^k \pi_i P_{i,j} = \pi_j$  for all  $1 \leq j \leq k$ ).

For any Markov chain, there exists at least one stationary distribution; furthermore, there exists exactly one stationary distribution if the Markov chain is irreducible.

## Convergence

The *total variation distance* between two probability distributions  $\nu^{(1)}$  and  $\nu^{(2)}$  on  $\Omega$  is defined as

$$\begin{aligned} d_{\text{TV}}(\nu^{(1)}, \nu^{(2)}) &= \frac{1}{2} \sum_{i=1}^k |\nu_i^{(1)} - \nu_i^{(2)}| \\ &= \max_{A \subseteq \Omega} |\nu^{(1)}(A) - \nu^{(2)}(A)|. \end{aligned}$$

Let  $(\nu^{(t)})_{t=1}^{\infty}$  be an infinite collection of probability distributions on  $\Omega$ , and let  $\nu$  be a probability distribution on  $\Omega$ . If  $\lim_{t \rightarrow \infty} d_{\text{TV}}(\nu^{(t)}, \nu) = 0$ , then  $\nu^{(t)}$  is said to *converge to  $\nu$  in total variation as  $t \rightarrow \infty$* , denoted  $\nu^{(t)} \xrightarrow{\text{TV}} \nu$ .

**Fact 1.1** ([10], Theorem 5.2 — Markov chain convergence). *For arbitrary initial distribution  $\mu^{(0)}$ , subsequent distributions  $(\mu^{(t)})_{t=1}^{\infty}$ , and unique stationary distribution  $\pi$  of an ergodic Markov chain, it is known that  $\mu^{(t)} \xrightarrow{\text{TV}} \pi$ . The Markov chain is said to approach stationarity as  $t \rightarrow \infty$ .*

## Reversibility

A probability distribution  $\pi$  on  $\Omega$  is said to be *reversible* for  $\mathcal{X}$  if  $\pi_i P_{i,j} = \pi_j P_{j,i}$  for all  $1 \leq i, j \leq k$ .  $\mathcal{X}$  is said to be reversible if there exists at least one reversible distribution for  $\mathcal{X}$ .

**Fact 1.2** ([10], Theorem 6.1). *If  $\pi$  is a reversible distribution for  $\mathcal{X}$ , then  $\pi$  is also a stationary distribution for  $\mathcal{X}$ .*

### 1.2.3 Markov chain Monte Carlo

*Markov chain Monte Carlo* (MCMC) is a method of sampling from a state space  $\Omega$  according to an arbitrary distribution  $\pi$  by simulating an ergodic Markov chain  $\mathcal{X} = (X_t)_{t=0}^{\infty}$  on  $\Omega$ , with  $|\Omega| \times |\Omega|$  transition matrix  $P$ , where  $\pi$  is the unique stationary distribution of  $\mathcal{X}$ .

## Constructing the Markov chain

The first step in MCMC is to construct an ergodic  $\mathcal{X}$  having the desired distribution  $\pi$  as its unique stationary distribution. Most of the time, ergodicity is easily produced and shown.<sup>7</sup> After that, ensuring that  $\pi$  is the stationary distribution of  $\mathcal{X}$  is easy due to the *Metropolis-Hastings algorithm* ([8], §1.3.3), which is a general procedure that defines  $P(\omega, \omega')$  for all  $\omega, \omega' \in \Omega$  in such a way as to cause  $\pi$  to be the stationary distribution of  $\mathcal{X}$ .

When  $\pi$  is the *uniform distribution* on  $\Omega$  (i.e., when  $\pi(\omega) = \frac{1}{|\Omega|}$  for all  $\omega \in \Omega$ ), the Metropolis-Hastings algorithm simplifies nicely. Since  $\pi$  is uniform, demonstrating  $P(\omega, \omega') = P(\omega', \omega)$  for all  $\omega, \omega' \in \Omega$  suffices to show reversibility of  $\pi$ . Then Fact 1.2 and the ergodicity of  $\mathcal{X}$  imply that  $\pi$  is the unique stationary distribution of  $\mathcal{X}$ . In other words, if  $\mathcal{X}$  is ergodic and  $P$  is symmetric, then  $\mathcal{X}$  has *uniform stationarity*.

## Simulating the Markov chain

A simulation of  $\mathcal{X}$  begins in some definite state  $X_0 = \alpha \in \Omega$ , continues for some  $T$  time-steps, and ends in some definite state  $X_T = \omega \in \Omega$ . For a simulation to proceed from any step  $X_t$  to the next step  $X_{t+1}$ , the transition matrix  $P$  must be known in one of two ways: explicitly, by computing and storing each entry of  $P$  beforehand; or implicitly, by defining an algorithm (typically called a *trial*) that transforms the structure of any state to that of another state.<sup>8</sup>

If  $P$  is explicitly known, then  $X_{t+1}$  can be computed from  $X_t$  by an easily derived function  $\phi : \Omega \times [0, 1] \rightarrow \Omega$  given by Häggström ([10], (21)). However, in a typical MCMC application, the direct evaluation, storage, and manipulation of  $P$  is rendered impractical by the astronomical (or even infinite) magnitude of  $|\Omega|$ . When this happens,  $P(\omega, \omega')$  is implicitly defined for all  $\omega, \omega' \in \Omega$  by a trial that takes  $X_t$  as input and gives  $X_{t+1}$  as output.

---

<sup>7</sup>That  $\mathcal{X}$  is aperiodic can be ensured by incorporating positive-probability loops  $P(\omega, \omega) > 0$  for all  $\omega \in \Omega$ . The difficulty of producing and proving irreducibility varies depending on the state space.

<sup>8</sup>The Metropolis-Hastings algorithm may be applied in either of these situations.



## Mixing time of the Markov chain

Note that a simulation's definite initial state  $X_0 = \alpha$  means that the initial distribution  $\mu^{(0)}$  will always be of the form  $\mu^{(0)}(\alpha) = 1$  for some  $\alpha \in \Omega$  and  $\mu^{(0)}(\omega) = 0$  for all  $\omega \in \Omega, \omega \neq \alpha$ . Fact 1.1 states that, regardless of which  $\alpha \in \Omega$  is selected for  $X_0$ ,  $\mu^{(t)}$  converges to the stationary distribution  $\pi$  in the limit. Thus for some appropriately large simulation-time  $T$ , the final distribution  $\mu^{(T)}$  (from which  $X_T$  is effectively drawn) is “close enough” to  $\pi$ . Here, “close enough” means that  $d_{\text{TV}}(\mu^{(T)}, \pi) \leq \epsilon$  for some desired *tolerance*  $\epsilon$ .

What is not stated in Fact 1.1, yet what is nonetheless true, is that the rate of convergence (i.e., how quickly a simulation reaches “close enough”) generally does depend on  $\alpha$ . Thus the rate of convergence, or *mixing time*, of  $\mathcal{X}$  when started in  $X_0 = \alpha \in \Omega$  is defined for any tolerance  $\epsilon$  as

$$\tau_\alpha(\epsilon) = \min\{t : d_{\text{TV}}(\mu^{(t)}, \pi) \leq \epsilon\}.$$

An important property to note here is that  $d_{\text{TV}}(\mu^{(t)}, \pi)$  is a non-increasing function of  $t$ .<sup>9</sup> In other words, once a simulation's current distribution is “close enough” to  $\pi$ , no future distribution will ever be “not close enough.”

The key to efficient sampling with MCMC is finding a polynomial upper-bound on the mixing time  $\tau_\alpha$ . When this is accomplished, the Markov chain is said to be *rapidly mixing*. The upper-bound should be polynomial in “the size of the input” which, for our purpose of sampling edge covers, would be a polynomial in  $n = |V|$  and  $m = |E|$  of the input graph  $G = (V, E)$  that defines the set of edge covers from which we would like to sample. In the next chapter, we develop such an upper-bound using the *canonical flow* approach, a technique originated by Diaconis and Stroock [5] and developed by Jerrum and Sinclair [12].

---

<sup>9</sup>For details, see Jerrum's text ([14], Lemma 4.2).

# Chapter 2

## Efficiently sampling $(1, 2)$ -edge covers

Let a  $(1, 2)$ -edge cover be an edge cover where each vertex is covered by either exactly one or exactly two edges.<sup>1</sup> Say that a  $(1, 2)$ -edge cover has one *pile* when some edge covers some vertex that is already covered by at least two other edges. Let a *near- $(1, 2)$ -edge cover* be a  $(1, 2)$ -edge cover having either exactly one pile or exactly two piles.

Let  $G = (V, E)$  be any finite simple graph with no isolated vertices. Denote the set of all  $(1, 2)$ -edge covers in  $G$  as  $\mathcal{P}$ , and denote the set of all near- $(1, 2)$ -edge covers in  $G$  as  $\mathcal{N}$ . Let  $\mathcal{X} = (X_t)_{t=0}^\infty$  be a lazy Markov chain on state space  $\Omega = \mathcal{P} \cup \mathcal{N}$  whose transition matrix is defined by the trial in Figure 2.1.

We now make a crucial assumption that the ratio of  $(1, 2)$ -edge covers to near- $(1, 2)$ -edge covers is bounded by a polynomial in the order  $n = |V|$  of  $G$ ; that is,  $\frac{|\mathcal{N}|}{|\mathcal{P}|} \in O(n^k)$  for some constant  $k$ .<sup>2</sup> With this restriction and with efficient uniform sampling over  $\Omega$ , one may efficiently sample uniformly over  $\mathcal{P}$  by taking  $p(n)$  samples from  $\Omega$  for every one desired sample from  $\mathcal{P}$ . Note that we are also implicitly assuming that  $\mathcal{P}$  is non-empty, which can be efficiently verified using an algorithm that generates a  $(1, 2)$ -edge cover in  $G$  if and only if  $\mathcal{P}$  is non-empty. (This issue is discussed in Section 2.4.)

---

<sup>1</sup>A  $(1, 2)$ -edge cover may also be characterized as a spanning collection of disjoint paths and cycles.

<sup>2</sup>Note that there are graphs for which the ratio  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  is exponential. For example, one such family of graphs is sketched in Figure 2.2.

1. With probability  $\frac{1}{2}$ , set  $X_{t+1} \leftarrow X_t$  and stop.
2. Uniformly at random, select an edge  $e \in E \supseteq W$ .
3. There are four exhaustive and mutually exclusive possibilities:
  - ( $\uparrow$ ) If  $e \notin W$ , then set  $W' \leftarrow W \cup \{e\}$ .
  - ( $\downarrow$ ) If  $e \in W$  and both endpoints of  $e$  are covered in  $W \setminus \{e\}$ , then set  $W' \leftarrow W \setminus \{e\}$ .
  - ( $\leftrightarrow$ ) If  $e \in W$  and exactly one endpoint  $v$  of  $e$  is uncovered in  $W \setminus \{e\}$ , then:
    - a. Let  $F$  be the set of all edges incident on  $v$ .
    - b. Uniformly at random, select an edge  $f \in F$ .
    - c. Set  $W' \leftarrow (W \setminus \{e\}) \cup \{f\}$ .
  - ( $\circ$ ) If  $e \in W$  and neither endpoint of  $e$  is covered in  $W \setminus \{e\}$ , then set  $W' \leftarrow W$ .
4. If  $W' \in \Omega$ , then set  $X_{t+1} \leftarrow W'$  and stop; otherwise, set  $X_{t+1} \leftarrow W$  and stop.

Figure 2.1: Trial defining the transition matrix of a lazy Markov chain  $\mathcal{X} = (X_t)_{t=0}^\infty$  over  $\Omega = \mathcal{P} \cup \mathcal{N}$  in a finite simple graph  $G = (V, E)$  with no isolated vertices. The current state and edge cover are respectively denoted  $X_t = W$ .

We will begin by showing that  $\mathcal{X}$  is an ergodic Markov chain having uniform stationarity. Aperiodicity of  $\mathcal{X}$  is obvious; irreducibility will be shown by defining a canonical flow between any two states  $I, F \in \Omega$ . With ergodicity, uniform stationarity of  $\mathcal{X}$  is immediate by Fact 1.2 since each transition of  $\mathcal{X}$  is symmetric.

We will then analyze this canonical flow to determine a polynomial upper bound on the congestion of  $\mathcal{X}$ , thus bounding the mixing time of  $\mathcal{X}$  by a polynomial. The upper bound on mixing time will be a polynomial in the order  $n = |V|$ , the size  $m = |E|$ , and the maximum degree  $\Delta = \max_{v \in V} \{\deg(v)\}$  of  $G$ . This will show that it is possible to use a Markov chain Monte Carlo approach to efficiently sample  $(1, 2)$ -edge covers in arbitrary finite simple graphs with no isolated vertices.

## 2.1 Defining the canonical flow

For any vertex  $v \in V$ , let  $N_e(v)$  denote the set of all edges incident on  $v$ . Let  $I, F \in \Omega$  be any two states of  $\mathcal{X}$ , and let  $I \oplus F$  denote their symmetric difference. With respect to  $I$  and

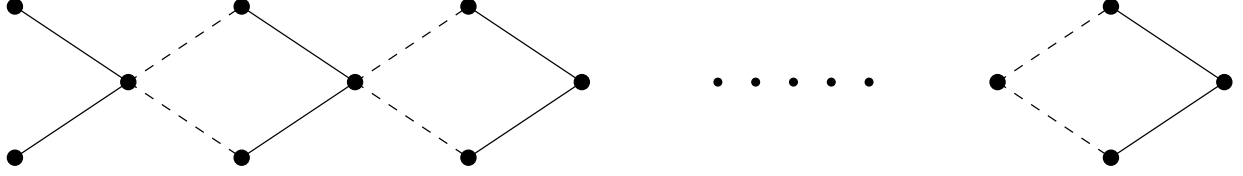


Figure 2.2: Graph family for which the ratio of near-(1,2)-edge covers-to-(1,2)-edge covers  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  is exponential. The dots indicate indefinite continuation of the chained four-cycles. The solid lines indicate the only (1,2)-edge cover, while the dashed lines indicate other edges. For any amount of chaining, there exists exactly one (1,2)-edge cover; whereas, the number of near-(1,2)-edge covers grows exponentially with the addition of each four-cycle.

$F$ , let  $N_I(v) = (I \setminus F) \cap N_e(v)$  and  $N_F(v) = (F \setminus I) \cap N_e(v)$ . For each vertex  $v$  of  $G$ , form a maximal collection of disjoint edge-pairs where each pair consists of one edge from  $N_I(v)$  and one edge from  $N_F(v)$ . We call such a collection a *pairing*. Let  $D_v = \max(|N_I(v)|, |N_F(v)|)$  and  $d_v = \min(|N_I(v)|, |N_F(v)|)$ ; then, the number of possible pairings at any vertex  $v$  is

$$\binom{D_v}{d_v} d_v!$$

A *pairing system* is a collection consisting of exactly one pairing for each vertex. Denote the set of all possible pairing systems as  $\Psi_{I,F}$ . The number of possible pairing systems is

$$|\Psi_{I,F}| = \prod_{v \in V} \binom{D_v}{d_v} d_v!$$

Any pairing system gives rise to a set of edge-disjoint (but not necessarily vertex-disjoint) trails and circuits.<sup>3</sup> Each trail or circuit, called a *component*, consists of alternating edges from  $I$  and  $F$ .

For a pairing system  $\psi \in \Psi_{I,F}$  and its corresponding components, we want to “process” the edges of each component in sequence, using the specific transitions of  $\mathcal{X}$  in such a way so that the complete processing of all components represents a directed path from  $I$  to  $F$  in the transition graph of  $\mathcal{X}$ . To this end, let  $V$  and  $E$  each already possess some fixed and

---

<sup>3</sup>The trails and circuits arise by chaining together overlapping edge-pairs from the pairings of adjacent vertices.

arbitrary ordering of their elements, and order each component of  $\psi$  by the lowest-ordered edge it contains.

### 2.1.1 Processing the components of $\psi$

**Trails** To process the edges in a trail, start from its lower-ordered end-edge. Four patterns of edge-alternation may occur:

1.  $IFI \dots FIF$  : Process each  $IF$ -pair of consecutive edges in one of two ways: if the pair's shared vertex would be uncovered without the  $I$ -edge, then use one  $(\leftrightarrow)$ -transition for both edges; otherwise, use one  $(\downarrow)$ -transition for the  $I$ -edge followed by one  $(\uparrow)$ -transition for the  $F$ -edge.
2.  $IFI \dots FI$  : Process as pattern 1, except that the last  $I$ -edge is processed with one  $(\downarrow)$ -transition.
3.  $FIF \dots IF$  : Process as pattern 1, except that the first  $F$ -edge is processed with one  $(\uparrow)$ -transition.
4.  $FIF \dots IFI$  : Process as pattern 1, except that the first  $F$ -edge and the last  $I$ -edge are processed as in patterns 3 and 2, respectively.

It is clear that each vertex incident on an  $F$ -edge in a trail remains covered throughout processing; however, it is less clear that an end-vertex always remains covered when the edge incident on that vertex is an  $I$ -edge. This is nonetheless true, and may be seen to be so by noting that exactly one of two conditions must be true about each end-vertex: either (a) the vertex has a non-empty pairing, in which case the vertex will always be covered by exactly one of the edges from at least one of the edge-pairs in the pairing, or (b) the vertex has an empty pairing, in which case there must be at least one edge in  $I \cap F$  that is incident on the vertex.

The number of piles is never increased by more than two while processing a trail. Furthermore, the absolute difference between the number of piles before processing a trail and the number of piles after processing a trail is no greater than two. (The addition or removal of these piles can only take place on the start- and stop-vertices via the start- and stop-edges.) The actual difference will depend on the trail's end-edges and the current edge-coverage at its end-vertices.

**Circuits** To process the edges in a circuit, start with its lowest-ordered  $F$ -edge and from that edge's lower-ordered endpoint. In the direction of the higher-ordered endpoint, continue with the  $IF \dots IFI$  edge-sequence through the remainder of the circuit. Process this entire  $FIF \dots IFI$  edge-sequence as pattern 4 in processing a trail. It is clear that each vertex in a circuit remains covered throughout processing.

The number of piles is never increased by more than two while processing a circuit, and the number of piles before processing a circuit is equal to the number of piles after processing a circuit.

### 2.1.2 The flow, congestion, and connectedness of the state space

Let  $\mathcal{P}_{I,F}$  be the set of all directed paths from  $I$  to  $F$  in the transition graph of  $\mathcal{X}$ . We define a *flow* in that transition graph to be a function

$$f : \bigcup_{I,F \in \Omega, I \neq F} \mathcal{P}_{I,F} \rightarrow \mathbb{R}_0^+$$

such that  $\sum_{p \in \mathcal{P}_{I,F}} f(p) = \pi(I)\pi(F)$ . Recall that  $\pi$  denotes the stationary distribution of  $\mathcal{X}$ , which in our case is the uniform distribution. Thus  $\pi(I)\pi(F)$  is simply  $|\Omega|^{-2}$ , and this is the total flow value that must be distributed over  $I$ -to- $F$  paths defined by the canonical flow.<sup>4</sup>

We want to route this flow “evenly”, which is to say that we want to route it in such a way as

---

<sup>4</sup>There may be some directed paths between  $I$  and  $F$  that are not possible to obtain within our canonical flow definition. This is acceptable, since these paths are given  $I$ -to- $F$  flow values of zero.

to avoid creating “congested” transitions in the transition graph of  $\mathcal{X}$ . Therefore, we need a measure of the *congestion*  $\rho(f)$  caused by our canonical flow  $f$ . From Sinclair [17], we know this allows us to bound the mixing time  $\tau_\alpha(\epsilon)$  of  $\mathcal{X}$  in terms of the congestion as

$$\tau_\alpha(\epsilon) \leq \rho(f)(\log \pi(\alpha)^{-1} + \log \epsilon^{-1}) \quad (2.1)$$

where  $\alpha \in \Omega$  is a starting state for  $\mathcal{X}$  and  $\epsilon$  is the tolerance or *sampling error* determining how close our approximately uniform sampling is to exactly uniform sampling. The congestion  $\rho$  of the flow  $f$  is defined as

$$\rho(f) = \max_{T=(W,W')} \left\{ \underbrace{\frac{1}{\pi(W)P(W,W')}}_{(\text{capacity of } T)^{-1}} \underbrace{\sum_{p \ni T} f(p) |p|}_{\text{total flow through } T} \right\}$$

where the maximum is taken over all positive-probability transitions (where  $P(W, W') > 0$ ), the summation is taken over all paths  $p \in \cup_{I,F} \mathcal{P}_{I,F}$  that contain the transition  $T$ , and  $|p|$  is the number of transitions (i.e. edges) in the path  $p$ .

### The canonical flow

We now describe a canonical flow between all pairs  $I \in \Omega$ ,  $F \in \mathcal{P}$  by detailing the two possible cases thereof:  $I \in \mathcal{P}$ ,  $F \in \mathcal{P}$  and  $I \in \mathcal{N}$ ,  $F \in \mathcal{P}$ . We present only these two cases since this flow can be extended to a flow between all pairs  $I, F \in \Omega$  by reversing the flow-description when  $I \in \mathcal{P}$ ,  $F \in \mathcal{N}$  and by routing through a random intermediate state  $M \in \mathcal{P}$  when  $I \in \mathcal{N}$ ,  $F \in \mathcal{N}$ .<sup>5</sup> We present these two cases since processing the components of a pairing system  $\psi \in \Psi_{I,F}$  may change the number of piles in an intermediate edge cover  $M \in \Omega$  along the  $I$ -to- $F$  path derived from the components of  $\psi$ . We must ensure that the number of piles never exceeds two; if it were to exceed two, then the intermediate edge cover

---

<sup>5</sup>This can be safely done under our working assumption that the ratio  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  is upper-bounded by a polynomial in the order of the input graph.

at that point would no longer be a  $(1, 2)$ -edge cover or a near- $(1, 2)$ -edge cover and would thus be an illegal state, outside the state space of the Markov chain.

$I, F \in \mathcal{P}$  : Components are processed in their original arbitrary ordering; that is, in order of the lowest-ordered edge contained in each. This can be done because the intermediate state before or after the processing of any component is always a  $(1, 2)$ -edge cover with no piles and because there can be no more than two additional piles during the processing of any component. This ensures that all intermediate states are within the state space.

To see that this is the case, consider the following. It is clear that the processing of a circuit cannot produce an illegal intermediate edge cover when the edge cover immediately prior to that processing was a  $(1, 2)$ -edge cover. In this case, where both  $I$  and  $F$  are  $(1, 2)$ -edge covers, trail-processing has the same property. This is because a trail's start- and stop-vertices must have certain kinds of edge-coverage, as illustrated in Figure 2.3: type C or G for  $I$ -end-edges; or type D or H for  $F$ -end-edges. It is clear that, for these pairs of coverage types, the respective removal of an  $I$ -end-edge or addition of an  $F$ -end-edge cannot remove or add piles at the end-vertices. Since the end-vertices are the only places where piles may be removed or added, we have shown that the state before or after the processing of any trail or circuit is always a  $(1, 2)$ -edge cover.

Now that the validity of the canonical  $I$ -to- $F$  paths is established, we allocate the flow between  $I$  and  $F$  by assigning each path a flow value of

$$\frac{\pi(I)\pi(F)}{|\Psi_{I,F}|} = \frac{1}{|\Omega|^2 |\Psi_{I,F}|}.$$

Since different pairing systems correspond to different sets of components which, in turn, correspond to different  $I$ -to- $F$  paths, we have that each pairing system in  $\Psi_{I,F}$  corresponds to a unique  $I$ -to- $F$  path, and so the total flow  $\pi(I)\pi(F) = |\Omega|^{-2}$  between  $I$  and  $F$  is wholly and evenly distributed.



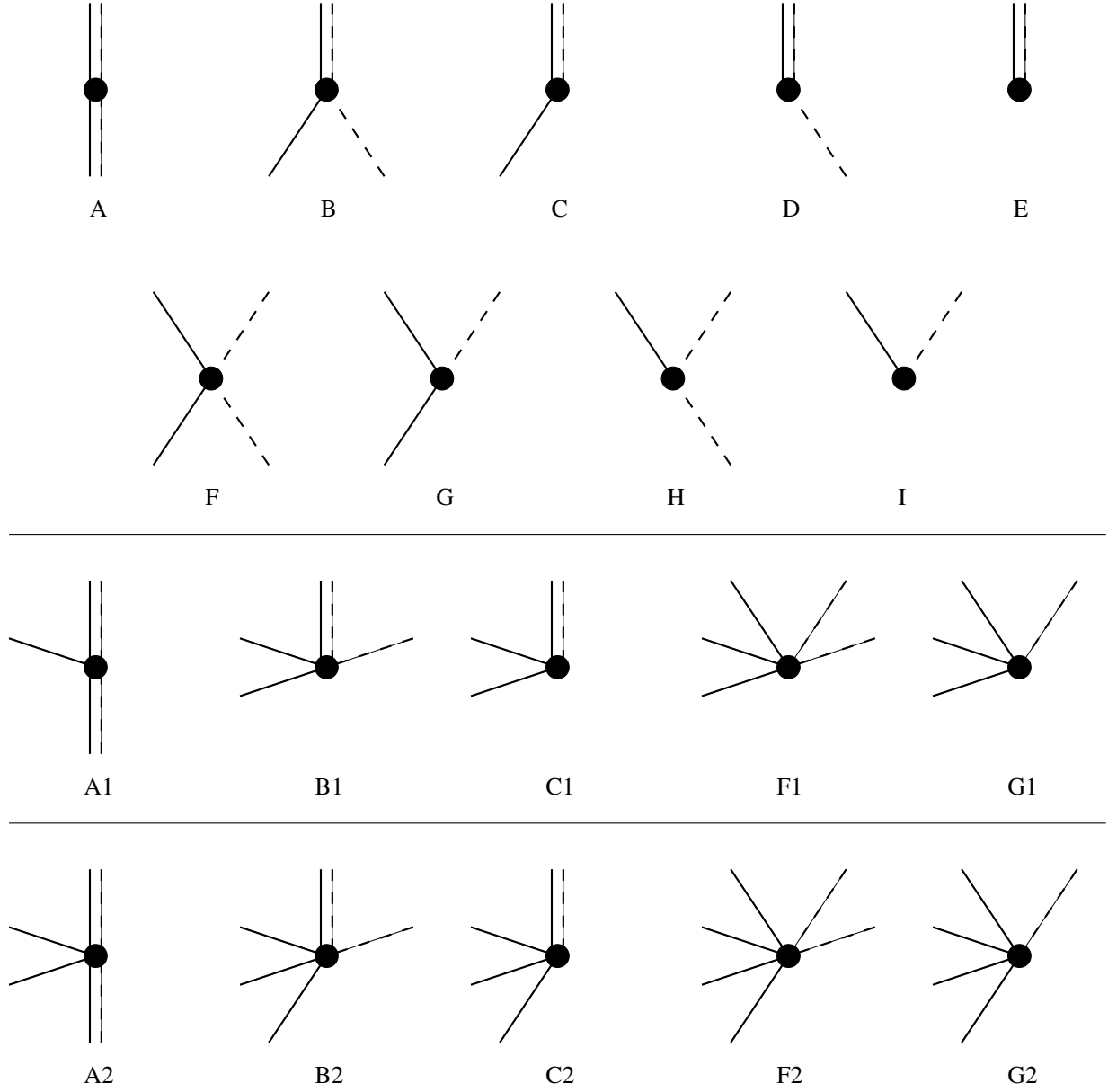


Figure 2.3: Types of  $(1,2)$ -edge-coverage with respect to  $(I, F) \in \Omega \times \mathcal{P}$ .  $I$ -edges and  $F$ -edges are indicated by solid lines and dashed lines, respectively. Types A–I are *non-piled* coverages; types A1, B1, C1, F1, and G1 are *one-piled* coverages; and types A2, B2, C2, F2, and G2 are *two-piled* coverages. When  $I \in \mathcal{P}$ , only non-piled coverages may occur. When  $I \in \mathcal{N}$ , either (a) exactly one one-piled coverage occurs, (b) exactly two one-piled coverages occur, or (c) exactly one two-piled coverage occurs; all other coverages are non-piled.

$I \in \mathcal{N}, F \in \mathcal{P}$ : Here we must be more careful to not exceed two piles in an intermediate state. For example, suppose we begin component processing with a circuit. Now suppose there is a vertex (not on that circuit for the sake of simplicity) with a one-piled or two-piled edge-coverage, and we introduce the maximum of two additional piles during the processing of that circuit. In this event, we have more than two piles and thus an illegal intermediate state.

To overcome this potential problem, we must give priority to certain components in the order of processing. These components are every trail with a piled edge-coverage on at least one of its end-vertices. We process these *pile-trails* before any other components, but otherwise in their original relative ordering. There may be either one or two pile-trails. If there is one pile, then there is only one pile-trail; if there are two piles, then either there is one pile-trail (of pattern 2) having both piles, or there are two pile-trails.

The way in which each pile-trail is processed must also be slightly changed, so that the end-edge with which processing begins is an edge that is causing a pile. (Note that any pile-causing edge is necessarily an  $I$ -edge since  $I \in \mathcal{N}$  and  $F \in \mathcal{P}$ .) If both end-edges of the pile-trail are causing piles, the lower-ordered one is selected. By processing the pile-trails in this manner, we ensure that one pile is removed at the very start of the processing of each pile-trail. Consequently, any additional pile occurring during processing along the rest of the trail is offset, and we are guaranteed to avoid illegal intermediate states. After every pile-trail is processed, all other components may be processed in their original relative ordering.

As when both  $I$  and  $F$  are in  $\mathcal{P}$ , we allocate the flow between them here by assigning each path a flow value of  $(|\Omega|^2 |\Psi_{I,F}|)^{-1}$ .

**Connectedness** Since we have described a reversible sequence of transitions to take any  $I \in \Omega$  and obtain any  $F \in \mathcal{P}$ , we know that there is a path in the transition graph of  $\mathcal{X}$  between any two states as long as there is at least one element in  $\mathcal{P}$ . As a consequence, we have shown that the state space  $\Omega$  is connected and that the Markov chain  $\mathcal{X}$  is irreducible.

Since  $\mathcal{X}$  is then ergodic, we know by the symmetry of its transition probabilities and by Fact 1.2 that the unique stationary distribution of  $\mathcal{X}$  is uniform and that we are sampling  $(1, 2)$ -edge covers in the manner we desire.

## 2.2 Analyzing the canonical flow

We will now bound the congestion of the flow described above by defining an encoding function and showing that it has certain properties. Once the congestion is upper-bounded by a polynomial in the size ( $n$ ,  $m$ , and  $\Delta$ ) of the input graph, a bound on the mixing time of  $\mathcal{X}$  is easily shown.

Let  $\text{cp}(T)$  denote the set of all  $(I, F, \psi)$  such that the pairing system  $\psi \in \Psi_{I,F}$  defines a directed path from  $I$  to  $F$  through transition  $T$  in the transition graph of  $\mathcal{X}$ . We can simplify the expression for the congestion  $\rho(f)$  of our canonical flow as follows:

$$\begin{aligned} \rho(f) &= \max_{T=(W,W')} \left\{ \frac{1}{\pi(W)P(W,W')} \sum_{p \ni T} f(p) |p| \right\} \\ &\leq |\Omega| \, 2\Delta m^2 \max_T \sum_{p \ni T} f(p) \end{aligned} \tag{2.2}$$

$$= |\Omega| \, 2\Delta m^2 \max_T \sum_{(I,F,\psi) \in \text{cp}(T)} \frac{1}{|\Omega|^2 |\Psi_{I,F}|} \tag{2.3}$$

$$= \frac{2\Delta m^2}{|\Omega|} \max_T \sum_{(I,F,\psi) \in \text{cp}(T)} \frac{1}{|\Psi_{I,F}|} \tag{2.4}$$

Inequality 2.2 is obtained from the following facts:  $\pi(W) = |\Omega|^{-1}$  for all  $W \in \Omega$  since  $\pi$  is uniform;  $P(W, W')^{-1}$  is largest for  $(\leftrightarrow)$ -transitions, when it is  $2\Delta m$ ; and the length  $|p|$  of any path is at most  $m = |E|$ , since no more than  $m$  edges could be changed along any canonical path from  $I$  to  $F$ . Equation 2.3 is due to the one-to-one correspondence between each path that uses  $T$  and the  $(I, F, \psi)$  triple in  $\text{cf}(T)$  that produces it. Equation 2.4 simply extracts the factor of  $|\Omega|^{-2}$  from the summation.

### 2.2.1 The encoding function

In order to bound the summation, we will define an encoding function  $\eta_T : \text{cp}(T) \rightarrow \text{enc}(T)$  for each transition  $T = (W, W')$ :

$$\eta_T(I, F, \psi) = (I \oplus F \oplus (W \cup W'), \psi)$$

The function's range  $\text{enc}(T)$  is a set of pairs (called *encodings*), the first element of each being a  $(1, 2)$ -edge cover or near- $(1, 2)$ -edge cover that is allowed to leave at most two vertices uncovered and the second element of each being the pairing system given as input.

#### The range $\text{enc}(T)$ of $\eta_T$

Let  $\Omega'$  denote the set of all  $(1, 2)$ -edge covers and near- $(1, 2)$ -edge covers from  $G$  and from each 1-, 2-, 3-, and 4-vertex-deleted subgraph of  $G$ :

$$\Omega' = \Omega \cup \bigcup_{u,v,w,x \in V} \Omega_{G-\{u,v,w,x\}}$$

Observe that deleting a vertex and all of its incident edges cannot produce a graph with more edge covers than its original, which implies

$$|\Omega'| \leq (n^4 + 1)|\Omega| \tag{2.5}$$

This modified state space  $\Omega'$  is the set from which the first element of each pair in  $\text{enc}(T)$  is taken. In other words:

**Claim 2.1.** *For all transitions  $T$  and all triples  $(I, F, \psi) \in \text{cp}(T)$ , the encoding  $(C, \psi) = \eta_T(I, F, \psi)$  is such that  $C \in \Omega'$ .*

*Proof.* Consider the construction of  $C = I \oplus F \oplus (W \cup W')$ . The symmetric difference  $I \oplus F$  comprises all edges where  $I$  and  $F$  disagree and no edges from  $I \cap F$ . The union  $W \cup W'$

comprises all “current” (with respect to the canonical path defined by the input triple) edges in  $I \oplus F$  plus the one or two involved in the transition. It also contains all edges from  $I \cap F$ . Thus, we know that  $C$  comprises all edges from  $I \cap F$  plus all non-“current” edges in  $I \oplus F$ , not including the one edge or two edges involved in the transition. This tells us that  $C$  covers all vertices outside of  $I \oplus F$  as both  $I$  and  $F$  already do: as a  $(1, 2)$ -edge cover or near- $(1, 2)$ -edge cover.

We now examine the edge-coverage of vertices incident to edges from  $I \oplus F$ . As described above, the basic effect of the encoding function is to invert the “current” edges within the symmetric difference. Suppose we take any  $(I \oplus F)$ -vertex  $v$  that *is not* incident to any edge involved in the transition. By examining the types of edge-coverage in Figure 2.3, we can see that the inversion of any coverage on  $v$  at any moment of processing cannot produce an illegal, non- $(1, 2)$ -edge coverage. Furthermore, those inversions that would produce piles can only occur when  $v$  was piled to begin with and can only produce piles that have been removed from  $v$ , thus guaranteeing that  $C$  has no more piles than  $I$ .

There is a special case, however, and that is when  $v$  is the start-vertex of a circuit that is currently being processed. As defined in the canonical flow, circuits begin by processing the first  $F$ -edge with an  $(\uparrow)$ -transition. This may cause  $v$  to be uncovered in  $C$  when  $v$  has no other incident  $(I \cap F)$ -edges, processed  $I$ -edges, or unprocessed  $F$ -edges. Therefore, there may be at most one uncovered vertex that is not incident to any transition edges.

Now let us consider  $(I \oplus F)$ -vertices that *are* incident to an edge involved in the transition. The former reasoning also applies here, but we must be aware of the fact that the edge or edges involved in the transition will be *absent* from  $C$ , thus opening the possibility that one or more of these vertices are uncovered in  $C$ . Regardless of the transition type, each of the two (for  $(\downarrow)$ - or  $(\uparrow)$ -transitions) or three (for  $(\leftrightarrow)$ -transitions) vertices incident on transition edges is covered in  $C$  if and only if it is incident to at least one of the following: an  $(I \cap F)$ -edge, a processed  $I$ -edge, or an unprocessed  $F$ -edge. Therefore, there may be at most three uncovered vertices that are incident to transition edges.

As a result of the above, we know that there can be no more than four uncovered vertices in  $C$  and that  $C$  is a valid  $(1, 2)$ -edge cover or near- $(1, 2)$ -edge cover otherwise. Thus  $C \in \Omega'$ , and our claim is proven.  $\square$

Now we will define the set from which the second element  $\psi'$  (the pairing system) of each pair  $(C, \psi') \in \text{enc}(T)$  is drawn. It is clear that this set is actually  $\Psi_{I,F}$ , but we want to define the set in terms of the encoding  $C$  since the decomposition of  $I \oplus F$  into  $I$ - and  $F$ -edges is unknown without using the pairing system itself. Let  $\Psi'_{C,T}$  denote the set of all pairing systems that can be derived from each possible decomposition of  $I \oplus F$  into  $I$ - and  $F$ -edges with respect to our knowledge of  $C$  and  $T = (W, W')$ .

In order to proceed, we want to upper-bound  $|\Psi'_{C,T}|$  by a polynomial factor of  $|\Psi_{I,F}|$ . Note that  $I \cap F$  and  $I \oplus F$  are uniquely reconstructable from  $C$  without knowledge of  $\psi'$ , since  $I \oplus F = C \oplus (W \cup W')$  where  $T = (W, W')$  is known, and since  $I \cap F = C \setminus (I \oplus F)$ . Having this knowledge, we may now assign a “signature”  $x \cdot y$  to each vertex  $v$  in  $I \oplus F$ , consisting of the number  $x$  of  $(I \cap F)$ -edges incident on  $v$  and the number  $y$  of  $(I \oplus F)$ -edges incident on  $v$ . Each possible signature is listed in Figure 2.4. With each signature, we can deduce some information on how many pairings we need to account for at that vertex when constructing  $\Psi'_{C,T}$  and determining its size. This information does not always allow for unambiguous determination of which pairings we must include; hence, we include any pairing that is possible at that vertex for any type of original  $(I, F)$ -edge-coverage it may have had. In some cases, this means that we will be including more pairings than were actually present at a vertex when  $\Psi_{I,F}$  was constructed.

If differences of this sort occur on more than a constant number of vertices, then  $\Psi'_{C,T}$  will be exponentially larger than  $\Psi_{I,F}$ ; however, such a scenario will not happen. In Figure 2.4, we have three signatures  $(1 \cdot 2, 0 \cdot 4, \text{ and } 0 \cdot 3)$  that can both increase the size of  $\Psi'_{C,T}$  and occur more than a constant number of times. When they do occur, though, only a small constant number of them can be in a situation where we must account for more pairings than necessary. For each signature, this depends on an analysis of the current coverage in

Signature		Coverage		Pairings			Conditions for non-identity Factors from non-piled Types
$I \cap F$	$I \oplus F$	Type	$W' \cap (I \oplus F)$	Max	Min	Factor	
2	0	A	0	1	1	1	
2	1	A1	1	1	1	1	
			0	1	1	1	
2	2	A2	2	1	1	1	
			1	1	1	1	
			0	1	1	1	
1	0	E	0	1	1	1	
0	2	I	2	1	1	1	
			1	1	1	1	
1	4	B2	3	6	3	2	
			2	6	3	2	
			1	6	3	2	
			0	6	3	2	
0	6	F2	4	45	12	3.75	
			3	45	12	3.75	
			2	45	12	3.75	
			1	45	12	3.75	
1	3	B1,C2	3	4	1	4	
			2	4	1	4	
			1	4	1	4	
			0	4	1	4	
0	5	F1,G2	4	25	4	6.25	
			3	25	4	6.25	
			2	25	4	6.25	
			1	25	4	6.25	
1	2	B,C1	2	2	1	2	$B_{\text{SoC}}, B_{\text{HoP}}, C1$
			1	2	1	2	$B_{(1)}, C1$
			0	2	1	2	$B_{\text{HoP}}, C1$
0	4	F,G1	4	9	2	4.5	$F_{\text{SoC} \cdot \text{HoP}}, G1$
			3	9	2	4.5	$F_{\text{SoC}}, F_{\text{HoP}}, G1$
			2	9	2	4.5	$F_{\text{SoC} \cdot \text{HoP}}, F_{(1)}, G1$
			1	9	2	4.5	$F_{\text{HoP}}, G1$
1	1	C,D	1	1	1	1	
			0	1	1	1	
0	3	G,H	3	3	2	1.5	$(G,H)_{\text{SoC}}, (G,H)_{\text{HoP}}$
			2	3	2	1.5	$(G,H)_{\text{SoC}}, (G,H)_{\text{HoP}}, (G,H)_{(1)}$
			1	3	2	1.5	$(G,H)_{\text{HoP}}, (G,H)_{(1)}$

Figure 2.4: Every possible signature and edge-coverage for vertices in  $I \oplus F$  at any moment in component processing. The “Max” and “Min” columns respectively denote the maximum and minimum numbers of pairings that need to be accounted for at that vertex; the “Factor” column denotes the corresponding multiplicative increase of an upper bound on  $|\Psi'_{C,T}|$  in terms of  $|\Psi_{I,F}|$ .

$W'$  where the transition under examination is  $T = (W, W')$ .

This analysis is summarized in the last column of Figure 2.4. The notation  $X_{\text{SoC}}$  denotes that a vertex with the corresponding signature, coverage, and Factor-value can be of type X only if it is “Start of Cycle”, i.e. the start-vertex of a cycle currently being processed. The notation  $X_{\text{HoP}}$  denotes that a vertex with the corresponding signature, coverage, and Factor-value can be of type X only if it is “Head of Processing”, i.e. the vertex is at the leading intermediate point of processing in the current component. The notation  $X_{\text{SoC-HoP}}$  denotes that a vertex with the corresponding signature, coverage, and Factor-value can be of type X only if it is *both* “Start of Cycle” and “Head of Processing”. The notation  $X_{(1)}$  denotes that the vertex with the corresponding signature and coverage can be of type X *with no restrictions* only if its Factor-value is identity. The notation C1 or G1 denotes that a vertex with the corresponding signature, coverage, and Factor-value can be of type C1 or G1, clearly limited by the fact that there can no more than two piles.

As an example of how this analysis is conducted, let us refer to the case when a vertex  $v$  has the signature  $0 \cdot 4$  and a coverage of two edges in  $W'$ . First, we determine that the minimum number of pairings to be accounted for is two, when the original edge-coverage on  $v$  is of type F. Next, we determine the maximum number of distinct pairings that can occur at  $v$ .

- If  $v$  was of type G1 to begin with, then there are at most six possible pairings, one for each way a single edge-pair can be taken from four edges.
- If  $v$  was of type F, then there are at most three possible pairings, one for each way two edge-pairs can be taken from four edges. In this case, however, the third pairing (when the two “on”-edges at  $v$  are paired together and the two “off”-edges at  $v$  are paired together) can only be a possibility when  $v$  is both “Start of Cycle” and “Head of Processing”; otherwise, each “on”-edge must be paired with an “off-edge”, giving rise to just two possible pairings.



Now, we immediately know that we must account for the six pairings of a type-G1 and the three pairings of a special type-F at most three times among all vertices with  $0 \cdot 4$  signatures and covered by two edges — once for  $B_{\text{SoC-HoP}}$  and twice for the one-piled edge-coverage G1. All other such vertices will be of type F with two possible pairings. Thus we know that the corresponding non-identity Factor-value of 4.5 will occur no more than three times in no more than  $\binom{n}{3}$  ways throughout the  $n$  vertices in the graph.

As summarized in Figure 2.4, similar analyses hold for every other case when a non-identity Factor-value is obtainable from a non-piled Type. In order to find an upper-bound on the size of  $\Psi'_{C,T}$ , we will exclude from consideration all signatures and coverages with (a) identity Factor-values, (b) only piled coverage types, or (c) no condition of the form  $X_{(1)}$ . The reason that the latter two categories are excluded is because they allow signatures and coverages that must be described by the special cases of SoC, HoP, and piled coverage. For instance, in the above analysis of a  $0 \cdot 4$  signature with two-edge coverage, if we knew there was another  $0 \cdot 4$  signature but with four-edge coverage, we would have to describe it as  $F_{\text{SoC-HoP}}$  or as G1. This would modify our conclusion in the last paragraph: the Factor-value of 4.5 could occur no more than *two* times in no more than  $\binom{n}{2}$  ways, thus eliminating an entire factor of  $n$  from the upper bound.

Finally, we are ready to derive an upper bound on  $|\Psi'_{C,T}|$  using the four signature-coverage cases having the  $X_{(1)}$ -form condition.

- For signature  $1 \cdot 2$  with one-edge coverage, the Factor-value of 2 can occur no more than two times in no more than  $\binom{n}{2}$  ways.
- For signature  $0 \cdot 4$  with two-edge coverage, the Factor-value of 4.5 can occur no more than three times in no more than  $\binom{n}{3}$  ways.
- For signature  $0 \cdot 3$  with two-edge coverage, the Factor-value of 1.5 can occur no more than two times in no more than  $\binom{n}{2}$  ways.
- For signature  $0 \cdot 3$  with one-edge coverage, the Factor-value of 1.5 can occur no more

than one time in no more than  $n$  ways.

From this, we have

$$\begin{aligned} |\Psi'_{C,T}| &\leq 2^2 \binom{n}{2} \cdot 4.5^3 \binom{n}{3} \cdot 1.5^2 \binom{n}{2} \cdot 1.5n \cdot |\Psi_{I,F}| \\ &\leq 103n^4(n-1)^3(n-2)|\Psi_{I,F}| \end{aligned} \tag{2.6}$$

This inequality is critical in the proof of Theorem 2.5, our main result.

And we are now led to the following claim, analogous to Claim 2.1 but stated for the second element of each pair in  $\text{enc}(T)$ .

**Claim 2.2.** *For all transitions  $T$  and all triples  $(I, F, \psi) \in \text{cp}(T)$ , the encoding  $(C, \psi) = \eta_T(I, F, \psi)$  is such that  $\psi \in \Psi'_{C,T}$ .*

*Proof.* The claim is immediate since the set  $\Psi_{I,F}$  to which  $\psi$  belongs is a subset of  $\Psi'_{C,T}$ .  $\square$

The range of  $\eta_T$  may now be characterized. Let  $\text{enc}(T)$  denote the set of all  $(C, \psi')$  such that the pairing system  $\psi' \in \Psi'_{C,T}$  defines a directed path from some  $I$  to some  $F$  through transition  $T$  in the transition graph of  $\mathcal{X}$ , where the pair  $(I, F)$  is any valid derivation from the symmetric difference  $I \oplus F$  uniquely determined by  $C \in \Omega'$ .

### The injectivity of $\eta_T$

The last piece of our puzzle is to show that  $\eta_T$  is an injective function. Once this has been proven, we will be ready to bound the congestion and, thus, the mixing time of our Markov chain  $\mathcal{X}$ .

**Claim 2.3.** *For all transitions  $T$ , the function  $\eta_T : \text{cf}(T) \rightarrow \text{enc}(T)$  is injective.*

*Proof.* We want to show that if we know some pair  $(C, \psi') \in \text{enc}(T)$ , we can uniquely reconstruct the triple  $(I, F, \psi) \in \text{cf}(T)$ .

Clearly,  $\psi = \psi'$ . The crux of our claim is in determining  $I$  and  $F$  from  $C$ , using knowledge of the pairing system  $\psi$ . From the function  $\eta_T$  itself, we know that  $I \oplus F = C \oplus (W \cup W')$ . From the pairings in  $\psi$  at each vertex, we know the breakdown of  $I \oplus F$  into components. From the transition  $T$ , we know which component is currently being processed. Now, assume for a moment that we know the complete ordering of all components and the ordering of edges within the current component. Call the current component  $A_i$ , those before it  $A_1, \dots, A_{i-1}$ , and those after it  $A_{i+1}, \dots, A_k$ . The edge cover  $I$  agrees with  $C$  on components  $A_1, \dots, A_{i-1}$  and with  $W$  on components  $A_{i+1}, \dots, A_k$ . On the current component  $A_i$ ,  $I$  agrees with  $C$  on the already-processed edges and with  $W$  on all other edges. For edges outside of  $I \oplus F$ , we know that  $I$  agrees only with  $I \cap F = W \setminus (I \oplus F)$ . The edge cover  $F$  is then simply  $I \oplus (I \oplus F)$ .

So, in order to make use of the above reasoning, we must show that we can find the complete ordering of all components as well as the ordering of edges within the current component. As described in our flow definition, the ordering of most components and the ordering of edges within those components is trivially determined via some fixed, arbitrary ordering of the edges in the graph. The components that break this easily-determined scheme are the pile-trails, all of which must be processed before any other component, and whose edges must also be processed in a special manner. To proceed, we must identify the pile-trails and the order in which their edges are processed.

This is done by first identifying the vertices with piled coverages. From Figure 2.4, we already know that some piled coverages are discernible simply by knowing their signatures derived from  $C$ . Some pairs of piled coverages are ambiguous, however. These include B1-C2, F1-G2, B-C1, and F-G1. We can discern even these, though, by simply taking note of how many edge-pairs are in the pairing at the vertex in question.<sup>6</sup> For instance, if we see a vertex with a  $0 \cdot 4$  signature with one edge-pair in its pairing, we know that it has to have a coverage of G1, not F, since a coverage of type F would necessitate two edge-pairs in the

---

<sup>6</sup>The knowledge of how many edge-pairs are in each pairing is given to us by the pairing system  $\psi = \psi'$ .

pairing. The other ambiguous pairs resolve similarly.

Now that we know which vertices are piled, we also know which trails are pile-trails — namely, any trail with a one- or two-piled coverage at either of its end-vertices. Thus, since component processing begins with any pile-trails (each in order of the lowest-ordered edge it contains) and ends with all other components (again, each in order of the lowest-ordered end-edge it contains), we know the complete ordering of all components.

Finally, we must be able to determine the ordering of edges within a pile-trail if it is the current component. To this end, we will examine the transition-type of  $T$ ; that is, whether it is a  $(\leftrightarrow)$ -transition, an  $(\uparrow)$ -transition, or a  $(\downarrow)$ -transition.

- $(\leftrightarrow)$  : In this case, one edge is removed while another is added. Processing began in the direction of the removed edge, away from the added edge; processing will end in the direction of the added edge, away from the removed edge.
- $(\uparrow)$  : One edge is added. If there are edges in this pile-trail that are adjacent to this added edge, then one is present in  $W$  and the other is not. (This is because any component originally consists of alternating edges from  $I$  and  $F$ .) Note that there may be both, one, or zero of these adjacent edges. If there are zero such edges, then this component consists of only one edge, and the edge-ordering in this component is trivial. Otherwise, processing began in the direction of the edge not in  $W$  and will end in the direction of the edge in  $W$ .
- $(\downarrow)$  : One edge is removed. The situation here is identical to that of an  $(\uparrow)$ -transition, except that processing began in the direction of the edge in  $W$  and will end in the direction of the edge not in  $W$ .

Our arguments above show that it is possible to determine both the complete ordering of components and the ordering of edges within each component. Consequently, we can uniquely reconstruct  $(I, F, \psi)$  from  $(C, \psi')$ , and the encoding function  $\eta_T$  is injective.  $\square$

### 2.2.2 Bounding the congestion and mixing time

We now have all the machinery necessary to derive bounds for the congestion and mixing time of  $\mathcal{X}$ .

**Proposition 2.4.** *Let  $G = (V, E)$  be any finite simple graph with no isolated vertices, where  $n = |V|$ ,  $m = |E|$ , and  $\Delta = \max_{v \in V} \deg(v)$ . Let  $\mathcal{P}$  and  $\mathcal{N}$  respectively denote the set of  $(1, 2)$ -edge covers and the set of near- $(1, 2)$ -edge covers in  $G$ .*

*For all starting states  $\alpha \in \Omega = \mathcal{P} \cup \mathcal{N}$  and for error tolerance  $\epsilon$ , the mixing time  $\tau_\alpha(\epsilon)$  of the Markov chain  $\mathcal{X}$  on  $(1, 2)$ -edge covers and near- $(1, 2)$ -edge covers in  $G$  is bounded as*

$$\tau_\alpha(\epsilon) \leq 206\Delta n^4 m^2 (n^4 + 1)(n - 1)^3 (n - 2)(m \log 2 + \log \epsilon^{-1})$$

*Proof.* We bound the mixing time by plugging into Inequality 2.1 a derived bound on congestion and simplifying  $\log \pi(\alpha)^{-1} = \log |\Omega| \leq \log 2^m = m \log 2$ . The congestion  $\rho$  is bounded as follows:

$$\begin{aligned} \rho &\leq \frac{2\Delta m^2}{|\Omega|} \max_T \sum_{(I, F, \psi) \in \text{cp}(T)} \frac{1}{|\Psi_{I, F}|} && \text{by Equation 2.4,} \\ &\leq \frac{2\Delta m^2}{|\Omega|} \max_T \sum_{(C, \psi') \in \text{enc}(T)} \frac{1}{|\Psi_{I, F}|} && \text{by Claim 2.3,} \\ &\leq \frac{2\Delta m^2}{|\Omega|} \max_T \sum_{(C, \psi') \in \text{enc}(T)} \frac{103n^4(n - 1)^3(n - 2)}{|\Psi'_{C, T}|} && \text{by Inequality 2.6,} \\ &\leq \frac{2\Delta m^2}{|\Omega|} \max_T \sum_{C \in \Omega'} \sum_{\psi' \in \Psi'_{C, T}} \frac{103n^4(n - 1)^3(n - 2)}{|\Psi'_{C, T}|} && \text{by Claims 2.1 and 2.2,} \\ &= \frac{2\Delta m^2}{|\Omega|} \sum_{C \in \Omega'} 103n^4(n - 1)^3(n - 2) \\ &\leq \frac{2\Delta m^2}{|\Omega|} 103n^4(n - 1)^3(n - 2)(n^4 + 1)|\Omega| && \text{by Inequality 2.5.} \\ &= 206\Delta n^4 m^2 (n^4 + 1)(n - 1)^3 (n - 2) \end{aligned}$$

By simple substitution, the bound is obtained. □

Finally, this allows us to state the following main result.

**Theorem 2.5.** *Let  $G = (V, E)$  be any finite simple graph with no isolated vertices, where  $n = |V|$ ,  $m = |E|$ , and  $\Delta = \max_{v \in V} \deg(v)$ . Let  $\mathcal{P}$  and  $\mathcal{N}$  respectively denote the set of  $(1, 2)$ -edge covers and the set of near- $(1, 2)$ -edge covers in  $G$ .*

*For constant  $k$  such that  $\frac{|\mathcal{N}|}{|\mathcal{P}|} \in O(n^k)$ , it is possible to efficiently sample  $(1, 2)$ -edge covers from  $G$  in  $O(\Delta m^2 n^{12+k} [m + \log \epsilon^{-1}])$  steps of the Markov chain  $\mathcal{X}$ , from a distribution within  $\epsilon$  total variation distance of the uniform distribution.*

*If expressed solely in terms of  $n$  and for fixed sampling error  $\epsilon$ , then  $O(n^{19+k})$  steps are required.*

*Proof.* This follows directly from the bound on mixing time in Proposition 2.4 and from the fact that, due to our crucial assumption at the beginning of this chapter, we must expect to take about  $\frac{|\mathcal{N}|}{|\mathcal{P}|} \in O(n^k)$  samples from  $\Omega = \mathcal{P} \cup \mathcal{N}$  in order to obtain one  $(1, 2)$ -edge cover sample from  $\mathcal{P}$ . □

Having developed and stated our main result, we now turn to a short discussion of an actual class of graphs for which  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  is bounded by a polynomial in  $n$ .

## 2.3 Graphs for which $\frac{|\mathcal{N}|}{|\mathcal{P}|} \in O(n^k)$

A *sparse graph* is one in which the number of edges  $m$  is linear in the number of vertices  $n$ , i.e.  $m \in O(n)$ . Notice that the type of graph illustrated in Figure 2.2 is sparse since, no matter how large we make a graph like this, an easy upper bound on the number of edges is  $m \leq 2n$ . As stated in the Figure's caption, graphs of this type have an exponential  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  ratio. Thus, in pursuit of graphs for which  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  is polynomially-bounded, we may want to explore graphs that are not sparse, i.e. *dense graphs* or graphs in which  $m \in \Omega(n^d)$  for some  $1 < d \leq 2$ .

In particular, we have the following result.

**Theorem 2.6.** *For any graph  $G = (V, E)$  of order  $n = |V|$  with minimum degree  $\delta > \frac{n-1}{3}$ , the ratio  $\frac{|\mathcal{N}|}{|\mathcal{P}|}$  of near-(1, 2)-edge covers to (1, 2)-edge covers is upper-bounded by  $n^8$ .*

*Proof.* We prove this bound by showing an injection from  $\mathcal{N}$  to  $\mathcal{P}$ . Essentially, we will take any  $A \in \mathcal{N}$ , record some information, transform  $A$  into some  $B \in \mathcal{P}$ , and then show that we can uniquely reconstruct  $A$  given  $B$  and the recorded information.

So, suppose we have some near-(1, 2)-edge cover  $A \in \mathcal{N}$ . We want to get rid of whatever piles  $A$  has and, in so doing, transform  $A$  into some (1, 2)-edge cover  $B \in \mathcal{P}$ . Take a vertex  $u_0$  with a pile in  $A$ , and call the pile-causing edge  $u_0u$ ; we remove this edge from  $A$ . Now, we know that  $u_0$  must still be properly covered by some edge in  $A$ , but we don't know if the vertex  $u$  is still covered in  $A$ , since it may have been covered only by  $u_0u$ . If  $u$  is still covered, then we have eliminated a pile and maintained proper edge-coverage, and so we would be done; however, if  $u$  is now uncovered, then we must re-cover it in some manner.

In the latter case, if  $u$  is adjacent to any 1-covered vertex  $u'$ , then the edge  $uu'$  can be added to  $A$  as a re-covering edge for  $u$ . (Note that  $uu'$  will not add another pile.) If not, then all neighbors of  $u$  are 2-covered. In this case, if there exists a neighbor  $u'$  that is covered by an edge  $u'u''$  such that  $u''$  is 2-covered, then the edge  $uu'$  can be added to  $A$  while the edge  $u'u''$  can be taken away from  $A$ . This would maintain proper edge-coverage of  $u'$  and  $u''$  while also re-covering  $u$ . However, if there is no such  $u'$  and  $u''$ , then we reach an impasse, the basic structure of which is illustrated in Figure 2.5.

We resolve this problem by giving a pre-condition *guaranteeing* that such vertices  $u', u''$  exist if the process of re-covering  $u$  reaches this point. This pre-condition is that the minimum degree  $\delta$  of the graph in which  $A$  exists is strictly greater than  $\frac{n-1}{3}$ . If this holds, then it is impossible for the situation in Figure 2.5 to be realized, since  $\delta > \frac{n-1}{3}$  implies that there must be (a) one more edge between  $u$  and some 1-covered neighbor of one of its neighbors or (b) there must be one less 1-covered neighbor of a neighbor.

So, we define an injective function  $z : \mathcal{N} \rightarrow \mathcal{P} \times V^8$  as  $z(A) = (B, u_0, u, u', u'', v_0, v, v', v'')$ , where the  $u$ 's and  $v$ 's correspond to dealing separately with the two possible piles in  $A$ . If

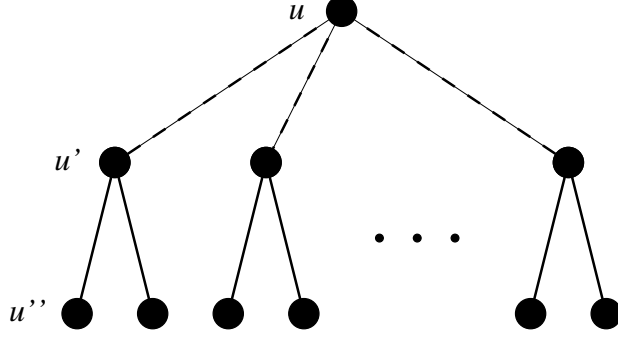


Figure 2.5: Troublesome scenario for our injection  $z : \mathcal{N} \rightarrow \mathcal{P} \times V^8$ . The vertex  $u$  has been uncovered by removal of the edge  $u_0u$  from the edge cover, and each neighbor  $u'$  of  $u$  is 2-covered with each co-covered neighbor  $u''$  of  $u'$  being 1-covered. Dashed lines indicate edges that exist in the graph but which are not currently in the edge cover, while solid lines indicate edges in the edge cover.

some vertices aren't needed (i.e.,  $u'$  and  $u''$  are not needed when removal of  $u_0u$  does not uncover  $u$ ), then we “null” them by assigning them a previously-used vertex. For example, if there was only one pile, and we handle it with the  $u$ 's, then we pick any vertex for  $v_0$  and then assign the same vertex to  $v, v', v''$ .

The near-(1,2)-edge cover  $A$  is uniquely reconstructed as  $B \cup \{u_0u, u'u'', v_0v, v'v''\} \setminus \{uu', vv'\}$ , ignoring any invalid edges (e.g.,  $u'u''$  is invalid when  $u' = u''$ ). Since  $z$  is injective, we know that  $|\mathcal{N}| \leq |\mathcal{P}| \cdot |V|^8$ . This completes the proof.  $\square$

We now know that  $\frac{|\mathcal{N}|}{|\mathcal{P}|} \in O(n^8)$  for graphs with minimum degree  $\delta > \frac{n-1}{3}$ , and so we know, by Theorem 2.5, that (1,2)-edge covers can be efficiently sampled from this kind of dense graph using  $O(n^{27})$  steps of our Markov chain.

## 2.4 Initializing the Markov chain

There is one aspect of our approach that has been overlooked so far, and that is how to initialize the Markov chain to a (1,2)-edge cover or near-(1,2)-edge cover starting state. This initialization is essential for an MCMC approach to be successfully implemented for sampling (1,2)-edge covers. An efficient algorithm that accepts  $G$  as input and guarantees



a  $(1, 2)$ -edge cover from  $G$  as output (if any  $(1, 2)$ -edge cover exists in  $G$ ) would be ideal. Fortunately, by virtue of the construction of the injection  $z$  exhibited in the proof of Theorem 2.6, we can indeed already sketch such an algorithm for the dense graphs discussed above (those with minimum degree  $\delta > \frac{n-1}{3}$ ):

1. Accept a graph  $G = (V, E)$  as input, where minimum degree  $\delta_G > \frac{n-1}{3}$ .
2. Generate a maximum matching  $M$  in  $G$ .<sup>7</sup>
3. Extend  $M$  to a minimum edge cover  $W$  by greedy addition of edges that cover any vertices left uncovered by  $M$ .
4. If  $W$  is not a  $(1, 2)$ -edge cover, then it must contain at least one pile. For each pile, apply the reasoning and construction from the proof of Theorem 2.6 to remove the pile from  $W$  while maintaining  $W$  as an edge cover; do this until no piles are left, i.e. until  $W$  is a  $(1, 2)$ -edge cover. (Though we do not present a time-complexity analysis in this brief sketch, the running-time of this step is indeed bounded by a polynomial in the size of  $G$ .)
5.  $W$  is a  $(1, 2)$ -edge cover in  $G$ . Return  $W$ .

Beyond this, no efficient algorithm for generating  $(1, 2)$ -edge covers was discovered during the course of this thesis work, though attempts were made that only resulted in heuristic-based approaches with no apparent guarantees on correctness.

Alternatively, instead of asking for an algorithm to generate any  $(1, 2)$ -edge cover, we may ask for an efficient algorithm to generate a perfect matching, which is actually also a  $(1, 2)$ -edge cover.<sup>8</sup> An efficient algorithm to generate a maximum matching would suffice for this purpose.

Thus, we must add at least one of two restrictions to the input graph  $G$  in order for our approach to allow the MCMC method to be successfully applied for efficiently sampling

---

<sup>7</sup>Efficient algorithms for generating maximum matchings are well-known and well-studied.

<sup>8</sup>In an extension of our notation, a perfect matching would be a 1-edge cover.

$(1, 2)$ -edge covers: the minimum degree of  $G$  must be strictly greater than  $\frac{n-1}{3}$ , or there must be at least one perfect matching in  $G$ .

## 2.5 Discussion

We have shown that the Markov chain Monte Carlo method can be used to efficiently sample  $(1, 2)$ -edge covers in any finite simple graph  $G$  satisfying two properties: the ratio of near- $(1, 2)$ -edge covers to  $(1, 2)$ -edge covers in  $G$  is bounded by a polynomial in the size of  $G$ , and some  $(1, 2)$ -edge cover in  $G$  can be generated in similarly polynomially-bounded time. Suplementarily, we showed that some dense graphs (those with minimum degree strictly greater than  $\frac{n-1}{3}$ ) satisfy these two properties; it would be an interesting and natural extension of this work to discover other graphs that satisfy them.

In showing the second property for the dense graphs considered above, we presented a brief sketch of an algorithm for generating a  $(1, 2)$ -edge cover. Another natural (and perhaps more well-defined) extension of this thesis work would be to fully explicate and analyze this algorithm, then use it to implement a complete MCMC sampler for  $(1, 2)$ -edge covers in graphs with minimum degree  $\delta > \frac{n-1}{3}$ . This sampler could then be used in experiments to determine an empirically-supported bound on its running-time. If this bound was found to be considerably lower than the very large theoretical bound of  $O(n^{27})$  implied by our main result, this could provide some evidence that the actual mixing time of our Markov chain is correspondingly lower. For practical applications of sampling  $(1, 2)$ -edge covers, this would be an important result. In addition, there might then be some impetus to theoretically demonstrate the lower mixing time. On the other hand, if the bound suggested by the experimental results was found to be very high or close to the theoretical bound given here, then that may also suggest that the latter is close to being a tight bound.

# Appendix A

## Edge covers in context

It is always helpful to know the practical and theoretical surroundings of a combinatorial object when it is under discussion, and the edge cover is certainly no exception.

For further reference, let  $G = (V, E)$  be an arbitrary finite simple graph, and let  $\delta_X$  denote the minimum degree of graph  $X$ .

### A.1 An application

One application of edge covers is in change detection algorithms for pairs of XML documents. The purpose of XML change detection is to efficiently generate a minimal script of standard edit operations that, when applied to the first XML document, will produce the second XML document. Some algorithms compute a solution by reducing the change detection problem to that of finding a minimum-cost edge cover in a weighted bipartite graph. Al-Akram, Adma, and Baysal [1] provide an original change detection algorithm and a discussion of results in the area.

## A.2 Related structures

There are three types of graph structure whose definitions are closely related in form to that of edge cover: matching, independent set, and vertex cover.

### A.2.1 Matchings and edge covers

A *matching* in  $G$  is a subset of  $E$  containing no adjacent edges. A *maximal matching* in  $G$  is a matching that is not a proper subset of any other matching. A *maximum matching* in  $G$  is a matching with cardinality greater than or equal to the cardinality of any other matching. This maximum cardinality is called the *edge independence number* of  $G$ .

A *perfect matching* in  $G$  is a subset of  $E$  that is both a matching (necessarily maximum) and an edge cover (necessarily minimum). Note that  $G$  has a perfect matching only if  $|V|$  is even.

For arbitrary  $G$ , an explicit relationship between edge covers and matchings is not clear.<sup>1</sup> To see why this is so, consider the following two observations.

#### Non-complementarity

At first glance, we might expect to obtain an edge cover  $E \setminus M$  from a matching  $M$ , or we might expect to obtain a matching  $E \setminus W$  from an edge cover  $W$ . However, these expectations are generally unfounded.

To see why the first construction fails, consider an edge  $(u, v) \in M$ , and suppose that  $\deg(u) = 1$ . Then  $E \setminus M$  cannot cover  $u$ , and  $E \setminus M$  is not an edge cover. Note that if  $\delta_G \geq 2$ , then the above situation cannot hold, and  $E \setminus M$  is always an edge cover.

The second construction fails under slightly more intricate circumstances. Consider a vertex  $v \in V$  with three distinct neighbors  $x, y, z$ . Furthermore, let  $\deg(x) \geq 2$  and  $\deg(y) \geq$

---

<sup>1</sup>When  $G$  has perfect matchings, the set of perfect matchings is the set of maximum matchings which is, in turn, the set of minimum edge covers. Furthermore, when  $G$  has no perfect matchings, the set of maximum matchings is still closely related to the set of minimum edge covers. This is because each maximum matching is easily extensible to at least one minimum edge cover by greedily adding edges that cover uncovered vertices.

2, and let  $x', y'$  be respective neighbors of  $x, y$ , distinct from  $v$  (but not necessarily distinct from each other or from  $x, y, z$ ). Now, suppose  $(x, x'), (y, y'), (v, z) \in W$  and  $(v, x), (v, y) \notin W$ . Then  $(v, x), (v, y) \in E \setminus W$ , and  $E \setminus W$  is not a matching. Again, note that if there is no such  $v$ , then the above situation cannot hold, and  $E \setminus W$  is always a matching.

## Asymptotic imbalance

The growth of the number of edge covers in a graph can vary substantially from the growth of the number of matchings. For example, the star graph  $K_{1,n}$  has  $n + 1$  matchings but always exactly one edge cover. On the other hand, the complete graph  $K_n$  has  $2^{O(n \log n)}$  matchings and  $2^{O(n^2)}$  edge covers, the latter growing asymptotically faster than the former.

Specifically, the number of matchings in  $K_n$  is given by the recursive function

$$M(n) = M_{\text{rec}}(n, 0)$$

where

$$M_{\text{rec}}(n, \ell) = \begin{cases} 1 + \frac{\binom{n-2\ell}{2}}{\ell+1} M_{\text{rec}}(n-2, \ell+1) & \text{for } n \geq 2, \\ 1 & \text{for } n = 0 \text{ or } n = 1, \end{cases}$$

and the number of edge covers in  $K_n$  is given by the recursive function

$$W(n) = \begin{cases} (2^{\binom{n}{2}} - 1) - \sum_{i=1}^{n-2} \binom{n}{i} W(n-i) & \text{for } n > 2, \\ 1 & \text{for } n = 2. \end{cases}$$

Although both  $M(n)$  and  $W(n)$  grow exponentially, the number of matchings is rapidly outpaced by the number of edge covers. For example,  $K_8$  has 764 matchings and 252,522,481 edge covers, while  $K_9$  has 2,620 matchings and 66,376,424,160 edge covers.<sup>2</sup>

---

<sup>2</sup>I implemented a backtracking search for each function and verified its values for  $2 \leq n \leq 9$ .

### A.2.2 Independent sets and vertex covers

An *independent set* in  $G$  is a subset of  $V$  containing no adjacent vertices. A *maximal independent set* in  $G$  is an independent set that is not a proper subset of any other independent set. A *maximum independent set* in  $G$  is an independent set with cardinality greater than or equal to the cardinality of any other independent set. This maximum cardinality is called the *vertex independence number* of  $G$ .

A *vertex cover* in  $G$  is a subset of  $V$  containing vertices sufficient to cover all edges in  $E$ . A *minimal vertex cover* in  $G$  is a vertex cover that is not a proper superset of any other vertex cover. A *minimum vertex cover* in  $G$  is a vertex cover with cardinality less than or equal to the cardinality of any other vertex cover. This minimum cardinality is called the *vertex covering number* of  $G$ .

Independent sets and vertex covers are *dual*, in the sense that maximizing the former is exactly equivalent to minimizing the latter.

**Fact A.1.** *For any independent set  $I$  in  $G$ ,  $V \setminus I$  is a vertex cover; likewise, for any vertex cover  $J$  in  $G$ ,  $V \setminus J$  is an independent set.*

The following statement helps to imply Proposition A.8.

**Proposition A.2.** *A bijection exists from the space of vertex covers in  $G$  to the space of independent sets in  $G$ .*

*Proof.* This follows directly from Fact A.1. □

## A.3 Related properties

### A.3.1 Gallai identities

When  $G$  has no isolated vertices (i.e., when  $G$  has at least one edge cover), the *Gallai identity for edges* ([3], Theorem 8.7) states that  $|V|$  is equal to the sum of the edge independence

number of  $G$  and the edge covering number of  $G$ . The *Gallai identity for vertices* ([3], Theorem 8.8) states that  $|V|$  is equal to the sum of the vertex independence number of  $G$  and the vertex covering number of  $G$ .

### A.3.2 Relations via line graphs

Using the concept of the line graph, we can demonstrate relations among edge covers, vertex covers, independent sets, and matchings. The *line graph* of  $G$  is denoted  $L(G)$  and is defined by  $V_{L(G)} = \{uv : (u, v) \in E_G\}$  and  $E_{L(G)} = \{(uv, vw) : (u, v), (v, w) \in E_G\}$ . ( $G$  is called the *root graph* of  $L(G)$ .)

#### Independent sets in $L(G)$ and matchings in $G$

**Fact A.3.** *For any matching  $M$  in  $G$ , the vertex-set  $\{uv \in V_{L(G)} : (u, v) \in M\}$  is an independent set in  $L(G)$ ; likewise, for any independent set  $I$  in  $L(G)$ , the edge-set  $\{(u, v) \in E_G : uv \in I\}$  is a matching in  $G$ .*

**Proposition A.4.** *A bijection exists from the space of independent sets in  $L(G)$  to the space of matchings in  $G$ .*

*Proof.* This follows directly from Fact A.3. □

#### Edge covers in $L(G)$ and vertex covers in $G$

**Claim A.5.** *For any edge cover  $W$  in  $L(G)$ , the vertex-set  $J = \{v \in V_G : (uv, vw) \in W\}$  is a vertex cover in  $G$ .*

*Proof.* For the sake of contradiction, suppose that  $J$  is not a vertex cover in  $G$ . Then some edge  $(v, z) \in E_G$  is not covered by  $J$ , thus neither  $v$  nor  $z$  are in  $J$ . Take  $v$ ; since  $v \notin J$ , we know that there is no edge  $(uv, vw)$  in  $W$  for any  $u$  or  $w$ . Let  $w = z$ ; by the preceding statement, we know that there is no edge in  $W$  with  $vz$  as an endpoint. Thus  $W$  does not cover  $vz \in E_G$ , and  $W$  is not an edge cover in  $L(G)$ . But  $W$  is given as an edge cover in

$L(G)$ ; therefore, our supposition that  $J$  is not a vertex cover in  $G$  is wrong. So  $J$  is a vertex cover in  $G$ .  $\square$

**Proposition A.6.** *A many-to-one total function  $\alpha$  exists from the space of edge covers in  $L(G)$  to the space of vertex covers in  $G$ .*

*Proof.* This follows directly from Claim A.5.  $\square$

**Claim A.7.** *If  $\delta_G \geq 2$ , then  $\alpha$  is surjective.*

*Proof.* To demonstrate this, we will first construct an edge-set  $W \subseteq E_{L(G)}$  that, by the construction given in Claim A.5, exactly produces any vertex cover  $J \subseteq V_G$ . Then we will show that  $W$  is an edge cover in  $L(G)$ .

Take any vertex cover  $J$  in  $G$ . Since  $\delta_G \geq 2$ , there exist at least two distinct edges  $(u, v), (v, w) \in E_G$  for any vertex  $v \in V_G$ , thus also for each vertex in  $J \subseteq V_G$ . Now let  $W = \{(uv, vw) \in E_{L(G)} : v \in J \wedge u, w \in V_G\}$ . Even though  $W$  is probably larger than necessary, it is clear that  $J = \{v \in V_G : (uv, vw) \in W\}$ .

For the sake of contradiction, suppose that  $W$  is not an edge cover in  $L(G)$ . Then there exists some uncovered  $ab \in V_{L(G)}$ . Since  $ab \in V_{L(G)}$  if and only if  $(a, b) \in E_G$ , and since  $J$  has been taken to be a vertex cover in  $G$ , we know that  $(a, b)$  must be covered by one or both of  $a$  and  $b$  in  $J$ . Take  $b \in J$ ; then, by the way we defined  $W$  in the previous paragraph, we know that  $(ab, bv) \in W$  for each vertex in  $\{v \in V_G : (v, b) \in E_G \wedge v \neq a\}$ . (The precondition  $\delta_G \geq 2$  implies that this set is not empty.) Any one of these edges covers  $ab$ ; thus,  $W$  covers  $ab$ . But  $ab$  was taken to be uncovered by  $W$ ; therefore, our supposition that  $W$  is not an edge cover in  $L(G)$  is wrong. So  $W$  is an edge cover in  $L(G)$ .

We have shown that for any vertex cover  $J$  in  $G$ , there exists the constructed edge cover  $W$  in  $L(G)$  such that  $J = \{v \in V_G : (uv, vw) \in W\}$ . Therefore,  $\alpha$  is surjective.  $\square$



## Edge covers in $L(L(G))$ and matchings in $G$

**Proposition A.8.** *If  $\delta_{L(G)} \geq 2$ , then a many-to-one total surjection exists from the space of edge covers in  $L(L(G))$  to the space of matchings in  $G$ .*

*Proof.* This follows directly from Claim A.7 and Propositions A.6, A.2, and A.4. □

# Bibliography

- [1] R. Al-Ekram, A. Adma, and O. Baysal. diffx: An algorithm to detect changes in multi-version XML documents. In *CASCON*, 2005.
- [2] I. Bezáková, N. Bhatnagar, and E. Vigoda. Sampling binary contingency tables with a greedy start. *Random Structures and Algorithms*, 2007.
- [3] G. Chartrand and P. Zhang. *Introduction to Graph Theory*. McGraw-Hill, 2005.
- [4] C. Cooper, M. Dyer, and C. Greenhill. Sampling regular graphs and a peer-to-peer network. *Combinatorics, Probability and Computing*, 2007.
- [5] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of markov chains. *The Annals of Applied Probability*, 1991.
- [6] M. Dyer, A. Frieze, and M. Jerrum. On counting independent sets in sparse graphs. *SIAM Journal on Computing*, 2002.
- [7] M. Dyer and C. Greenhill. On markov chains for independent sets. *Journal of Algorithms*, 2000.
- [8] W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1998.
- [9] J. L. Gross and J. Yellen, editors. *Handbook of Graph Theory*. CRC Press, 2004.

- [10] O. Häggström. *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, 2002.
- [11] M. Huber. Exact sampling from perfect matchings of dense regular bipartite graphs. *Algorithmica*, 2006.
- [12] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithm for the Ising model. *SIAM Journal of Computing*, 1993.
- [13] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 1986.
- [14] M. R. Jerrum. *Counting, Sampling and Integrating: Algorithms and Complexity*. Birkhäuser Verlag, 2003.
- [15] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley, 2005.
- [16] J. Martin. *Introduction to Languages and the Theory of Computation*. McGraw-Hill, 2003.
- [17] A. Sinclair. *Algorithms for Random Generation & Counting: A Markov Chain Approach*. Birkhäuser, 1993.
- [18] E. Vigoda. A note on the Glauber dynamics for sampling independent sets. *Electronic Journal of Combinatorics*, 2001.