



HAL
open science

Dynamic network formation

Rémi Varloot

► **To cite this version:**

Rémi Varloot. Dynamic network formation. Data Structures and Algorithms [cs.DS]. Université Paris sciences et lettres, 2018. English. NNT : 2018PSLEE048 . tel-02385281

HAL Id: tel-02385281

<https://theses.hal.science/tel-02385281>

Submitted on 28 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à l'École Normale Supérieure

Dynamic Network Formation DYNAMIQUE DE FORMATION DES RÉSEAUX

École doctorale n°386

SCIENCE MATHÉMATIQUES DE PARIS CENTRE

Spécialité

INFORMATIQUE, MATHÉMATIQUES

Soutenue par

Rémi VARLOOT

le 1^{er} juin 2018

Dirigée par **Laurent MASSOULIÉ**
et **Ana BUŠIĆ**

Composition du jury

Ana BUŠIĆ
Inria Paris-ENS
Co-directrice de thèse

George GIAKKOUPIS
Inria Rennes
Rapporteur

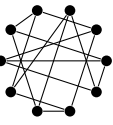
Adrian KOSOWSKI
Université Paris Diderot
Rapporteur

Laurent MASSOULIÉ
Inria Paris-ENS
Directeur de thèse

Giovanni NEGLIA
Inria Sophia Antipolis - Méditerranée
Examineur, Président du jury

Milan VOJNOVIC
London School of Economics
Examineur





Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse : Laurent Massoulié, pour avoir été particulièrement disponible et m'avoir maintes fois conseillé de nouvelles pistes à explorer, et Ana Bušić, pour nos nombreux et enrichissants échanges depuis mes premiers pas chez TREC il y a maintenant six ans.

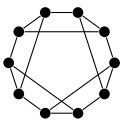
Ensuite, des remerciements qui se doivent d'être rédigés en anglais. I would like to thank my reviewers, George Giakkoupis and Adrian Kosowski, for taking the time to work their way through this thesis, and for their personal contributions: George and I had a most constructive discussion at the IHP back in 2016, and Adrian provided numerous constructive ideas on how to improve the thesis as a whole. This gratitude extends to Giovanni Neglia and Milan Vojnovic, who immediately accepted to be part of the jury.

Je ne serais pas là sans la contribution de certains professeurs, qui m'ont accompagné depuis de longues années. Je pense notamment à François Baccelli, grâce à qui je me suis intéressé de plus près aux réseaux ; à Marc Lelarge, qui m'a conseillé pour cette thèse ; et à Anne Bouillard, pour son co-encadrement en Master et son suivi régulier. Une mention toute particulière à Ana, Anne et Marc, pour qui j'ai eu la chance d'assurer des TDs à l'ENS durant ces trois années.

La thèse au quotidien, c'est aussi une équipe de recherche. Ou deux. Un grand merci, donc, à tous les membres d'Infine et de Dyogene : Lennart, Christelle, Loïc, Pierre, Rafael, Léo, Raul, Alexandre, Hirah, Bartek, Francisco, Émilie, Guangshuo, Thomas, Cédric, Anastasios, Aline, Paul, Alexandre, Virag, Benoit, Sean, et tous les autres habitués des lieux, qu'ils soient adeptes de la caféine chez SpecFun, fidèles du DLP au SED, ou encore ornithoryncophiles au Lincs. Et comme il n'y pas que des chercheurs, je souhaiterais exprimer ma gratitude envers Isabelle Delais et Hélène Bessin-Rousseau, sans qui cette thèse aurait été un cauchemar administratif.

Les nouveaux collègues, aussi, pour qui VOId est un nom de classe acceptable : Nicolas, Pierre, Ludovic, Mathieu, Richard, Dinh Thai, François, Michel, Vincent, Natalya, Fabien, Marc-Olivier, Serge, Matthews, Nesrine et Imaran.

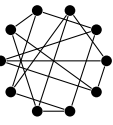
Une pensée aussi à tous ceux qui ont subi cette thèse en dehors de tout contexte professionnel : l'ATR de chez Connexion Swing, et plus généralement les chroniqueurs du *Rémi point* ; les aventuriers hauts et courts, hobbits comme chasseurs de géants, qui veulent clairement ma peau ; les adeptes du 41-30 et du squeeze cardiaque ; les fidèles du séminaire de l'Abbé Mole et du C6 ; les planteurs de clous ; les bétonneurs de lemmings ; vous tous. Tenter une liste exhaustive est un pari trop osé pour que je m'y risque, donc voici très officiellement une liste non-exhaustive : les Camomilles, Émilie, Delphine, Sand & Ludo, Audrey, Philippe, VéroNike, Pedro, Chaton, Sam, Chacha, Albin, Iron Man, Pierrot, Karine, SimJay, Marine, Matthieu, Anne, Éric, Silvain, Catherine, Max, Fathi, Marie, David,



Julia, Bastien, Yasmine, Jack, Quentin, Olivier, Baptiste, Adeline, Jean-Baptiste, Adeline, Denise, Ismaël, Igor, Céline, Vincent, Valérie, Vincent, Marie, Nicolas, Benjamin, Christine, Juliette, Iryna.

Un très grand merci à Dominique et Roland pour leurs accueils réguliers en terres volcaniques, à Floriane et Anouk avec qui nous avons parcouru l'Écosse, et à Aymeric et Jordane qui m'ont fait voir les lunes de Jupiter et les anneaux de Saturne. À ma famille aussi : les cousins et petits-cousins, les oncles et tantes, les grands-parents. Patrick, qui travaille tellement dans l'ombre qu'on ne sait pas trop ce qu'il fait ; Estelle, qui suit de si près mes pas que nos discussions font fuir la famille ; et Gabby, sans qui le nid familial eût été injoignable durant tout ce temps. À François, dont les jeux de plateaux indénombrables ne sont sans doute pas pour rien dans mon intérêt pour les graphes, et à Laurence, dont les parties de rail ont changé à tout jamais ma définition du plus court chemin et de la connexité.

Il ne me reste plus qu'à remercier chaleureusement Daphné de sa présence à mes côtés tout au long de ce voyage.



Abstract

This thesis focuses on the rapid mixing of graph-related Markov chains.

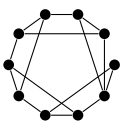
The main contribution concerns graphs with local edge dynamics, in which the topology of a graph evolves as edges slide along one another. We propose a classification of existing models of dynamic graphs, and illustrate how evolving along a changing structure improves the convergence rate. This is complemented by a proof of the rapid mixing time for one such dynamic. As part of this proof, we introduce the partial expansion of a graph. This notion allows us to track the progression of the dynamic, from a state with poor expansion to good expansion at equilibrium.

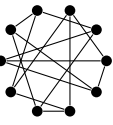
The end of the thesis proposes an improvement of the Propp and Wilson perfect sampling technique. We introduce oracle sampling, a method inspired by importance sampling that reduces the overall complexity of the Propp and Wilson algorithm. We provide a proof of correctness, and study the performance of this method when sampling independent sets from certain graphs.

Cette thèse porte sur la rapidité du temps de mélange de chaînes de Markov sur des graphes.

La contribution principale concerne les graphes avec des dynamiques locales sur les arêtes, la topologie du graphe évoluant au fur et à mesure que les arêtes glissent les unes le long des autres. Nous proposons une classification des différents modèles existants de graphes dynamiques, tout en illustrant l'importance des transitions le long d'une structure mouvante pour améliorer la vitesse de convergence. Cette étude est complétée par la preuve, pour l'une de ces dynamiques, d'un temps de mélange rapide. Nous définissons notamment l'expansion partielle d'un graphe. Celle-ci permet de suivre l'avancement de la dynamique, partant d'un état de faible expansion, jusqu'à obtention d'une bonne expansion à l'équilibre.

La fin de cette thèse porte sur une amélioration de l'algorithme de simulation parfaite de Propp et Wilson. Nous introduisant un oracle pour les transitions, inspiré de l'échantillonnage préférentiel, qui permet de réduire la complexité de l'algorithme. Nous fournissons une preuve de correction, ainsi qu'une étude de l'impact de cette méthode sur la vitesse d'échantillonnage d'ensembles indépendants pour certains graphes.





Contents

- 1 Introduction** **1**

This chapter review concepts and results on Markov chains that will be used throughout the following chapters. It focuses on methods for establishing upper bounds on the mixing time of Markov chains. The end of this chapter is dedicated to presenting the main results of this thesis.

- 2 Taxonomy of Dynamic Graphs** **21**

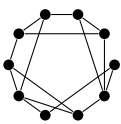
This survey presents a number of possible models for dynamic graphs, and establishes some classifications based on several design choices. By establishing patterns between known bounds on the mixing time and the features of each dynamic, it gives some insight as to the ingredients that make up a rapidly converging graph dynamic. The key observation is a need for bootstrapped transitions in order to obtain fast mixing with local dynamics.

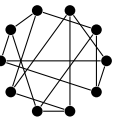
- 3 Rapid Mixing of Local Dynamic Graphs** **45**

Comprising the main result of this thesis, this section presents a proof of rapid convergence for a specific local graph dynamic. Starting from an arbitrary configuration, the topology of the graph has mixed after only a polylogarithmic number of updates per vertex. This proof introduces the notion of partial graph expansion, which plays an essential role in tracking the evolution of the graph throughout the transient phase.

- 4 Oracle Sampling** **65**

This last section presents an extension of the perfect sampling algorithm by Propp and Wilson. It introduces a method for ignoring unnecessary transitions throughout the simulation, while at the same time compensating for the bias that this approach introduces. The algorithm is given, along with a proof of correctness and case study of its performance when sampling random independent sets over certain graphs.





Chapter 1

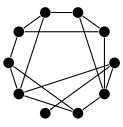
Introduction

1	Markov Chains	2
2	Stationary Distribution	5
3	Markov Chain Mixing Time	8
4	Spectral Methods	9
5	The Coupling and Path Methods	13
6	Contributions	15
6.1	Partial Isoperimetric Constants	16
6.2	Rapid Mixing of a Local Edge Dynamic	16
6.3	Oracle Sampling	17

Markov chains were introduced at the start of the 20th century by A. Markov [40], who used them to prove that the weak law of large numbers did not necessarily require independence—the established result is comparable to modern versions of the ergodic theorem—and to study the correlation between successive vowel and consonant occurrences in Russian literature. The scope of Markov chains has since then expanded to domains such as sampling and simulation, the study of dynamic systems, or combinatorial optimization. It is namely the theoretical basis for the acclaimed Markov chain Monte Carlo (MCMC) algorithm.

The results presented in Sections 1 to 5 of this chapter are for the most part standard results, that can be found in works such as *Markov Chains and Mixing Times* [36], *Reversible Markov Chains and Random Walks on Graphs* [2] and *Finite Markov Chains and Algorithmic Applications* [24]. They are adapted here to best reflect their use in the following chapters.

The main contributions of this thesis are presented in Section 6.



Throughout this chapter, we consider a finite set \mathcal{S} , which we refer to as the *state space*.

1 Markov Chains

A Markov chain is a memoryless stochastic process over a countable state space. The memoryless property, also known as Markov property, establishes that the distribution of the process at any given time only depends on its last known position, and is independent of its trajectory prior to that.

Formally, one defines discrete and continuous time Markov chains as follows.

Definition 1 (Discrete Time Markov Chain). *A (discrete time) Markov chain over \mathcal{S} is a random sequence $(X_t)_{t \in \mathbb{N}} \in \mathcal{S}^{\otimes \mathbb{N}}$ that satisfies the Markov property*

$$\mathbf{P}(X_{t+1} = x_{t+1} \mid X_t = x_t, \dots, X_0 = x_0) = \mathbf{P}(X_{t+1} = x_{t+1} \mid X_t = x_t).$$

If $\mathbf{P}(X_{t+1} = x' \mid X_t = x)$ does not depend on t , i.e.

$$\mathbf{P}(X_{t+1} = x' \mid X_t = x) = \mathbf{P}(X_1 = x' \mid X_0 = x),$$

the Markov chain is said to be time homogeneous. The transition matrix of a time homogeneous Markov chain is the matrix P such that

$$P(x, x') = \mathbf{P}(X_1 = x' \mid X_0 = x).$$

For $S, S' \subseteq \mathcal{S}$, it is often convenient to also denote

$$P(S, S') = \sum_{x \in S} \sum_{x' \in S'} P(x, x').$$

We have the following analogous definition for the continuous setting.

Definition 2 (Continuous Time Markov Chain). *A continuous time Markov chain over \mathcal{S} is a process $(X_t)_{t \in \mathbb{R}}$ such that, for any increasing non-negative sequence $(t_i)_{i \in \mathbb{N}}$ and $n \in \mathbb{N}$,*

$$\mathbf{P}(X_{t_{n+1}} = x_{t_{n+1}} \mid X_{t_n} = x_{t_n}, \dots, X_{t_0} = x_{t_0}) = \mathbf{P}(X_{t_{n+1}} = x_{t_{n+1}} \mid X_{t_n} = x_{t_n})$$

and, for all $t \geq 0$ and $x \neq x'$,

$$\frac{\mathbf{P}(X_{t+h} = x' \mid X_t = x)}{h}$$

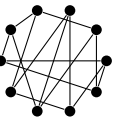
converges to a finite value as h goes to 0.

If $\mathbf{P}(X_t = x' \mid X_s = x) = \mathbf{P}(X_{t-s} = x' \mid X_0 = x)$ for all $0 < s < t$, the Markov chain is time homogeneous. We then define the infinitesimal generator of the Markov chain as the matrix Q such that

$$Q(x, x') = \lim_{t \rightarrow 0} \frac{\mathbf{P}(X_t = x' \mid X_0 = x)}{t}$$

for all $x \neq x'$, and

$$Q(x, x) = - \sum_{x' \neq x} Q(x, x').$$



1. MARKOV CHAINS

The distributions of time homogeneous Markov chains at time t , starting from state $x \in \mathcal{S}$, are respectively

$$P^t(x, \cdot) \quad \text{and} \quad H_t(x, \cdot) = e^{tQ}(x, \cdot)$$

in the discrete and continuous settings. H_t is called the heat kernel of Q .

All Markov chains studied in this thesis are time-homogeneous.

A classical example of a Markov chain is the birth and death process detailed below. It is also known as the M/M/1 queue in queueing theory [34].

Example 1 (The Birth and Death Process). *Let $N \in \mathbb{N}$, and consider a group of individuals whose total population evolves between 0 and N over time.*

For the discrete setting, we have that, at each time step, either a new individual is born (unless the population count is already N), or an individual dies. Births occur with probability $p \in (0, 1)$, and deaths with probability $q = 1 - p$.

Denote $(X_t)_{t \in \mathbb{N}}$ the number of individuals at instant t . $(X_t)_{t \in \mathbb{N}}$ is a time homogeneous Markov chain over $\mathcal{S} = \llbracket 0, N \rrbracket$, and its transition matrix is

$$P = \begin{pmatrix} q & p & 0 & \cdots & 0 \\ q & 0 & p & \ddots & \vdots \\ 0 & q & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & p \\ 0 & \cdots & 0 & q & p \end{pmatrix}.$$

In the continuous setting, consider instead that births occur at rate $\lambda > 0$, and deaths at rate $\mu > 0$. Consider an i.i.d. sequence of random variables $(t_i)_{i \in \mathbb{N}}$, exponentially distributed with parameter $\lambda + \mu$, and denote $s_i = \sum_{j=1}^i t_j$. Define $(X_t)_{t \in \mathbb{R}}$ as being constant except at instances $(s_i)_{i \in \mathbb{N}}$, and that at each such instance, either a birth occurs (with probability $\frac{\lambda}{\lambda + \mu}$), or a death occurs (with probability $\frac{\mu}{\lambda + \mu}$). The population is once more clamped to $\llbracket 0, N \rrbracket$.

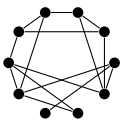
The infinitesimal generator of this Markov chain is

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & \cdots & 0 \\ \mu & -(\lambda + \mu) & \lambda & \ddots & \vdots \\ 0 & \mu & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -(\lambda + \mu) & \lambda \\ 0 & \cdots & 0 & \mu & -\mu \end{pmatrix}.$$

Notice that a discrete time Markov chain is the stochastic equivalent of a sequence defined by a recurrence relation. As such, it is often convenient to define them through a recurrence relation map:

Definition 3 (Random Mapping Representation). *Let P be a transition matrix over \mathcal{S} , and θ be a random variable in a set Θ . A random mapping representation of P is a function $f : \mathcal{S} \times \Theta \rightarrow \mathcal{S}$ such that, for all $x, x' \in \mathcal{S}$,*

$$\mathbf{P}(f(x, \theta) = x') = P(x, x').$$



CHAPTER 1. INTRODUCTION

Given a random mapping representation f of P , it is possible to define a Markov chain $(X_t)_{t \in \mathbb{N}}$ with transition matrix P recursively as

$$X_{t+1} = f(X_t, \theta_{t+1}),$$

where $(\theta_t)_{t \in \mathbb{N}}$ is a sequence of i.i.d. random variables with the same distribution as θ .

In the case of Example 1, a possible random mapping representation would be to have θ be equal to 1 with probability p , -1 otherwise, and f be defined as

$$f(x, \theta) = \max(\min(x + \theta, 0), N).$$

While random mapping representations are powerful in expressing the evolution of a Markov chain, they can quickly lead to cumbersome notations:

$$X_3 = f(f(f(X_0, \theta_1), \theta_2), \theta_3). \quad (1.1)$$

To avoid this pitfall, we make use of an alternate representation: Markov automata [44].

Definition 4 (Markov automaton). *Let P be a transition matrix over \mathcal{S} . Let \mathcal{A} be a finite set, and D be a distribution over \mathcal{A} . Finally, let $\cdot : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ be a transition function such that, for all $x, x' \in \mathcal{S}$,*

$$\mathbf{P}(x \cdot A = x') = P(x, x'),$$

where A is distributed according to D .

The quadruple $\mathbf{A} = (\mathcal{S}, \mathcal{A}, D, \cdot)$ is called a Markov automaton. \mathcal{A} is the alphabet of \mathbf{A} , and its elements are referred to as letters or events. A sequence of letters is called a word.

A Markov chain $(X_t)_{t \in \mathbb{N}}$ with transition matrix P can now be defined recursively as

$$X_{t+1} = X_t \cdot A_{t+1},$$

where $(A_t)_{t \in \mathbb{N}}$ is a sequence of i.i.d. random events distributed according to D .

With this formalism, Equation (1.1) becomes

$$X_3 = X_0 \cdot A_1 \cdot A_2 \cdot A_3.$$

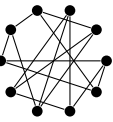
A natural Markov automaton for Example 1 would be

$$\mathbf{A} = \left(\begin{array}{l} \mathcal{S} = \llbracket 0, N \rrbracket \\ \mathcal{A} = \{b, d\} \\ D: b \mapsto p, d \mapsto q \end{array} \quad \cdot : (x, a) \mapsto \begin{cases} x + 1 & \text{if } a = b \text{ and } x < N \\ x - 1 & \text{if } a = d \text{ and } x > 0 \\ x & \text{otherwise} \end{cases} \right). \quad (1.2)$$

Finally, a common method of representing both discrete and continuous time Markov chains is by means of a graph.

Definition 5 (Transition graph). *For a transition matrix P over state space \mathcal{S} , consider the directed weighted graph $G = (V, E, w)$ where*

- $V = \mathcal{S}$,



2. STATIONARY DISTRIBUTION

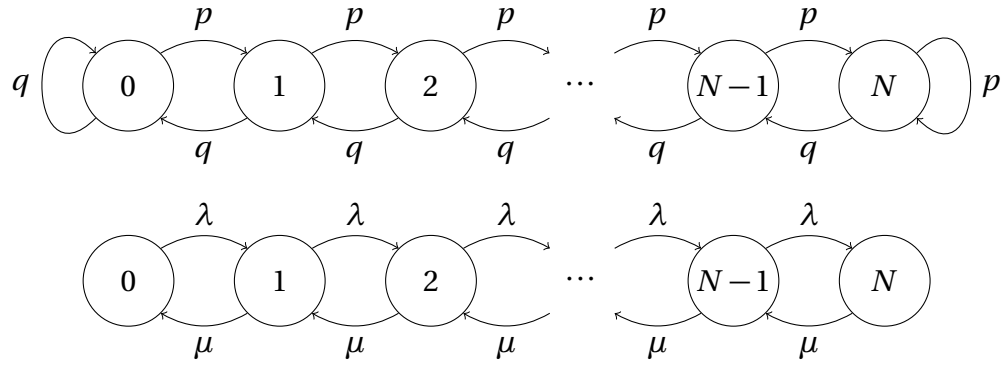


Figure 1.1 – Discrete and Continuous Time Birth-and-Death Processes

- $E = \{(x, x') \in \mathcal{S}^2 \mid P(x, x') > 0\}$, and
- $w : E \rightarrow \mathbb{R}_+$ such that $w(x, x') = P(x, x')$.

G is the graph representation of P .

For a continuous time Markov chain with infinitesimal generator Q , the graph representation is defined similarly, but contains no self-loops.

Conversely, given a graph, one can define the Markov chain represented by this graph, called the random walk on the graph.

Definition 6 (Random walk). Given a directed weighted graph $G = (V, E, w)$, define the out-degree of a vertex as the sum of the weights of outgoing edges:

$$\deg_+ x = \sum_{x':(x,x') \in E} w(x, x').$$

The discrete time random walk on G is the Markov chain $(X_t)_{t \in \mathbb{N}}$ such that, for every $t \in \mathbb{N}$, X_{t+1} is distributed according to

$$P(X_t, \cdot) = \frac{w(X_t, \cdot)}{\deg_+ X_t}.$$

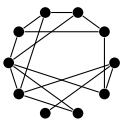
Similarly, the continuous time random walk on G is the Markov chain $(X_t)_{t \in \mathbb{R}}$ that spends an average time of $\deg_+ x = -Q(x, x)$ in a state x , then jumps to a new state distributed according to $Q(x, \cdot)$.

In this case of an unweighted graph, the Markov chain is referred to as the simple random walk.

The graph representations of the Markov chains from Example 1 are given in Figure 1.1.

2 Stationary Distribution

One of the key properties of Markov chains is the existence, in most cases, of a unique stationary distribution, and the convergence of the chain to that distribution. This is the basis for the Monte Carlo sampling method presented in Section 3.



CHAPTER 1. INTRODUCTION

Recall that, throughout this chapter, we restrain ourselves to a finite state space \mathcal{S} .

Definition 7 (Stationary distribution). *Let P be a transition matrix over \mathcal{S} . A distribution π , represented as a line vector $(\pi(x))_{x \in \mathcal{S}}$, is said to be a stationary distribution of P if $\pi = \pi P$.*

Similarly, a stationary distribution π of an infinitesimal generator Q is one that satisfies $0 = \pi Q$.

Going back once more to Example 1, denoting $r = \frac{p}{q}$ and $\rho = \frac{\lambda}{\mu}$, one can easily check that the distributions

$$\pi_p(k) = r^k \frac{1-r}{1-r^{N+1}} \quad \left(\text{with } \pi_{\frac{1}{2}}(k) = \frac{1}{N+1}\right)$$

and

$$\pi_{\lambda,\mu}(k) = \rho^k \frac{1-\rho}{1-\rho^{N+1}} \quad \left(\text{or } \pi_{\lambda,\mu}(k) = \frac{1}{N+1} \text{ if } \lambda = \mu\right)$$

are stationary distributions of P and Q , respectively.

To establish the existence and uniqueness of a stationary distribution, and the convergence of the Markov chain to that stationary distribution, the most common approach is to check for irreducibility and aperiodicity.

Definition 8 (Irreducible Markov Chains). *Let P be transition matrix over \mathcal{S} . If, for all $x, x' \in \mathcal{S}$, there exists $t \in \mathbb{N}$ such that $P^t(x, x') > 0$, P is said to be irreducible.*

An infinitesimal generator Q is irreducible if, for all $x, x' \in \mathcal{S}$, there exists $t \in \mathbb{R}$ such that $H_t(x, x') > 0$.

Alternatively, a Markov chain is irreducible if its graph representation is strongly connected. Explicitly giving a path between two states is probably the most common method of proving irreducibility.

Property 1. *An irreducible transition matrix or infinitesimal generator has a unique stationary distribution.*

Note that irreducibility is not necessarily required in order to have uniqueness. The stationary distribution can be unique despite the presence of transient states (i.e. states to which the Markov chain will eventually never return), in which case the transition matrix is not irreducible.

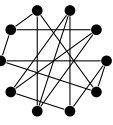
Definition 9 (Aperiodic Markov chain). *Let P be transition matrix over \mathcal{S} . For each $x \in \mathcal{S}$, define the period of x as*

$$T_x = \gcd\{t \in \mathbb{N} \mid P^t(x, x) > 0\}.$$

P is aperiodic if, for all $x \in \mathcal{S}$, $T_x = 1$.

For irreducible Markov chains over finite state spaces, this is equivalent to asking that there exists $t \in \mathbb{N}$ such that, for all $x, x' \in \mathcal{S}$, $P^t(x, x') > 0$.

There is no analog of periodicity for infinitesimal generators. More precisely, for any $x, x' \in \mathcal{S}$, if there exists $t \in \mathbb{R}$ such that $H_t(x, x') > 0$, then for all $s > 0$, $H_s(x, x') > 0$.



2. STATIONARY DISTRIBUTION

Property 2. *If the transition matrix P of a Markov chain $(X_t)_{t \in \mathbb{N}}$ is irreducible and aperiodic, then for all $x \in \mathcal{S}$,*

$$\lim_{t \rightarrow +\infty} \mathbf{P}(X_t = x) = \pi(x),$$

where π is the unique stationary distribution of P .

If the infinitesimal generator Q of a Markov chain $(X_t)_{t \in \mathbb{R}}$ is irreducible, then the same convergence holds, with π the unique stationary distribution of Q .

Note that neither of these conditions is very strong. To prove irreducibility, one generally picks a base state and shows that, for any other state, it is possible to reach and be reached by this base state. Constructing such paths is often quite straightforward. Similarly, aperiodicity can be established by finding a single $x \in \mathcal{S}$ such that $P(x, x) > 0$. If this is not feasible, it is also possible to consider the “lazy” version of the Markov chain, i.e. given by the transition matrix $\frac{1}{2}(P + I_{\mathcal{S}})$, where $I_{\mathcal{S}}$ is the identity matrix over \mathcal{S} . Another possibility is to consider a continuous time equivalent of that Markov chain.

In the case of Example 1, any state x can be reached from state 0 with x successive births, and can reach state 0 with x deaths, proving the chain to be irreducible. Furthermore, since $P(0, 0) = q > 0$, the chain is aperiodic.

For discrete time random walks on bipartite graphs, the period of every state is even; this is a typical example of a situation in which a lazy or continuous time random walk is required in order to have convergence in distribution.

We finish by presenting a specific category of Markov chains, called reversible Markov chains.

Property 3. *Let P be a transition matrix over \mathcal{S} , and π be a distribution over \mathcal{S} . If, for all $x, x' \in \mathcal{S}$, π satisfies*

$$\pi(x)P(x, x') = \pi(x')P(x', x),$$

then π is a stationary distribution of P , and P is said to be reversible.

For an infinitesimal generator Q , we ask that π satisfy

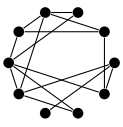
$$\pi(x)Q(x, x') = \pi(x')Q(x', x)$$

instead.

Working with reversible Markov chains is often convenient, because the stationary distribution is easily determined (up to a normalization constant).

Note that the Markov chains presented in Example 1 are reversible: for all $x \in \llbracket 0, N - 1 \rrbracket$, we have that

$$\frac{\pi_p(x+1)}{\pi_p(x)} = \frac{p}{q} = \frac{P(x, x+1)}{P(x+1, x)} \quad \text{and} \quad \frac{\pi_{\lambda, \mu}(x+1)}{\pi_{\lambda, \mu}(x)} = \frac{\lambda}{\mu} = \frac{Q(x, x+1)}{Q(x+1, x)}.$$



3 Markov Chain Mixing Time

One of the most common uses of Markov chains is the Monte Carlo sampling method, presented in Algorithm 1. It generates random samples from a target distribution π by computing the successive terms of an irreducible aperiodic Markov chain with stationary distribution π , up to a predefined rank T , and returning X_T . The distribution of X_T can be arbitrarily close to π , provided that T is large enough.

Algorithm 1 Monte Carlo Sampling Algorithm

```

function MONTECARLO( $\mathbf{A} = (\mathcal{S}, \mathcal{A}, D, \cdot), x_0, T$ )
   $X \leftarrow x_0$ 
   $t \leftarrow 0$ 
  while  $t < T$  do
     $A \leftarrow \text{DRAW}(D)$ 
     $X \leftarrow X \cdot A$ 
     $t \leftarrow t + 1$ 
  end while
  return  $X$ 
end function

```

How big does T need to be to achieve a desired precision? This is generally determined by computing the *mixing time* of the Markov chain.

In order to properly define the mixing time, we must first decide upon a metric between distributions. Multiple possibilities have been introduced in the literature, but we will focus on the most common approach: the total variation distance.

Definition 10 (Total Variation Distance). *Let μ and ν be two probability distributions over a finite set \mathcal{S} . The total variation distance between μ and ν is defined as*

$$\|\mu - \nu\|_{\text{TV}} = \max_{S \subseteq \mathcal{S}} |\mu(S) - \nu(S)|.$$

The right-hand side of this equation is not always easy to use, however, and we therefore often rely on the two following equivalent formulae instead:

$$\|\mu - \nu\|_{\text{TV}} = \frac{1}{2} \sum_{x \in \mathcal{S}} |\mu(x) - \nu(x)|$$

and

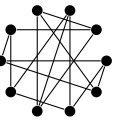
$$\|\mu - \nu\|_{\text{TV}} = \sum_{\substack{x \in \mathcal{S} \\ \mu(x) \geq \nu(x)}} (\mu(x) - \nu(x)).$$

Definition 11 (Distance to stationarity). *For a transition matrix P , the distance to stationarity at time t is defined as*

$$d(t) = \max_{x \in \mathcal{S}} \|P^t(x, \cdot) - \pi\|_{\text{TV}}.$$

For an infinitesimal generator with heat kernel H_t , this becomes

$$d(t) = \max_{x \in \mathcal{S}} \|H_t(x, \cdot) - \pi\|_{\text{TV}}.$$



4. SPECTRAL METHODS

An important property of the distance to stationarity (see [36], Lemma 4.12) is that, for any $c > 0$,

$$d(ct) \leq (2d(t))^c.$$

This implies that, if we know of a $t_* > 0$ such that $2d(t_*) < 1$, then for all $\epsilon > 0$, by setting

$$t_\epsilon = \frac{\log \epsilon}{\log(2d(t_*))} t_*,$$

we have that

$$d(t_\epsilon) < \epsilon.$$

Coming back to the Monte Carlo algorithm, we are then able to provide a termination time t_ϵ such that the distribution $P^{t_\epsilon}(x_0, \cdot)$ of the output satisfies $\|P^{t_\epsilon}(x_0, \cdot) - \pi\|_{\text{TV}} < \epsilon$.

Rather than having the definition of t_ϵ depend on the choice of t_* , we use a standardized value:

Definition 12 (Mixing Time). *Let t_{mix} be the first instant at which the distance to stationarity is at most $\frac{1}{4}$, i.e.*

$$t_{\text{mix}} = \min \left\{ t > 0 \mid d(t) \leq \frac{1}{4} \right\}.$$

t_{mix} is called the mixing time of the Markov chain.

We can now redefine t_ϵ as

$$t_\epsilon = \frac{\log \epsilon}{\log \frac{1}{2}} t_{\text{mix}} = \log_2 \left(\frac{1}{\epsilon} \right) \times t_{\text{mix}}.$$

Directly computing the mixing time is often not feasible. We present some common techniques used to establish an upper bound instead.

4 Spectral Methods

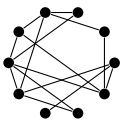
One of the fundamental results concerning Markov chains is the following:

Theorem 1 (Perron-Frobenius). *Let P be a transition matrix, and denote its eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{|\mathcal{S}|}$.*

- $\lambda_1 = 1$,
- P has non-negative left and right eigenvectors associated to λ_1 , and
- every (complex) eigenvalue λ of P satisfies $|\lambda| \leq 1$.

Furthermore:

- if P is irreducible, then λ_1 has multiplicity 1, and
- if P is also aperiodic, then every other eigenvalue λ satisfies $|\lambda| < 1$.



CHAPTER 1. INTRODUCTION

For an infinitesimal generator Q , let $L = -Q$ be the Laplacian matrix of the Markov chain, and denote its eigenvalues $\lambda'_1 \leq \lambda'_2 \leq \dots \leq \lambda'_{|\mathcal{S}|}$ (notice that the order is the opposite of that for transition matrices).

- $\lambda'_1 = 0$,
- L has non-negative left and right eigenvectors associated to λ'_1 , and
- every (complex) eigenvalue λ' of L satisfies $\Re(\lambda') \geq 0$, where $\Re(\lambda')$ denotes the real part of λ' .

Furthermore, if Q is irreducible, then λ'_1 has multiplicity 1, and every other eigenvalue λ' satisfies $\Re(\lambda') > 0$.

Notice that a non-negative left eigenvectors of λ_1 and λ'_1 are the (scaled) stationary distribution; this theorem proves the existence of such a distribution for any Markov chain over a finite state space. In both cases, the right eigenvector is necessarily the vector $\mathbf{1} = (1, \dots, 1)^T$.

The second half of the theorem furthermore proves Properties 1 and 2.

Definition 13 (Spectral Gap and Relaxation Time). *The spectral gap is equal to*

$$\gamma = \lambda_1 - \max(\lambda_2, |\lambda_{|\mathcal{S}|}|) > 0$$

for an irreducible aperiodic transition matrix P , and

$$\gamma = \lambda'_2 > 0$$

for an irreducible infinitesimal generator Q .

The relaxation time is then

$$t_{\text{rel}} = \frac{1}{\gamma}.$$

The relaxation time is closely linked to the mixing time. As such establishing bounds on the spectral gap is a common means of bounding the mixing time.

The relation between t_{rel} and t_{mix} is as follows:

Property 4. (see [36], Theorems 12.3 and 12.4) *For an irreducible aperiodic Markov chain with stationary distribution π , mixing time t_{mix} and relaxation time t_{rel} . We have that*

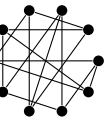
$$(t_{\text{rel}} - 1) \log 2 \leq t_{\text{mix}} \leq t_{\text{rel}} \times \log \left(\frac{4}{\pi_{\min}} \right),$$

where $\pi_{\min} = \min_{x \in \mathcal{S}} \pi(x)$ and \log denotes the natural logarithm.

Consider the case where π is the uniform distribution over \mathcal{S} . The above result tells us that the mixing time and the relaxation time differ by at most a factor $\log |\mathcal{S}|$:

$$\Omega(t_{\text{rel}}) \leq t_{\text{mix}} \leq O(t_{\text{rel}} \times \log |\mathcal{S}|).$$

Aside from bounding the mixing time, the spectral gap can also serve to directly bound the distance to stationarity.



4. SPECTRAL METHODS

Property 5 (From [42]). *For a reversible Markov chain with spectral gap γ and stationary distribution π , we have that*

$$d(t) \leq \frac{1}{2\sqrt{\pi_{\min}}} e^{-\gamma t}.$$

Proof. We prove this result in the continuous setting.

Let Q be the infinitesimal generator of the Markov chain, and $L = -Q$. Denote $D = \text{Diag}(\pi)$, and $A = D^{\frac{1}{2}} Q D^{-\frac{1}{2}}$. Since Q is reversible, A is necessarily symmetric, and can therefore be decomposed as

$$A = \sum_i (-\lambda_i) u_i u_i^T,$$

where the u_i form an orthonormal basis. Notice that the λ_i are precisely the eigenvalues of L , and namely that $\lambda_1 = 0$ and $u_1 = D^{\frac{1}{2}} \mathbf{1}$.

Let h_t be the distribution of the Markov chain at time t , i.e. $h_t = h_0 e^{tQ}$. We have that

$$h_t = h_0 D^{-\frac{1}{2}} e^{tA} D^{\frac{1}{2}} = h_0 D^{-\frac{1}{2}} \left(\sum_i e^{-t\lambda_i} u_i u_i^T \right) D^{\frac{1}{2}}.$$

Our goal is to bound $d(t) = \frac{1}{2} \sum_{x \in \mathcal{X}} |h_t(x) - \pi(x)|$. Notice that, for $i = 1$,

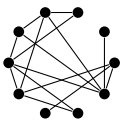
$$h_0 D^{-\frac{1}{2}} e^{-t\lambda_1} u_1 u_1^T D^{\frac{1}{2}} = \left(h_0 D^{-\frac{1}{2}} u_1 \right) \left(u_1^T D^{\frac{1}{2}} \right) = \mathbf{1} \times \pi.$$

Denoting $v_t^T = h_t - \pi$, we therefore have that

$$v_t = D^{\frac{1}{2}} \left(\sum_{i \geq 2} e^{-t\lambda_i} u_i u_i^T \right) D^{-\frac{1}{2}} h_0^T.$$

Let $\langle v_t, v_t \rangle_{\pi^{-1}} = v_t^T D^{-1} v_t$. We can bound $\langle v_t, v_t \rangle_{\pi^{-1}}$ as follows:

$$\begin{aligned} \langle v_t, v_t \rangle_{\pi^{-1}} &= h_0 D^{-\frac{1}{2}} \left(\sum_{i \geq 2} e^{-t\lambda_i} u_i u_i^T \right) D^{\frac{1}{2}} D^{-1} D^{\frac{1}{2}} \left(\sum_{i \geq 2} e^{-t\lambda_i} u_i u_i^T \right) D^{-\frac{1}{2}} h_0^T \\ &= h_0 D^{-\frac{1}{2}} \left(\sum_{i \geq 2} e^{-2t\lambda_i} u_i u_i^T \right) D^{-\frac{1}{2}} h_0^T \\ &\leq e^{-2\gamma t} \sum_{i \geq 2} h_0 D^{-\frac{1}{2}} u_i u_i^T D^{-\frac{1}{2}} h_0^T \\ &= e^{-2\gamma t} \sum_{i \geq 2} \left\langle D^{-\frac{1}{2}} h_0^T, u_i \right\rangle^2 \\ &\leq e^{-2\gamma t} \left\| D^{-\frac{1}{2}} h_0^T \right\|_2^2 \\ &= e^{-2\gamma t} \sum_{x \in \mathcal{X}} \frac{h_0(x)^2}{\pi(x)} \\ &\leq \frac{1}{\pi_{\min}} e^{-2\gamma t}. \end{aligned}$$



CHAPTER 1. INTRODUCTION

We now use the Cauchy-Schwartz inequality to conclude.

$$\begin{aligned}
 d(t) &= \frac{1}{2} \sum_{x \in \mathcal{S}} |v_t(x)| \\
 &= \frac{1}{2} \sum_{x \in \mathcal{S}} \frac{|v_t(x)|}{\sqrt{\pi(x)}} \sqrt{\pi(x)} \\
 &\leq \frac{1}{2} \sqrt{\sum_{x \in \mathcal{S}} \frac{v_t^2(x)}{\pi(x)}} \sqrt{\sum_{x \in \mathcal{S}} \pi(x)} \\
 &= \frac{1}{2} \sqrt{\langle v_t, v_t \rangle_{\pi^{-1}}} \\
 &\leq \frac{1}{2\sqrt{\pi_{\min}}} e^{-\gamma t}
 \end{aligned}$$

Note that this proof works in the discrete setting as well, using P rather than Q . The bound is then

$$d(t) \leq \frac{1}{2\sqrt{\pi_{\min}}} (1 - \gamma)^t,$$

from which we derive the same conclusion by observing that, for all $x \in \mathbb{R}$, $1 - x \leq e^{-x}$. \square

Direct computation of the spectral gap is not always computationally feasible. There are however a number of practical methods for establishing bounds on the spectral gap. We present one of these: Cheeger's inequality.

Definition 14 (Edge measure). *The edge measure μ of a Markov chain is defined as the rate at which each edge is traversed under the stationary distribution, i.e.*

$$\mu(x, x') = \pi(x)P(x, x')$$

for a transition matrix P , and

$$\mu(x, x') = \pi(x)Q(x, x')$$

for an infinitesimal generator Q .

We also denote

$$\mu(S, S') = \sum_{x \in S} \sum_{x' \in S'} \mu(x, x')$$

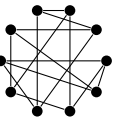
for all $S, S' \subset \mathcal{S}$.

Definition 15 (Isoperimetric Constant). *Let μ be the edge measure of a Markov chain over state space \mathcal{S} , and π be its stationary distribution. For every set $S \subseteq \mathcal{S}$ such that $S \neq \emptyset$, define the expansion of S as*

$$\phi(S) = \frac{\mu(S, \mathcal{S} \setminus S)}{\pi(S)}.$$

The isoperimetric constant of the Markov chain as a whole is

$$\Phi = \min_{1 \leq |S| \leq \frac{|\mathcal{S}|}{2}} \phi(S).$$



5. THE COUPLING AND PATH METHODS

The isoperimetric constant is sometimes referred to as the bottleneck ratio, the Cheeger constant, the conductance, or the expansion ratio.

For the simple, discrete time random walk on a graph $G = (V, E)$, the isoperimetric constant of a set of vertices $S \subseteq V$ can be rewritten

$$\phi(S) = \frac{|E(S, V \setminus S)|}{\sum_{x \in S} \deg x},$$

where $E(S, S')$ is the set of edges between S and S' , or even

$$\phi(S) = \frac{|E(S, V \setminus S)|}{\Delta|S|}$$

for Δ -regular graphs.

For the continuous time simple random walk, we instead have that

$$\phi(S) = \frac{|E(S, V \setminus S)|}{|S|}.$$

Property 6 (Cheeger's Inequality [30]). *Let Φ be the isoperimetric constant of a Markov chain with spectral gap γ . Then*

$$\frac{\Phi^2}{2} \leq \gamma \leq 2\Phi.$$

Combining Properties 4 and 6, we get that

$$\left(\frac{1}{2\Phi} - 1\right) \log 2 \leq t_{\text{mix}} \leq \log\left(\frac{4}{\pi_{\min}}\right) \frac{2}{\Phi^2}.$$

Applying this to situations where π_{\min} is of order $\frac{1}{|\mathcal{S}|}$ (e.g. when π is uniformly distributed), this yields that

$$\Omega\left(\frac{1}{\Phi}\right) \leq t_{\text{mix}} \leq O\left(\frac{\log|\mathcal{S}|}{\Phi^2}\right).$$

Note that, if Φ is bounded away from 0, the mixing time is at most logarithmic in the state space. Graphs which possess this property are often referred to as expanders [27].

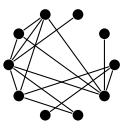
5 The Coupling and Path Methods

Another classical method for bounding the mixing time of discrete time Markov chains is the coupling method. First, we give a method for generating coupled Markov chains over a same probability space.

Definition 16 (Grand Coupling). *Let $\mathbf{A} = (\mathcal{S}, \mathcal{A}, D, \cdot)$ be a Markov automaton with transition matrix P , and $(A_t)_{t \in \mathbb{N}}$ be an infinite random word, i.e. a sequence of i.i.d. random variables distributed according to D . For every $x \in \mathcal{S}$, denote*

$$X_t^x = x \cdot A_1 \cdot A_2 \cdot \dots \cdot A_t.$$

The Markov chains in $\left(\left(X_t^x\right)_{t \in \mathbb{N}}\right)_{x \in \mathcal{S}}$ each have transition matrix P , but are correlated. They are referred to as a grand coupling.



CHAPTER 1. INTRODUCTION

Note that the correlation is such that, for all $x, x' \in \mathcal{S}$ and $s, t \in \mathbb{N}$, $s < t$,

$$X_s^x = X_s^{x'} \Rightarrow X_t^x = X_t^{x'}.$$

The usefulness of grand couplings in bounding the mixing time is a direct consequence of the following result:

Property 7. *Let P be a transition matrix, and $((X_t^x)_{t \in \mathbb{N}})_{x \in \mathcal{S}}$ be a grand coupling of P . Then*

$$d(t) \leq \max_{x, x' \in \mathcal{S}} \mathbf{P}(X_t^x \neq X_t^{x'}). \quad (1.3)$$

Equation (1.3) suggests that time it takes for a grand coupling to have all of its trajectories merge must somehow bound the mixing time. We therefore focus on the first instance at which this is the case.

Definition 17 (Coupling time). *Let P be a transition matrix, and $((X_t^x)_{t \in \mathbb{N}})_{x \in \mathcal{S}}$ be a grand coupling of P . The coupling time τ of the grand coupling is defined as the first instance at which all chains have merged:*

$$\tau = \min \{ t \in \mathbb{N} \mid \forall x, x' \in \mathcal{S}, X_t^x = X_t^{x'} \}.$$

Note that, unlike the mixing time, the coupling time is a random variable.

Corollary 1. *Let P be a transition matrix, $((X_t^x)_{t \in \mathbb{N}})_{x \in \mathcal{S}}$ be a grand coupling of P , and τ be the coupling time of $((X_t^x)_{t \in \mathbb{N}})_{x \in \mathcal{S}}$. Then*

$$d(t) \leq \mathbf{P}(\tau \geq t),$$

and, by means of Markov's inequality,

$$t_{\text{mix}} \leq 4\mathbf{E}[\tau].$$

Consider Example 1, with a grand coupling $((X_t^x)_{t \in \mathbb{N}})_{x \in \llbracket 0, N \rrbracket}$ defined by the Markov automaton from (1.2). Notice that, for all $x \in \llbracket 0, N \rrbracket$ and $t \in \mathbb{N}$, $X_t^x \geq X_t^0$. This implies that if, for some t , $X_t^0 = N$, then $\tau \leq t$.

For all $x \in \llbracket 0, N \rrbracket$, let t_x be the first instant at which $X_{t_x}^x = 0$, such that $\tau \leq t_N$. Let $s_x = t_x - t_{x-1}$. We have that

$$\mathbf{E}[t_x] = 1 + \frac{1}{2}\mathbf{E}[t_{x-1}] + \frac{1}{2}\mathbf{E}[t_{x+1}] \quad \Rightarrow \quad \mathbf{E}[s_x] = 2 + \mathbf{E}[s_{x+1}] \quad (1.4)$$

for $0 < x < N$, and

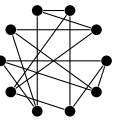
$$\mathbf{E}[t_N] = 1 + \frac{1}{2}\mathbf{E}[t_{N-1}] + \frac{1}{2}\mathbf{E}[t_N] \quad \Rightarrow \quad \mathbf{E}[s_N] = 2.$$

Summing (1.4) up to x yields that $\mathbf{E}[s_x] = \mathbf{E}[s_1] - 2(x-1)$. Substituting for $x = N$, we get that $\mathbf{E}[s_1] = 2N$, and therefore $\mathbf{E}[s_x] = 2N + 2 - 2x$. By summing over x :

$$\mathbf{E}[\tau] \leq \mathbf{E}[t_N] = \sum_{x=1}^N \mathbf{E}[s_x] = N^2 + 3N.$$

We conclude using Corollary 1:

$$t_{\text{mix}} \leq 4\mathbf{E}[\tau] \leq 4N^2 + 12N = O(N^2).$$



6. CONTRIBUTIONS

On a side note, notice that, unlike what one might expect, X_τ^x is *not* distributed according to the stationary distribution of the Markov chain, since its value is either 0 or N .

A similar technique can be used for continuous-time Markov chains. This is best achieved by means of a uniformization [23] of the Markov chain, and by coupling the resulting discrete time Markov chain and Poisson process.

We give one final technique, similar to coupling, for bounding the mixing time of discrete time Markov chains. It is used to compare the spectral gaps of two Markov chain, effectively bounding one with the other.

Consider two irreducible and aperiodic Markov chains over a same state space \mathcal{S} . We denote their transition matrices P and \tilde{P} , and their stationary distributions π and $\tilde{\pi}$. The aim is to establish a lower bound on the spectral gap γ of P , knowing the spectral gap $\tilde{\gamma}$ of \tilde{P} .

In order to do this, we decompose every transition of \tilde{P} as a sequence of transitions of P , called *paths*. Formally, let E (resp. \tilde{E}) be the set of edges $e = (x, x') \in \mathcal{S}^2$ such that $P(x, x') > 0$ (resp. $\tilde{P}(x, x') > 0$). For every $(\tilde{x}, \tilde{x}') \in \tilde{E}$, we choose a path $\Gamma_{(\tilde{x}, \tilde{x}')} = (x_0, x_1, \dots, x_k) \in \mathcal{S}^*$ such that:

- $x_0 = \tilde{x}$ and $x_k = \tilde{x}'$, and
- for all $i \in \llbracket 1, k \rrbracket$, $(x_{i-1}, x_i) \in E$.

For $\Gamma = (x_0, x_1, \dots, x_k)$, we denote $|\Gamma| = k$, and write that $e \in \Gamma$ if there exists $i \in \llbracket 1, k \rrbracket$ such that $e = (x_{i-1}, x_i)$.

Given this set of paths, let the *congestion ratio* be

$$B = \max_{e \in E} \left(\frac{1}{\mu(e)} \sum_{\substack{\tilde{e} \in \tilde{E} \\ \Gamma_{\tilde{e}} \ni e}} \tilde{\mu}(\tilde{e}) |\Gamma_{\tilde{e}}| \right).$$

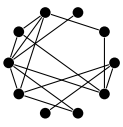
Property 8 (See [36], Theorem 13.23). *Given the above setup, we have that*

$$\gamma \geq \min_{x \in \mathcal{S}} \left(\frac{\tilde{\pi}(x)}{\pi(x)} \right) B^{-1} \tilde{\gamma}.$$

The efficiency of this method relies heavily on the ability to construct an efficient set of paths: these must be kept short, and be distributed evenly over E .

6 Contributions

In this thesis, we present three main contributions: a generalization of Cheeger's inequality to partial isoperimetric constants; a proof of rapid mixing for a given local edge dynamic, and an enhancement of the Propp and Wilson perfect sampling technique.



6.1 Partial Isoperimetric Constants

Consider a simple, continuous time random walk over a multi-graph $G = (V, E)$, and let $N = |V|$ and Δ be the maximum degree of G .

Definition 18 (Partial Isoperimetric Constants). *Let $k \in \llbracket 1, \frac{1}{2}N \rrbracket$. The k -th partial isoperimetric constant of G is*

$$\Phi_k = \min_{1 \leq |S| \leq k} \phi(S).$$

In particular, $\Phi_{\frac{N}{2}} = \Phi$.

We have seen that, by using Cheeger's inequality, we can bound the spectral gap using the isoperimetric constant. Using Property 5, we can then use this to bound the distance to stationarity. The following result shows how it is possible to directly use the partial isoperimetric constants to bound the distance to stationarity over small sets.

Theorem 2 (Reformulated from Theorem 9). *For all $k \leq \frac{N}{2}$ and $S \subset V$ such that $|S| \leq k$,*

$$H_t(\cdot, S) \leq \frac{|S|}{k+1} + \sqrt{k+1} e^{-\gamma_* t},$$

where

$$\gamma_* = \frac{1}{2\Delta} \Phi_k^2.$$

For comparison, combining Properties 5 and 6 gives the following result for the continuous time simple random walks:

$$H_t(\cdot, S) \leq \frac{|S|}{N} + \sqrt{N} e^{-\gamma t} \quad \text{with} \quad \gamma \geq \frac{\Phi^2}{2}.$$

6.2 Rapid Mixing of a Local Edge Dynamic

We consider graphs in which the edge sets evolve over time, forming a Markov chain (X_t) . Such dynamics are presented extensively in Chapter 2.

We focus primarily on local dynamics, i.e. in which every modification affects only a small, connected portion of the graph. Most dynamics we shall consider converge to the uniform distribution, under which Φ is bounded away from 0 with probability $1 - o(1)$. This may not be the case for the initial state, however, for which Φ could be of order $\frac{1}{N}$.

The question this raises is the following: can the mixing time of the graph-valued Markov chain be small, despite the poor initial conditions? Namely, can mixing occur with only a $O(\text{polylog } N)$ updates per edge, where $O(\text{polylog } N) = O(\log^k N)$ for some fixed k ?

We give a specific dynamic, represented in Figure 1.2, for which we prove this to be the case. The vertex set is denoted $[N] = \llbracket 1, N \rrbracket$, and the edge set is split into three: a set $E_o = \{(x, x+1) \mid x \in [N]\}$, with $N+1 \equiv 1$, that forms a static cycle; a set of blue edges $E_b = \{(x, b_x) \mid x \in [N]\}$; and a set of red edges $E_r = \{(x, r_x) \mid x \in [N]\}$, with b and r two permutations that evolve over time. The blue and red edges are called pointers, in the sense that one of their endpoints, referred to as their

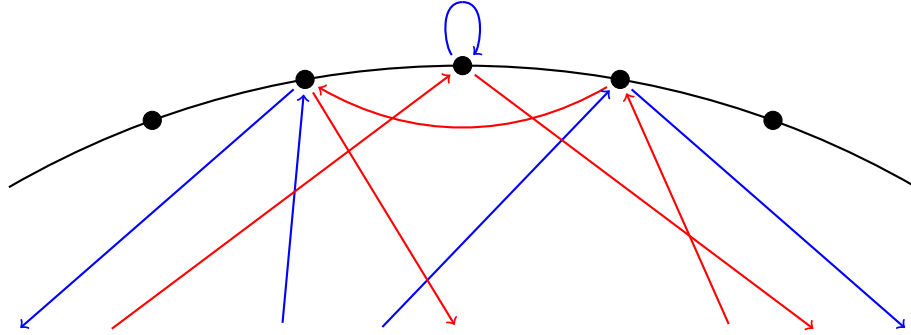
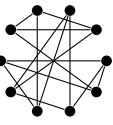


Figure 1.2 – Bi-color local dynamic model

destination, changes as the permutations are modified. They are nonetheless considered to be unoriented from a connectivity viewpoint.

The dynamic proceeds through alternating blue and red phases. In essence, edges of the active phase with adjacent destinations will repeatedly swap their destinations over time.

During a blue phase, only blue edges move, and red edges are static. Every static edge $(y, y') \in E_o \cup E_r$ maintains an exponential clock with rate 1; at every tick, the two blue edges $(x, b_x = y)$ and $(x', b_{x'} = y')$ are replaced with (x, y') and (x', y) . This boils down to applying the transposition (x, x') to b .

When the blue phase ends and the red phase begins, the roles of the red and blue edges are reversed. The dynamic is illustrated in Figure 1.3.

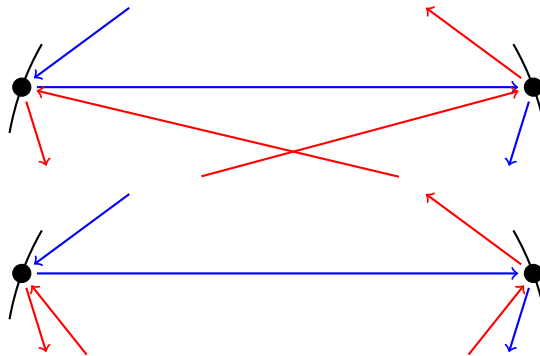


Figure 1.3 – Swapping two red pointers along a blue edge (red phase)

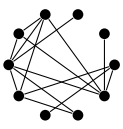
We prove the following result for this dynamic:

Theorem 3 (Adapted from Theorem 8). *Let $T = \log^a N$ where $a > 8$ is a constant. After $\log_2 N$ phases of length T , with high probability, the sets of blue and red pointers are uniformly and independently distributed permutations over the symmetric group \mathfrak{S}_N .*

This in turn gives that the number of modifications per edge required to mix is polylogarithmic in N , as stated in Corollary 2.

6.3 Oracle Sampling

Coupling from the past (CFTP) is a perfect sampling algorithm introduced by Propp and Wilson in 1996 [45]. To perform well, it requires the use of monotonous



CHAPTER 1. INTRODUCTION

Markov chains or bounding chains [28]. The details of these different approaches are given in Chapter 4.

In this thesis, we present an improvement of this algorithm, based on bounding chains.

Definition 19 (Bounding Chain). *Let $\mathbf{A} = (\mathcal{S}, \mathcal{A}, D, \cdot)$ be a Markov automaton. A bounding chain for \mathbf{A} is a pair (\mathcal{B}, \circ) such that \mathcal{B} is a subset of the power set of \mathcal{S} , and $\circ : \mathcal{B} \times \mathcal{A} \rightarrow \mathcal{B}$ is such that, for all $a \in \mathcal{A}$, $x \in \mathcal{S}$ and $B \in \mathcal{B}$,*

$$x \in B \Rightarrow x \cdot a \in B \circ a.$$

The usual bounding chains method is presented in Algorithm 2.

Algorithm 2 Coupling From the Past with Bounding Chains

```

function BOUNDEDCTFP( $\mathbf{A} = (\mathcal{S}, \mathcal{A}, D, \cdot)$ ,  $(\mathcal{B}, \circ)$ )
   $k \leftarrow 1$ 
   $U \leftarrow \epsilon$  Initialize with the empty word
   $B \leftarrow \mathcal{S}$ 
  repeat
     $V \leftarrow \text{DRAW}(D^{\otimes k})$  Draw a  $k$  letter prefix  $V$ 
     $U \leftarrow V \circ U$  Prepend  $V$  to  $U$ 
     $B \leftarrow \mathcal{S} \circ U$  Compute the bounding chain
     $k \leftarrow 2k$  Double the period
  until  $|B| = 1$ 
  return  $\text{UNIQUEELEMENTOF}(B)$ 
end function

```

The key feature in our approach is the ORACLESAMPLE function. First, introduce a new letter, \sharp , such that

$$\forall x \in \mathcal{S}, x \cdot \sharp = x \quad \text{and} \quad \forall B \in \mathcal{B}, B \circ \sharp = B.$$

For every $p \in (0, 1)$, let D_p be the distribution such that $D_p(\sharp) = p$ and, for all $a \in \mathcal{A}$, $D_p(a) = (1-p)D(a)$.

Let $p \in (0, 1)$ and $\mathcal{A}_+ \subseteq \mathcal{A}$. ORACLESAMPLE(p, \mathcal{A}_+) returns a random letter, distributed according to D_p conditioned on being in \mathcal{A}_+ .

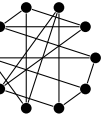
For every bounding state $B \in \mathcal{B}$, define the set of active letters in state B as

$$\mathcal{A}_+^B = \{a \in \mathcal{A} \mid B \circ a \neq B\} \cup \{\sharp\}.$$

We propose Algorithm 3. The structure is similar to that of the previous algorithm. It differs in regards to how the prefix is generated (DRAWPREFIX) and prepended to the previous word (UPDATE).

DRAWPREFIX constructs a prefix in which only active letters appear. This requires computing transitions at the same time as the letters are drawn. Notice that, if every letter are always active, the expected length of DRAWPREFIX(k) is 2^k , mimicking the period doubling of the base algorithm.

UPDATE is somewhat trickier. It modifies the word from the previous iteration, to which the prefix is to be prepended, so that the final word is also composed of

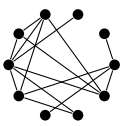


Algorithm 3 Coupling From the Past with Oracle Sampling

function ORACLECFTP($\mathbf{A} = (\mathcal{S}, \mathcal{A}, D, \cdot), (\mathcal{B}, \circ)$)
 $k \leftarrow 1$
 $U \leftarrow \epsilon$
 $B \leftarrow \mathcal{S}$
repeat
 $(V, B') \leftarrow \text{DRAWPREFIX}(k)$
 $(U', B) \leftarrow \text{UPDATE}(U, k-1, B, B')$
 $U \leftarrow V \odot U'$
 $k \leftarrow k+1$
until $|B| = 1$
return $\text{UNIQUEELEMENTOF}(B)$
end function

function DRAWPREFIX(k)
 $V \leftarrow \epsilon$
 $B \leftarrow \mathcal{S}$
repeat
 $A \leftarrow \text{ORACLESAMPLE}(2^{-k}, \mathcal{A}_+^B)$
 $V \leftarrow V \odot A$
 $B \leftarrow B \circ A$
until $A = \#$
return (V, B)
end function

function UPDATE(U, k, B, B')
if $k = 0$ **then**
return (ϵ, B')
end if
 $A \leftarrow \text{ORACLESAMPLE}(2^{-k}, \mathcal{A}_+^B \cup \mathcal{A}_+^{B'})$
if $A \notin \mathcal{A}_+^B$ **then**
return $A \odot \text{UPDATE}(U, k, B, B' \circ A)$
end if
 $H \leftarrow \text{HEAD}(U), T \leftarrow \text{TAIL}(U)$ *H is the first letter of U, T is the rest of U*
if $H = \#$ **then**
return $\# \odot \text{UPDATE}(T, k-1, B, B')$
else if $H \in \mathcal{A}_+^{B'}$ **then**
return $H \odot \text{UPDATE}(T, k, B \circ H, B' \circ H)$
else
return $\text{UPDATE}(T, k, B \circ H, B')$
end if
end function



CHAPTER 1. INTRODUCTION

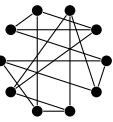
only active letters. At the same time, it introduces new letters in such a way as to produce an unbiased output.

These operations are detailed in depth in Chapter 4. We also prove the correctness of the algorithm. More specifically, we show the following result.

Theorem 4 (Transcribed from Theorem 12). *If there exists $u \in \mathcal{A}^*$ such that $\mathcal{S} \circ u$ is a singleton, then*

1. *ORACLECFTP terminates in finite expected time.*
2. *The output of ORACLECFTP distributed according to π , the stationary distribution of \mathbf{A} .*

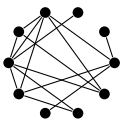
Along with the proof of this result, we give some indications as to how to bound the running time of the algorithm. We also provide some experimental results for generating random independent sets from a graph.



Chapter 2

Taxonomy of Dynamic Graphs

1	Introduction	22
2	The Zoo	23
2.1	The Erdős-Rényi Dynamic	23
2.2	Exponential Random Graphs	24
2.3	Pointer Models for Peer-to-Peer and Social Networks . .	24
2.4	Sliding Edges	25
2.5	The Interchange Process	27
2.6	Static Structures	27
2.7	Dynamic Triangulations	29
3	Genotypes and Phenotypes	30
3.1	Genotypes	30
3.2	Phenotypes	33
4	Case Analysis	36
4.1	Erdős-Rényi dynamic	36
4.2	The Interchange Process	39
4.3	Pointers on Rings	40



1 Introduction

Dynamic graphs have received a large amount of interest over the past years, in great part due to the central role they play in the design and analysis of distributed networks. This is especially true in fields such as social networks and communication infrastructures akin to peer-to-peer networks, but also in other domains, including biochemistry and statistical physics.

The resulting models have led to a better understanding of some key observations of real-world networks. Notorious examples include the power-law degree distribution [1], the small-world property [49], and high clustering [26]. This has served as a basis for the analysis and optimization of common algorithms such as gossiping [21, 22] and load balancing [46].

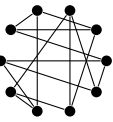
For the most part, research has been aimed at designing efficient algorithms for constructing a target topology. This approach is most common for distributed communication networks, such as peer-to-peer networks, in which the geometry of the overlay network plays a crucial role. In [5], for example, the authors consider the Skip Graph data structure for improving the resiliency of peer-to-peer queries in a faulty environment, and [29] gives an efficient distributed algorithm for constructing such structures. Such studies often make the implicit assumption that the network nodes are cooperative and follow a consensual strategy.

Conversely, non-committed dynamics, in which the evolution of the system is not specifically engineered to favor any given feature, have also had their share of attention. These serve to justify the observed properties of networks with unspecified evolutions, such as social networks. In this situation, dynamics are often presented as naturally-arising Markov chains over a broad set of admissible graphs (e.g. the set of connected graphs over N vertices). This is namely the case in [7], which considers a Glauber dynamic over the set of connected graphs with a natural bias in favor of triangles and other small connected subgraphs. Another notable example is [3], which focuses on local edge flips over Δ -regular graphs, and proves that such a dynamic converges in $O(n^2 \Delta^2 \sqrt{\log n})$ iterations. It is worth mentioning that with this dynamic, the limit distribution is the uniform distribution over Δ -regular graphs, which is known to give expanders with high probability.

Regardless of the nature of the dynamic, most studies aim to better understand the causality between the local characterization of the model, i.e. the dynamic, and its consequences on a global scale.

The survey This chapter aims to give the reader an overview of different possible models for dynamic graphs. It focuses on simple, randomized dynamics which yield the usual global properties one has come to expect: power-law degree distribution, small-world phenomenon, good expansion, etc.

Section 2, the Zoo, introduces a wide range of dynamics. The aim is twofold: to present the different parameters that can be adjusted when choosing the dynamic that best suits one's needs, and to familiarize the reader with the consequences such dynamics have on the global topology of the graph. Sections 3.1 and 3.2 then take this a step further. In Section 3.1, dynamics are studied at a microscopic level, grouping design decisions together to form *genotypes*. This



2. THE ZOO

is complemented by Section 3.2, in which recurring macroscopic features are categorized as *phenotypes*. Finally, Section 4 states a number of results concerning the mixing time of a selection of models from Section 2. The aim is to give some insight as to the impact of the choice of the dynamic on the convergence rate of the resulting dynamic graph, further bridging the gap between genotypes and phenotypes.

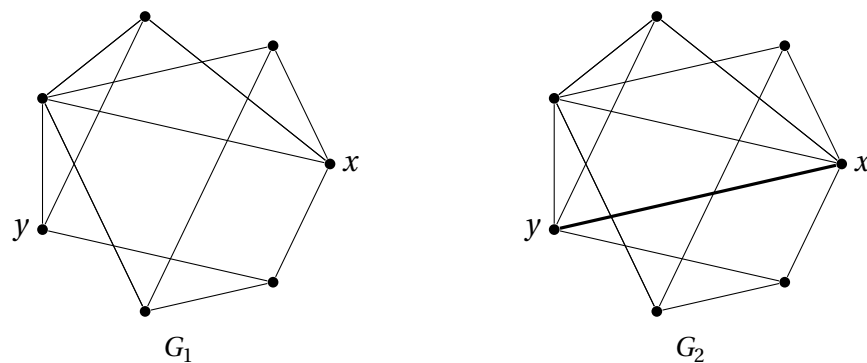
2 The Zoo

In this section, we present a wide variety of dynamic graphs. Note that, throughout this work, we only consider graphs over a fixed vertex set of size N ; the study of dynamic vertex sets falls outside of the scope of this paper. In a similar manner, we do not always require graphs to be simple: the appearance of multi-edges and self loops is deemed acceptable in most situations. Should the reader wish to enforce such a property, most of the dynamics presented here can be easily adapted (for example via rejection sampling) so as to satisfy that constraint.

2.1 The Erdős-Rényi Dynamic

The Erdős-Rényi random graph [16] is widely accepted as the simplest model of random graphs. We present here what we believe to be its dynamic counterpart.

Given an undirected simple graph $G = (V, E)$ and a fixed parameter p in $]0; 1[$, the dynamic is the following: Choose two vertices x and y uniformly and independently at random in V . With probability p , add (x, y) to E (if the edge was already there, do nothing), otherwise remove (x, y) from E (likewise, if the edge was already absent, do nothing). Applying this dynamic over and over again yields a Markov chain whose stationary distribution is the alternate Erdős-Rényi distribution $\mathcal{G}(N, p)$.

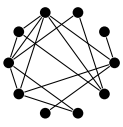


$$P_{G_1, G_2} = \frac{2}{N(N-1)} p$$

$$P_{G_2, G_1} = \frac{2}{N(N-1)} (1-p)$$

Transition probabilities

Figure 2.1 – The Erdős-Rényi Dynamic



This dynamic can easily be modified to converge to the original Erdős-Rényi distribution, $\mathcal{G}(N, M)$, i.e., the uniform distribution over graphs with N vertices and M edges. This is achieved by selecting not only x and y uniformly at random, but also an independent uniform edge e in E . If (x, y) is an edge, do nothing, otherwise remove e from E and add (x, y) in its place.

On a final note, recall that, in order to obtain a connected graph with high probability, p should be at least $\frac{(1+\epsilon)\log n}{n}$ for some positive ϵ [16].

2.2 Exponential Random Graphs

This model, studied thoroughly in [7], is similar to the $\mathcal{G}(N, p)$ model in that, at each iteration, an edge is chosen uniformly at random, and either added to or removed from the graph. Rather than fixing a probability p of including the edge, however, the chances of accepting the new edge depend on its impact on the local graph structure.

Formally, consider a finite set of small graphs $(g_i)_{i \in I}$, called configurations. The most common configurations include the edge (2-clique) and the triangle (3-clique). Each configuration is furthermore assigned a weight α_i . For all $i \in I$, denote $c_i(G)$ the number of occurrences of g_i in G , and define the energy of the graph as

$$H(G) = \sum_i \alpha_i c_i(G).$$

When updating edge e , denote G_{e+} and G_{e-} the graphs consisting of G with and without edges e , respectively, and $\partial_e H(G) = H(G_{e+}) - H(G_{e-})$. The probabilities of including e or not are

$$\mathbf{P}(\text{include } e) = \frac{e^{\partial_e H(G)}}{1 + e^{\partial_e H(G)}}, \quad \mathbf{P}(\text{exclude } e) = \frac{1}{1 + e^{\partial_e H(G)}}.$$

Note that computing $\partial_e H(G)$ can be achieved by only enumerating the number of occurrences of configurations that contain edge e . As such, if all configurations have diameter at most δ , than these probabilities can be computed by considering only the ball of radius δ around edge e .

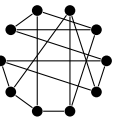
The stationary distribution is then

$$\pi(G) = \frac{1}{Z} e^{H(G)},$$

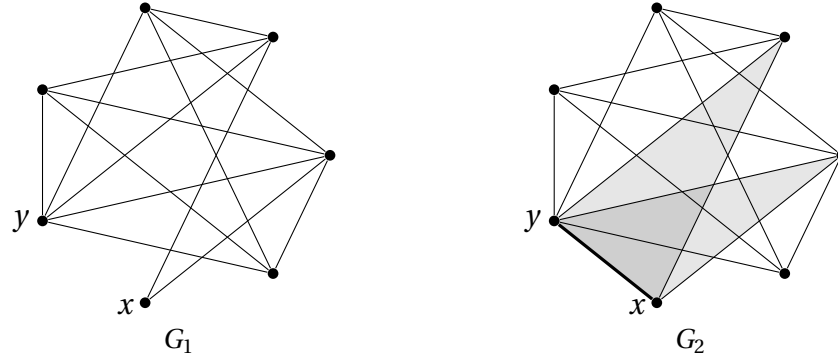
where Z is a normalization constant, called the partition function. These type of distributions and update rules are respectively referred to as Gibbs measures and Glauber dynamics.

2.3 Pointer Models for Peer-to-Peer and Social Networks

Consider a model in which every node maintains a fixed number of outgoing edges, called *pointers*, which it updates over time. In this model, each edge has one fixed extremity, its source, and one dynamic end, its destination. Also, whereas every node is the source of a fixed number of edges, it can serve as a destination for a varying number of these. Note that, despite pointers having an



2. THE ZOO



Consider g_1 to be the edge graph and g_2 the triangle graph.

$$\begin{aligned}\partial_e H(G) &= \alpha_1 + 2\alpha_2 \\ P_{G_1, G_2} &= \frac{1}{M} \frac{e^{\alpha_1 + 2\alpha_2}}{1 + e^{\alpha_1 + 2\alpha_2}} \\ P_{G_2, G_1} &= \frac{1}{M} \frac{1}{1 + e^{\alpha_1 + 2\alpha_2}}\end{aligned}$$

Transition probabilities

Figure 2.2 – Exponential Graphs

asymmetric dynamic, they are still considered to be non-oriented edges with respect to graph connectivity.

This structure naturally arises in peer-to-peer networks, where every client maintains a limited number of connections that it then manages on its own. The overall layout of the network is thus constructed in a distributed manner. The same is true in social networks, where individuals have a certain *budget of attention* [31], which they must distribute amongst their contacts.

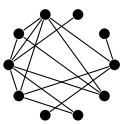
Regarding the dynamic, a uniformly distributed pointer (x, p_x) is updated at each time step. Its new destination p'_x is chosen according to a distribution P , which depends on the current configuration of graph. Generally, the probability $P(x, p_x, p'_x)$ of the destination going from p_x to p'_x is a function of the degree of p'_x in the graph; if it decreases with the degree, then the stationary distribution will tend to have a balanced degree distribution, whereas if it increases with the degree, the asymptotic degree distribution is likely to be exponential.

If P is the uniform distribution over the vertices, then the destination of each edge is uniformly distributed under the stationary distribution, and the overall inbound degree distribution is a multinomial with parameters M and N , where M is the total number of edges. The inbound degree of a given vertex is therefore asymptotically a Poisson random variable of parameter $\frac{M}{N}$.

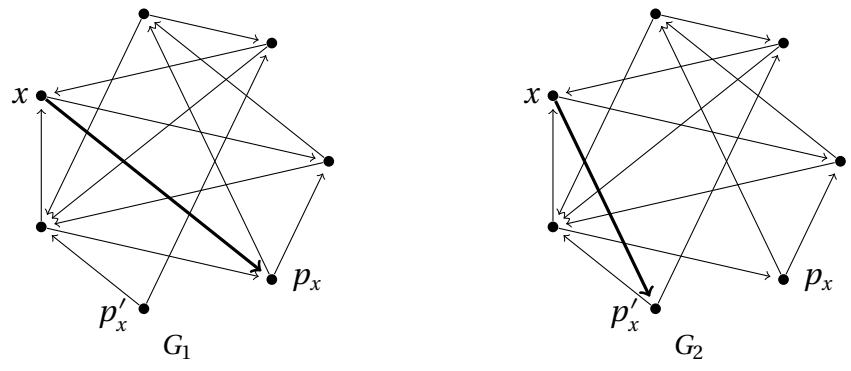
This model can be extended by having each node maintain a dynamic number of pointers, rather than keeping a fixed number of these.

2.4 Sliding Edges

In the previous models, our edge update rules rely on the fact that the global structure of the graph is known, and namely that the vertex set is accessible. This is often too strong of an assumption, however, which this fourth model aims to



CHAPTER 2. TAXONOMY OF DYNAMIC GRAPHS



$$P_{G_1, G_2} = \frac{1}{M} \frac{1}{N}$$

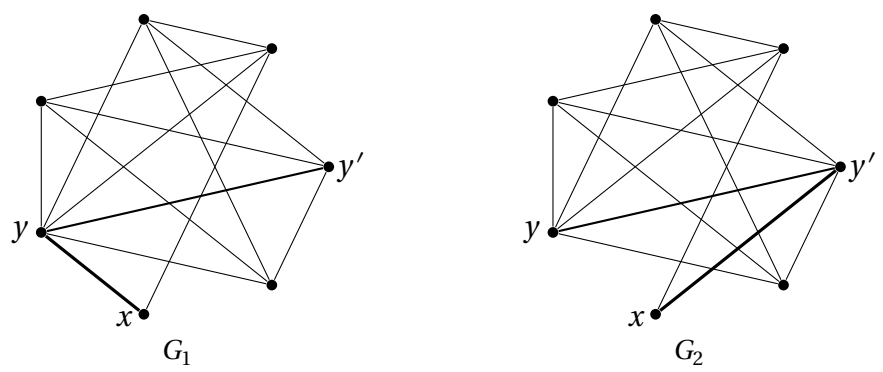
$$P_{G_2, G_1} = \frac{1}{M} \frac{1}{N}$$

Transition probabilities

Figure 2.3 – The Peer-to-Peer Model

bypass.

For any graph G and vertex x , denote $N_G(x)$ the set of neighbors of x in G , and $\deg_G(x) = |N_G(x)|$ its degree. The dynamic is as follows: choose an oriented edge (x, y) uniformly at random, and replace it with an edge (x, y') , where y' is chosen uniformly at random over $N_G(y) \cup \{y\} \setminus \{x\}$, i.e., the set comprising y and its neighbors, x excluded. With this policy, the updated edge need only know its local surroundings to change its target.

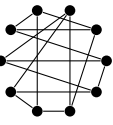


$$P_{G_1, G_2} = \frac{1}{2M} \frac{1}{\deg_{G_1}(y)}$$

$$P_{G_2, G_1} = \frac{1}{2M} \frac{1}{\deg_{G_2}(y')}$$

Transition probabilities

Figure 2.4 – Sliding Edges



2. THE ZOO

The stationary distribution for this dynamic is of the form

$$\pi(G) = \frac{1}{Z} \prod_{x \in V} \deg_G(x)!$$

This gives an exponential degree distribution. Other distributions can be obtained by drawing y' according to a non-uniform distribution P over $N_G(y) \cup \{y\} \setminus \{x\}$, as in the previous model. For example, take

$$P(y') = \frac{1}{\deg(y) - 1} \frac{1}{\deg(y')}$$

for every $y' \in N_G(y) \setminus \{x\}$, and

$$P(y) = 1 - \sum_{y' \in N_G(y) \setminus \{x\}} P(y').$$

The first factor in the expression of $P(y')$ consists in choosing y' uniformly as before, but the second factor indicates that the transition is only accepted with a probability inversely proportional to the destination's current degree. Using the distribution P rather than the uniform one at each iteration gives a dynamic whose stationary distribution is uniform over all regular connected graphs with the same number of edges.

2.5 The Interchange Process

The previous model can furthermore be adapted to preserve the degrees of individual vertices. This can for in particular be used to construct a dynamic over the set of Δ -regular graphs, in which case we refer to this dynamic as the Δ -regular interchange process.

The transition differs in that rather than having one edge slide along another, two edges are moved along a third one, in opposite directions. Formally, choose an edge (y, y') uniformly at random, and some neighbors x and x' of y and y' respectively, chosen uniformly at random (y' and y excluded). Replace the edges (x, y) and (x', y') by (x, y') and (x', y) . If either of these edges was already present in the graph, you can choose to systematically accept or refuse the change, respectively allowing for or preventing multi-edges and self-loops.

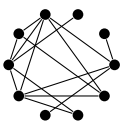
A simplified version of this dynamic, the *non-local interchange process*, is obtained by sampling (x, y) and (x', y') uniformly at random, without requiring that y and y' be neighbors. Both of these dynamics have been widely studied [10, 17, 3, 38, 39], often referred to as the flip (local) and switch (non-local) dynamics.

This simple dynamic converges to the uniform distribution over the set of connected graphs with the degree distribution of the original graph.

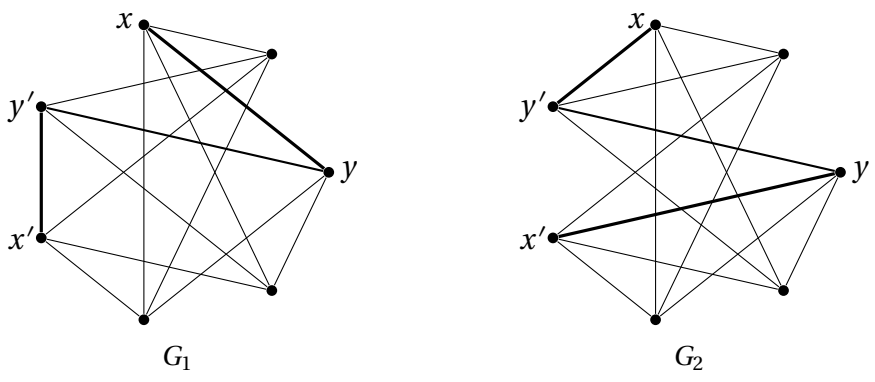
2.6 Static Structures

One of the inherent risks of local dynamics is the loss of connectivity. The locality constraint implies that, in such cases, this connectivity cannot be restored.

A possible enhancement to the previous models is the use of static edges, i.e., edges which are never moved. By choosing these edges so that they alone



CHAPTER 2. TAXONOMY OF DYNAMIC GRAPHS



$$P_{G_1, G_2} = \frac{1}{M(\Delta - 1)^2}$$

$$P_{G_2, G_1} = \frac{1}{M(\Delta - 1)^2}$$

Transition probabilities

Figure 2.5 – The Δ -Regular Interchange Process

form a connected graph over the vertex set, we can ensure that the dynamic itself is irreducible. Note that the non-static edges (aka dynamic edges) do not necessarily form a connected graph. It is however often possible to target stationary distributions for which this property holds with high probability.

Static structures can also greatly simplify the study of local graph dynamics, as they help “anchor” vertices in a persistent geometry. Common structures include spanning trees, paths and cycles.

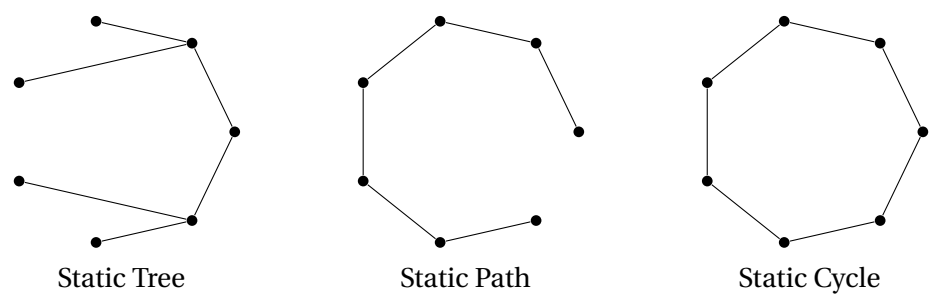
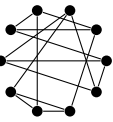


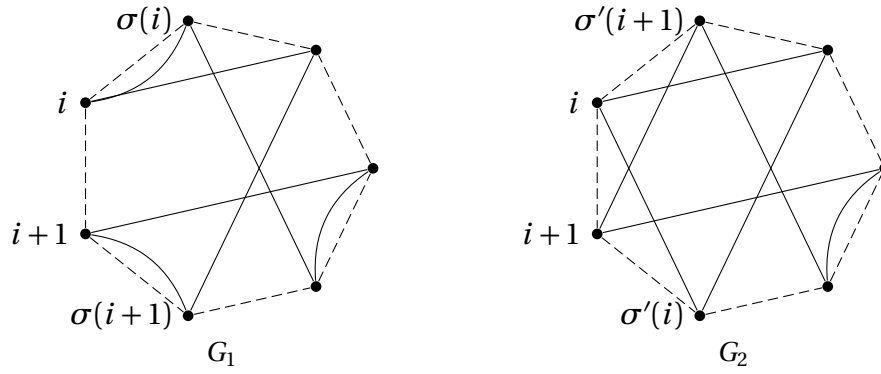
Figure 2.6 – Static Structures

We now give a sample dynamic based on a static structure. Consider that $V = \{1, 2, \dots, N\}$ to ease notation, and that the edges are partitioned into a static ring $R = \{(i, i + 1) \mid i \in V\}$, with $N + 1 \equiv 1$, and a dynamic set $C_\sigma = \{(i, \sigma(i)) \mid i \in V\}$, where σ is a permutation over V . The dynamic, thoroughly studied in [12], is the following: choose a static edge $(i, i + 1)$ uniformly at random, and denote $\tau_{\sigma(i), \sigma(i+1)}$ the transposition of $\sigma(i)$ and $\sigma(i + 1)$. Replace C_σ with $C_{\sigma'}$, where $\sigma' = \sigma \circ \tau_{\sigma(i), \sigma(i+1)}$.

As always for such symmetric dynamics, the stationary distribution is the uniform distribution.



2. THE ZOO



$$P_{G_1, G_2} = \frac{1}{N}$$

$$P_{G_2, G_1} = \frac{1}{N}$$

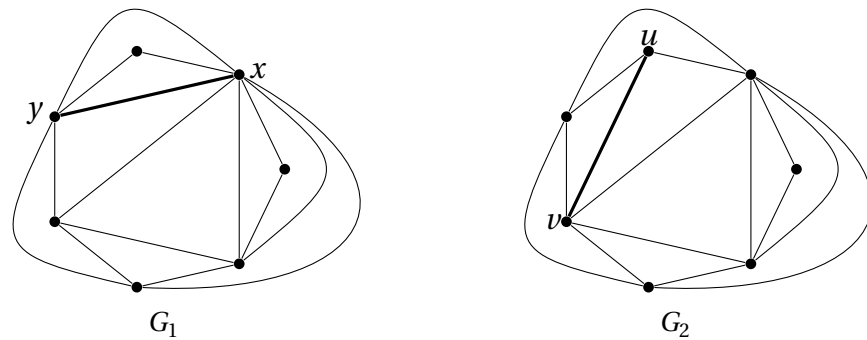
Transition probabilities

Figure 2.7 – Static Ring (dashed) with Permutation

2.7 Dynamic Triangulations

This last dynamic somewhat differs from the previous ones in that it deals with planar graphs, namely triangulations. The proposed update scheme converges to the uniform distribution over all triangulations over N vertices.

At each iteration, an edge (x, y) is chosen uniformly at random. Denote u and v the two vertices such that (x, y, u) and (x, y, v) are faces of the triangulation. Remove (x, y) from the set of edges, and replace it with (u, v) . The resulting graph is still a triangulation.



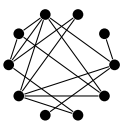
$$P_{G_1, G_2} = \frac{1}{M}$$

$$P_{G_2, G_1} = \frac{1}{M}$$

Transition probabilities

Figure 2.8 – Planar Triangulations

This dynamic converges to the stationary distribution over the set of planar triangulations over N vertices.



3 Genotypes and Phenotypes

The examples from the previous section outline the need for a formal description of dynamic graphs. This could then allow us to classify these models, and possibly identify classes over which certain properties can be generalized. Such descriptions often fall in one of the following two categories.

- On the one hand, those that describe the nature of the dynamics themselves, i.e., the set of rules which dictate the evolution of the graph. These often operate at a microscopic level, around a single vertex or edge, and form what we hereon refer to as the graph's *genotype*.
- On the other hand, there is the set of macroscopic properties of the graph, which do not define the dynamic as such, but a rather a consequence of the microscopic evolution. These form the graph's *phenotype*.

Genotypes can include things like fixed vertex degrees, static edges and interchange dynamics, whereas phenotypes refer to properties such as a uniform stationary distribution, degree power laws and mixing times.

We still consider that all our dynamics operate over graphs with N vertices and, for fixed edge counts, M edges.

3.1 Genotypes

A dynamic graphs genotype is primarily defined by the nature of its vertices, its edges and its transitions.

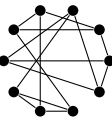
Vertices

As we limit our study to the case of graphs over fixed vertex sets, describing the evolution of vertices is quite straightforward.

Fixed degrees This category encompasses all dynamics where individual vertices' degrees are fixed. A typical example would be the Δ -regular interchange process, and most dynamics that fall under this heading make use of some variation of the interchange process to preserve degree distributions.

Non-homogeneous degrees Whereas most dynamics make no distinction between different vertices, taking only into account their degrees, some models partition the vertex sets into subsets with differing behaviors. In social networks, for example, studies often distinguish between content creators, relayers and consumers, each with their distinct behavioral patterns. This is the basis for models with underlying communities. Note that this partition can be replaced with continuous properties, such as different probabilities of accepting new neighbors, differing affinities for a set of features, etc.

Underlying geometry Some dynamics have their vertices embedded inside a metric space, which allows for defining dynamics with attraction or repulsion phenomena: edges can be more or less likely to link two given vertices depending on their embedded positions.



Edges

The nature of edges is somewhat richer than that of vertices, namely because they form the dynamic component of the graph.

Fixed edge count vs. birth-and-death One of the major distinctions between dynamics is the edge count. With fixed edge counts, the dynamic moves edges around, and it is often possible to map edges before the transition to edges after it. With unconstrained edge counts, edges can also be added and removed, and one quickly loses the ability to track individual edges.

Static edges, dynamic edges and pointers In the case of a fixed edge count (typically when moving edges around), one can distinguish between three types of edges:

Static edges, with two fixed ends,

Pointers, with one fixed end and one dynamic extremity, and

Dynamic edges, with two dynamic extremities.

These edges generally serve very distinct roles and carry different implications: static edges serve as a base structure over which the dynamic unfolds; pointers are maintained by a source vertex, allowing for vertex-driven dynamics; finally, dynamic edges cover a larger scope than pointers, but are also harder to keep track of, and therefore to analyze and interpret.

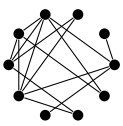
Multi-graphs Simple graphs are often more convincing by nature, as it is not always clear how multi-edges and self-loops should be interpreted. Nonetheless, multi-graphs can be easier to manipulate from a mathematical standpoint. The passage from multi-graphs to simple graphs can often be achieved by simply rejecting transitions which would violate the structural constraints.

Oriented edges The use of oriented edges can have an important impact not only on dynamics, e.g. by allowing sliding edges to follow other edges in only one direction, but also on macroscopic considerations, such as connectivity and expansion.

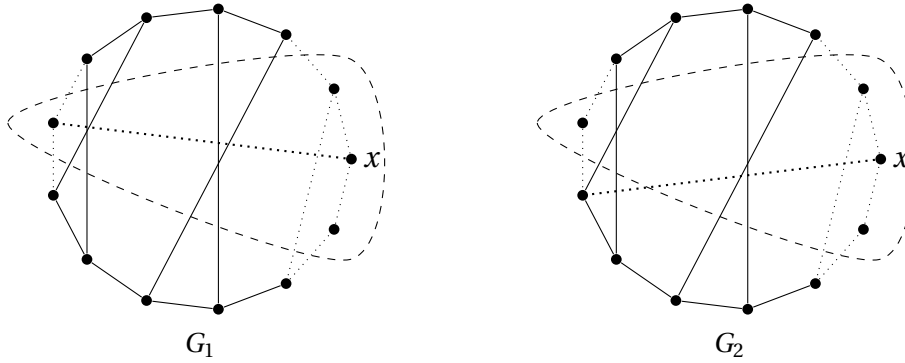
Labeled edges Just as vertices can differ in nature, edges need not all be identical. A graph can contain multiple edge sets, each with their own dynamic. These dynamics can naturally be linked, for example by imposing that two edge sets never cover a same vertex, that edges never slide along other edges of the same set, or by changing the nature of an edge during a transition.

Transitions

While the possibilities for designing transitions are near endless, certain recurrent patterns can be identified.



Local vs. non-local Looking at the examples from the previous section, there is a clear distinction between dynamics which operate in a localized portion of the graph, and others which can span the entire graph. Formally, we say a dynamic is *local* if there exists a fixed integer r such that, for every transition from a state G_1 to a new state G_2 , there exists a vertex x that satisfies $G_1 \setminus B_{G_1}^r(x) = G_2 \setminus B_{G_1}^r(x)$, where $B_G^r(x)$ is the ball of radius r centered on x in G .



$$G_1 \setminus B_{G_1}^2(x) = G_2 \setminus B_{G_1}^2(x)$$

The dynamic is local

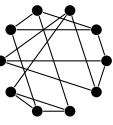
Figure 2.9 – Local Dynamic

Half-edge transitions When moving an edge, some common dynamics keep one extremity of the edge unmodified. It is often more practical to model these dynamics as operating on half-edges, i.e., a single edge extremity, rather than on full edges. Dynamics dealing with pointers naturally fall under the scope of this definition, whereas edge additions or removals never do.

With each half-edge attached to a single vertex, the set of half-edges at a given vertex can be represented as a set of labels; transitions can then be interpreted as moving these labels.

Sliding edges Amongst local dynamics, sliding edges are possibly the most natural approach. These are half-edge transitions in which the original and updated vertices to which the half-edge is attached are linked by another edge. The modified half-edge effectively slides along this other edge.

Interchanging edges The interchange process is the adaptation of half-edge transitions to graphs with fixed degree distributions: rather than move a single half-edge, select two of these and swap their positions. If these two vertices are guaranteed to always be connected by yet another edge, as is the case for sliding edges, then this dynamic is local. The most common occurrence is the Δ -regular interchange process detailed in the previous section.



3. GENOTYPES AND PHENOTYPES

Degree-dependant transitions As shown in some of the examples from the previous section, it is often possible to tinker with a given dynamic through accepting or rejecting simple transitions. The most common decision factor is the respective degrees of the intervening vertices. This is a recurrent means of adapting dynamics to a given target distribution.

Update distribution All of the update mechanisms presented in the previous section share a common aspect: at each iteration, one starts by selecting one or more edges or vertices at random, often following a uniform distribution. This initial decision of what to update plays a crucial role in reversibility.

Another essential role of the update distribution is when considering continuous time equivalents of the dynamics presented here. For a dynamic where edges were chosen uniformly at random, an exponential timer is now associated to each edge, and a transition occurs when that timer expires. For dynamics based on uniformly distributed vertices, the timers are placed on the vertices. Choosing the right expiration rate for vertex timers is primordial in order to ensure the reversibility of these continuous time dynamic graphs.

Combinatorial updates To finish, some dynamics rely on some form of combinatorial structure underlying the edge set. This was the case for the example comprising a static ring and a permutation. Such structures imply additional constraints on dynamics, and should be considered with care.

They furthermore highlight an important caveat in our definition of local dynamics. Consider the following model: a graph consisting of a static ring and a cycle $\{(\sigma(i), \sigma(i+1)) \mid i \in \{1, 2, \dots, N\}\}$, with $N+1 \equiv 1$. The dynamic consists in selecting an vertex i uniformly at random, then setting

$$\sigma_2 = \begin{pmatrix} 1 & 2 & \dots & \sigma_1^{-1}(i) & \sigma_1^{-1}(i+1) & \dots & \sigma_1^{-1}(i+1)-1 & \sigma_1^{-1}(i+1) & \dots & N \\ \sigma_1(1) & \sigma_1(2) & \dots & i & \sigma_1(\sigma_1^{-1}(i+1)-1) & \dots & \sigma_1(\sigma_1^{-1}(i)+1) & i+1 & \dots & \sigma_1(N) \end{pmatrix}.$$

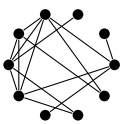
As one can see on Figure 2.10, this dynamic is local according to our definition (everything outside of $B_{G_1}^2(i)$ is unaltered), yet a global update of the permutation is required.

3.2 Phenotypes

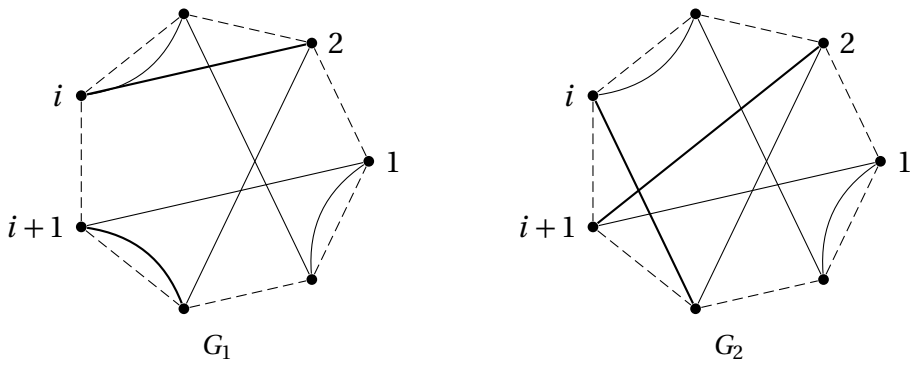
The phenotype of a dynamic graph is the set of global properties of the dynamic that one can quantify. It results from the choice of the genotype, and the later can often be retro-engineered to achieve certain global properties.

Stationary Distribution

The most common question when studying a Markov chain certainly consists in determining if it converges to a stationary distribution, and what that distribution might be. In every example from Section 2, the dynamic is *reversible*, which immediately answers the question.



CHAPTER 2. TAXONOMY OF DYNAMIC GRAPHS



$$\sigma_{G_1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 7 & 3 & 4 & 2 & 6 & 5 \end{pmatrix}$$

$$\sigma_{G_2} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 7 & 3 & 4 & 6 & 2 & 5 \end{pmatrix}$$

Permutations with $i = 4$

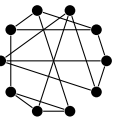
Figure 2.10 – Static Ring (dashed) with Cycle

	<i>Erdős-Rényi</i>	<i>Exponential Random Graphs</i>	<i>Pointer Models</i>	<i>Sliding Edges</i>	<i>Interchange Process</i>	<i>Static Structures</i>	<i>Dynamic Triangulations</i>
Fixed degrees				✓	✓		
Fixed edge count	(✓)	✓	✓	✓	✓	✓	✓
Static edges					✓		
Pointers		✓					
Multi-graph		(✓)	(✓)	(✓)			
Local transitions			✓	✓	✓	✓	
Sliding edges			✓				
Interchanging edges				✓			
Degree-dependent		✓					
Combinatorial					✓	✓	

Table 2.1 – Genotypes and Phenotypes from the Zoo

The most common application is with ergodic Markov chains for which we have $P_{i,j} = P_{j,i}$ for all pair of states i and j ; their stationary distribution is then necessarily the uniform distribution over the state space. This applies to most of our examples. For the others, checking that the stated distribution satisfies the detailed balance equation suffices.

The stationary distribution in itself is also often studied, though this falls somewhat outside of the scope of this survey. The following properties are often considered:



3. GENOTYPES AND PHENOTYPES

Degree Distribution Depending on the interpretation given to a dynamic, the desired degree distribution may vary. This includes things such as exponential distributions for online social networks and global communication networks, balanced distributions for local communication networks such as sensor networks, or even constant degrees, as with peer-to-peer networks.

Edge Count Distribution Most dynamics deal with sparse graphs, in which the number of edges, M , is quasi-linear in the number of vertices N : $M = O(N \log^k N)$ for some fixed integer k . Controlling this distribution is therefore essential when constructing a model.

Note that, in the case of dynamics with a fixed edge count, the question is dealt with from the start. This property is therefore mostly relevant for dynamics with a variable edge count.

Stationary Expansion Expansion is an important aspect of network models. It is a useful indicator of how well information travels in graphs, and is essential in studying epidemic propagation, rumor spreading, network capacity and resiliency, and more.

Define the expansion ratio of a strict subset S of V as

$$\phi(S) = \frac{|E(S, \bar{S})|}{\sum_{x \in S} \deg x}, \quad (2.1)$$

where $E(S, \bar{S})$ is the set of edges between S and its complementary. The expansion ratio of the graph as a whole is then defined as

$$\Phi = \min_{\substack{S \subset V \\ 1 \leq |S| \leq \frac{N}{2}}} \phi(S). \quad (2.2)$$

The higher the expansion ratio of the graph, the faster the simple random walk on that graph converges to its stationary distribution. For a rapid convergence rate, it is generally required that Φ be at least of order $\Omega(\log^{-k} N)$ for some fixed integer k , as opposed to a rate of order $O(N^{-\epsilon})$, which yields a slow convergence rate.

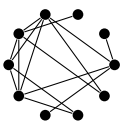
For stationary distributions such as the Erdős-Rényi model or uniform Δ -regular graphs, the random graph is an expander with high probability.

Mixing Time

Whereas the stationary distribution studies the asymptotic behavior of the dynamic, another point of interest is the convergence itself. Understanding the transitive phase can yield important insight into how dynamics operate, and can help engineer them to better suit ones needs.

One of the most common features of this transitive phase is the *mixing time*. Recall that the total variation distance between two distributions ρ and ν over the state space S is defined as

$$\|\rho - \nu\|_{\text{TV}} = \frac{1}{2} \sum_{x \in S} |\rho(x) - \nu(x)|,$$



and let π_x^t be the distribution of the dynamic at time t , starting in x at time 0. Denoting

$$d(t) = \max_{x \in S} \|\pi_x^t - \pi\|_{\text{TV}}$$

the distance to stationarity, the mixing time of the Markov chain is defined as

$$t_{\text{mix}} = \min \left\{ t > 0 \mid d(t) \leq \frac{1}{4} \right\}.$$

t_{mix} is the first instant at which, regardless of the initial distribution, the current state is within a predefined distance of the stationary distribution (here $\frac{1}{4}$, but any $\epsilon < \frac{1}{2}$ would serve).

4 Case Analysis

In this section, we analyze the mixing time of some of the examples from Section 2. We say a dynamic converges *rapidly* if its mixing time is of order at most $O(N \log^k N)$ for some integer k , and that it mixes *slowly* if its mixing time is of order at least $\Omega(N^2)$.

Note that rapid convergence implies at most $O(\log^k N)$ updates per vertex; in vertex-driven dynamics, this can be interpreted having a local complexity that scales well with the network.

4.1 Erdős-Rényi dynamic

In the sparse setting, the two models of Erdős-Rényi dynamics behave very differently.

Theorem 5. *Let $M = N \log N$ and $p = \frac{2 \log N}{N-1}$, such that the expected number of edges in the $\mathcal{G}(N, p)$ model is M .*

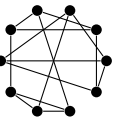
1. *The Erdős-Rényi dynamic for $\mathcal{G}(N, p)$ converges slowly, i.e. in at least $\Omega(N^2)$ steps.*
2. *The Erdős-Rényi dynamic for $\mathcal{G}(N, M)$ converges rapidly, reaching equilibrium in at most $O(N \log^2 N)$ steps.*

Note that with these values, the resulting graph is connected with high probability. We now prove both results separately.

Slow Mixing of $\mathcal{G}(N, p)$

We show that the $\mathcal{G}(N, p)$ dynamic converges slowly.

Proof. Observe that, given an initial configuration G with M edges, the probability that a graph G' distributed according to $\mathcal{G}(N, p)$ shares $\frac{M}{2}$ edges with G goes to 0 as N goes to infinity. As such, as long as $\frac{M}{2}$ edge of G remains untouched, the total variation distance is close to 1.



4. CASE ANALYSIS

More specifically, denoting X the number of edges that G and G' have in common, we have that $\mathbf{E}[X] = Mp = 2\frac{N}{N-1} \log^2 N$. Using Markov's inequality, this gives that

$$\mathbf{P}\left(X \geq \frac{M}{2}\right) \leq 4 \frac{\log N}{N-1} \xrightarrow{N \rightarrow +\infty} 0.$$

This implies that the mixing time is at least the time required to modify half of the initial edges. Since a modification to any of these edges happens on average once every

$$\frac{\binom{N}{2}}{M} = \frac{N-1}{2 \log N}$$

transitions, it takes at least

$$\frac{M}{2} \frac{N-1}{2 \log N} = \frac{N(N-1)}{4}$$

steps before this requirement is met. The mixing time is therefore at least of order $\Omega(N^2)$. \square

Rapid Mixing of $\mathcal{G}(N, M)$

We now show that, unlike $\mathcal{G}(N, p)$, $\mathcal{G}(N, M)$ converges rapidly to its stationary distribution.

Proof. The proof is very close to the previous one: we show that, once every edge has been touched at least once, we have reached the stationary distribution. Note that it is far quicker to modify all edges in this context than in the previous one.

Recall that, at each iteration, a pair of vertices (x, y) is chosen uniformly at random, alongside an independent edge uniformly distributed over the edges of the current configuration. If (x, y) was not already an edge, we then remove e , and replace it with (x, y) .

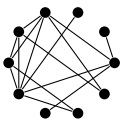
Suppose that, as you do this, you also tag edges as follows: if (x, y) was not previously an edge, it is now tagged, regardless of whether e was tagged or not; if (x, y) was already an edge, then

- if neither e nor (x, y) was tagged, tag (x, y) but not e (if $e = (x, y)$, tag it);
- if e was tagged, but not (x, y) , untag e and tag (x, y) ;
- otherwise do nothing.

All of the edges of the starting configuration are initially untagged.

We show by induction that, at all times, the set \mathcal{T} of tagged edges is distributed according to $\mathcal{G}(N, |\mathcal{T}|)$. This is clearly true at initialization. Two cases can arise at each iteration:

- e was already tagged. If it was left unchanged ((x, y) was a tagged edge), the distribution is unchanged, i.e. uniform. If it was replaced (untagged or removed), the new tagged edge is uniformly distributed over the edges not in \mathcal{T} , and the new distribution is once more uniform. Notice that the fact that (x, y) was or wasn't already an edge makes no difference.



CHAPTER 2. TAXONOMY OF DYNAMIC GRAPHS

- e was untagged. If (x, y) was a tagged edge, then \mathcal{T} and its distribution remain unchanged. Otherwise, (x, y) is tagged. Since it is uniformly distributed over the edges not in \mathcal{T} , the new set of edges is necessarily distributed according to $\mathcal{G}(N, |\mathcal{T}| + 1)$.

All that is left to show is that the edges are rapidly all tagged. For all i , denote $T_i = \min\{t \geq 0 \mid |\mathcal{T}_t| \geq i\}$ the time required before \mathcal{T} is of size i , and $\delta T_i = T_{i+1} - T_i$ the duration for which \mathcal{T} is of size exactly i . We are interested in bounding $T_M = \sum_i \delta T_i$.

If \mathcal{T}_t is of size i , then the probability of its size increasing at time $t + 1$ is that of selecting an untagged e and choosing (x, y) not in \mathcal{T} :

$$\mathbf{P}(\mathcal{T}_{t+1} = \mathcal{T}_t + 1) = \frac{M - i}{M} \cdot \frac{\binom{N}{2} - i}{\binom{N}{2}}.$$

δT_i is therefore geometrically distributed with the above parameter, and its average is

$$\mathbf{E}[\delta T_i] = \frac{M}{M - i} \cdot \frac{\binom{N}{2}}{\binom{N}{2} - i}.$$

Summing yields that

$$\begin{aligned} \mathbf{E}[T_M] &= \sum_{i=0}^{M-1} \mathbf{E}[\delta T_i] \\ &= \sum_{i=0}^{M-1} \frac{M}{M - i} \cdot \frac{\binom{N}{2}}{\binom{N}{2} - i} \\ &\leq \frac{\binom{N}{2}}{\binom{N}{2} - M} \cdot M \sum_{i=0}^{M-1} \frac{1}{M - i} \\ &= \frac{\binom{N}{2}}{\binom{N}{2} - M} \cdot M \sum_{i=1}^M \frac{1}{i} \\ &\sim \frac{\binom{N}{2}}{\binom{N}{2} - M} \cdot M \log M \\ &\sim \frac{1}{1 - \frac{2 \log N}{N-1}} \cdot N \log N \cdot \log(N \log N) \quad (M = N \log N) \\ &\sim N \log^2 N. \end{aligned}$$

This result could easily be foreseen, in that it is a modified instance of the coupon collector problem over the M edges, which would require $O(M \log M)$ iterations to complete.

T_M is what is known as a *strong stationary time*, a stopping time such that G_{T_M} is uniformly distributed, independently of T_M . We make use of the following property:

Property 9 (see [36], Proposition 6.10). *If T is a strong stationary time, then $d(t) \leq \max \mathbf{P}(T > t)$, where the max is taken over all initial states.*



Applying this property here yields that

$$\begin{aligned} d(t) &\leq \max \mathbf{P}(T_M > t) \\ &\leq \frac{N \log^2 N}{t} + o(1). \end{aligned} \quad (\text{Markov's inequality})$$

From this, we can deduce that

$$d(4N \log^2 N) \leq \frac{1}{4} + o(1),$$

and therefore that $t_{\text{mix}} = O(N \log^2 N)$. \square

The difference in convergence rate of these very similar models serves to outline the fact that there is no simple recipe to inferring the mixing time of given dynamics.

4.2 The Interchange Process

We now consider the interchange process. As mentioned in the previous section, it is best to consider half-edges when analyzing such a dynamic. For M initial edges, there are always $2M$ half-edges present in the graph, or ΔN for Δ -regular graphs.

Mixing Time of a Non-Local Dynamic

We first consider the non-local variant of the dynamic presented in Section 2. Recall that at each iteration, two half-edges are chosen uniformly and independently at random, and swapped. Formally, suppose the selected half edges correspond to the edges (x, y) and (x', y') , attached to vertices y and y' respectively. Replace these edges with (x, y') and (x', y) . Multi-edges and self-loops are considered acceptable and greatly simplify considerations.

To establish an upper bound on the mixing time of this dynamic, number the edges e_1, \dots, e_M , and denote e_i^\bullet and e_i° the two half-edges of edge e_i . Now consider the $2M$ -dimensional vector

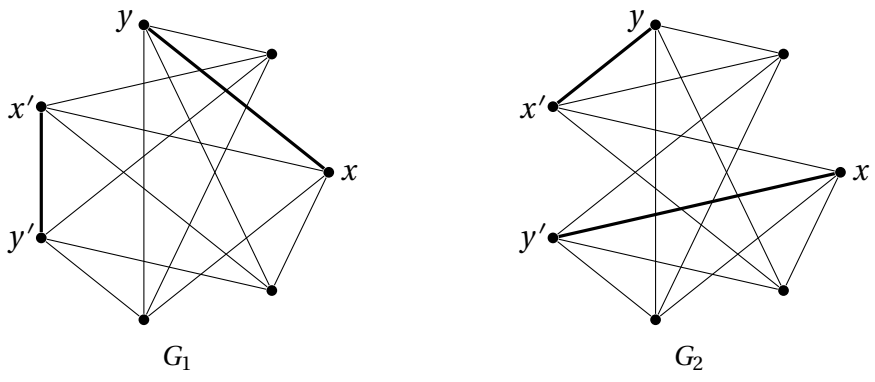
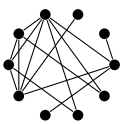
$$X = (e_1^\bullet, e_1^\circ, e_2^\bullet, e_2^\circ, \dots, e_M^\bullet, e_M^\circ).$$

The graph can be reconstructed from X alone by grouping the half-edges two by two to form edges. Furthermore, swapping two half-edges in the graph is reflected in X as a simple transposition. As such, the mixing time of the dynamic is upper-bounded by the mixing time of the random transposition process on X .

Property 10 (see [36], Corollary 8.10). *The mixing time of the random transposition process for permutations of size N is of order $O(N \log N)$.*

The proof of this property is very similar to that used for the $\mathcal{G}(N, M)$ model: tagging some of the coordinates of X as they are transposed allows us to keep track of a growing subset of coordinates whose values are uniformly distributed among themselves. A coupon collector argument then concludes the proof.

For the non-local interchange process, this implies that $t_{\text{mix}} = O(M \log(M))$, and namely $O(\Delta N \log(N))$ in the case of Δ -regular graphs.



$$P_{G_1, G_2} = \frac{1}{2M^2}$$

$$P_{G_2, G_1} = \frac{1}{2M^2}$$

Transition probabilities

Figure 2.11 – The 4-Regular Non-Local Interchange Process

Mixing Time of the Local Dynamic

Recall that the local dynamic differs from the previous one in that the half-edge swap occurs between two vertices y and y' selected conditioned on (y, y') being an edge. Bounding the mixing time in this scenario turns out to be far more difficult. To the extent of our knowledge, the best know result in this sense is the following

Theorem 6 (Allen-Zhu et al. [3]). *If $\Delta > 100 \log N$, then after $\Theta(\Delta^2 N^2 \sqrt{\log N})$ iterations of the local interchange dynamic, the resulting graph is an expander with high probability, namely*

$$\exists \gamma > 0, \quad \mathbf{P}(\Phi > \gamma) \geq 1 - N^{-2}.$$

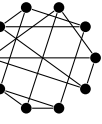
This result does not directly yield the mixing time; once the graph is an expander, however, the Markov chain is conceivably close to having mixed. As such, this suggests an upper bound on the convergence rate of no more than $O(\Delta^2 N^2 \sqrt{\log N})$, i.e. $O(N^2 \log^{\frac{5}{2}} N)$ if $\Delta = \Theta(\log N)$.

Note that this last bound is of order $o(N^3)$. In Section 4.3, we show that the mixing time of the interchange process on static graphs is at least $\Omega(N^3)$ in the worst cases. As such, though the above result still gives a somewhat slow upper bound, it does prove that the local dynamic does fundamentally better than the very similar interchange process.

4.3 Pointers on Rings

We finish with a last model, which mixes pointers and a static ring. While its construction may seem somewhat artificial, it gives some useful insight as to some necessary conditions in order to obtain rapidly mixing local dynamics.

The basic model is the following: let $V = \{1, \dots, N\}$ and $K \geq 1$ be a fixed integer. The edge set E consists of a static ring $R = \{(i, i + 1) \mid i \in V\}$, with $N + 1 \equiv$



1, and KN pointers, such that each vertex is the source and destination of exactly K pointers. Considering these pointers to be undirected edges, the resulting multigraph is $2(K+1)$ -regular.

All the dynamics are interchange processes: choose two pointers, and swap their destinations. This preserves the number of source and destination half-edges at each iteration. The difference results from the joint distribution of the two selected pointers:

Non-local dynamic The pointers are independent and uniformly distributed.

Ring dynamic The pointers are uniformly distributed, conditioned on having adjacent destinations in R .

Bootstrap dynamic The pointers are uniformly distributed, conditioned on having adjacent destinations in E .

Once more, multi-edges and self-loops may occur, but this is not a major concern, and simplifies proofs.

Theorem 7. *Within this framework, one has the following results:*

1. *The non-local dynamic converges in $O(KN \log(N))$.*
2. *The ring dynamic converges in $\Omega(KN^3)$, and, for $K = 1$, in $O(N^3 \log N)$.*

A conjecture concerning the mixing time of the bootstrap dynamic is given at the end of this section.

Non-Local Dynamic

Bounding the mixing time of the non-local dynamic is straightforward with an approach identical to that used for the interchange process over Δ -regular graphs

Proof. Consider the vector

$$X = (p_1^1, \dots, p_1^K, p_2^1, \dots, p_n^K)$$

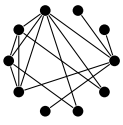
containing the destinations p_i^j of each pointer (for p_i^j , the j -th pointer with source i). An iteration is a transposition of two coordinates in that vector, and the previous results therefore tell us that $t_{\text{mix}} = O(KN \log(N))$. \square

Ring Dynamic

We now constrain our dynamic, requiring that swapped pointers have their destinations be adjacent along the ring. Intuitively, this should notably increase its mixing time. We show that this is indeed the case, and that $t_{\text{mix}} = \Omega(KN^3)$.

Proof. Denote (x_t) the destination of the first pointer of vertex 1. (x_t) is a lazy random walk over the ring, the probability of it staying in place is $p = 1 - \frac{2}{KN}$, and the probabilities of moving to the previous (-1) or next ($+1$) vertices are both $\frac{1}{KN}$. Call $\|x - y\|_R$ the distance between x and y along the ring:

$$\|x - y\|_R = \min(|x - y|, N - |x - y|).$$



CHAPTER 2. TAXONOMY OF DYNAMIC GRAPHS

Consider the random walk y_t over \mathbb{Z} such that $y_0 = x_0$ and the transitions $(+1, 0, \text{ or } -1)$ are the same for both chains. Notice that $y_t = x_t \pmod N$; this implies that $\|x_t - x_0\|_R \leq |y_t - y_0|$, and therefore that

$$\mathbf{P}\left(\|x_t - x_0\|_R \geq \frac{N}{4}\right) \leq \mathbf{P}\left(|y_t - y_0| \geq \frac{N}{4}\right).$$

Since $y_t - y_0$ is the sum of t i.i.d random variables with mean 0 and variance $\frac{2}{KN}$, its mean is 0 and its variance $t \frac{2}{KN}$. Chebyshev's inequality gives that

$$\mathbf{P}\left(|y_t - y_0| \geq \frac{N}{4}\right) \leq t \frac{2}{KN} \frac{16}{N^2},$$

hence

$$\mathbf{P}\left(\|x_t - x_0\|_R \geq \frac{N}{4}\right) \leq t \frac{32}{K} N^{-3}.$$

Under the stationary distribution, x is uniformly distributed, and therefore $\mathbf{P}\left(\|x - x_0\|_R \geq \frac{N}{4}\right) = \frac{1}{2}$. The mixing time must therefore satisfy

$$\mathbf{P}\left(\|x_{t_{\text{mix}}} - x_0\|_R \geq \frac{N}{4}\right) \geq \frac{1}{4},$$

as below this threshold,

$$d(t) = \max_{x_0} \|\mathbf{P}_{x_0}(x_t = \cdot) - \pi\|_{\text{TV}} \geq \frac{1}{4}.$$

Combining the above inequalities, we obtain that $t_{\text{mix}} \geq \frac{K}{128} N^3$. \square

We furthermore show that this bound is close to tight, as the mixing time of the ring-based shortcut dynamic with $K = 1$ is of order at most $N^3 \log^2 N$. For this, we use the following result from [6]:

Property 11. (see [6], Theorem 4.6, p. 47) For every pair of vertices x and y , denote $\Gamma_{x,y}$ a path from x to y . Let

$$\Lambda = \max_{x,y} |\Gamma_{x,y}|,$$

where $|\gamma|$ designates the length of path Γ , and

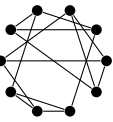
$$B = \max_{e \in E} |\{(x, y) \in V^2 : e \in \Gamma_{x,y}\}|.$$

Then the mixing time for the interchange of labels between adjacent vertices is of order at most $O(\Lambda B \log N)$.

Taking the $\Gamma_{x,y}$ to be the shortest paths between x and y along the cycle, we immediately have that $\Lambda = \lfloor \frac{N}{2} \rfloor$. Furthermore, the number of paths of length l crossing a single edge e is at most $2l$, hence

$$B \leq \sum_{l=1}^{\lfloor \frac{N}{2} \rfloor} 2l \approx N^2.$$

This concludes the proof.



4. CASE ANALYSIS

Bootstrap Dynamic

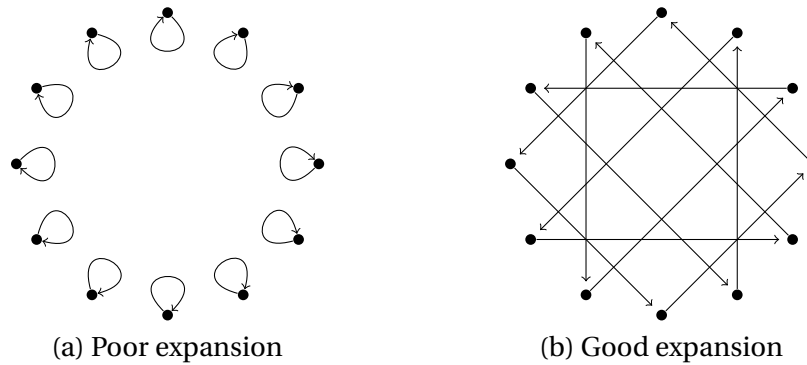
We finally introduce a local dynamic with a rapid empirical mixing time. Consider the non-local dynamic, and constrain it so that the destinations of the swapped pointers are adjacent in the graph, connected by any pointer or ring edge, other than the two pointers themselves (so as to preserve reversibility).

Conjecture 1. *The mixing time of the bootstrap dynamic is of order $O(KN \log^k N)$ for some fixed k .*

Chapter 3 is devoted to proving an analogous result for a similar continuous-time model.

Discussion

We give a brief overview as to why this result is somewhat surprising, and what makes it feasible.



$$\Phi_{\text{poor}} = \frac{1}{N}$$

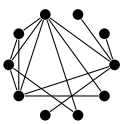
$$\Phi_{\text{good}} = \Omega(\log^{-k} N)$$

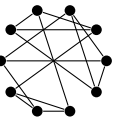
Expansion ratio

Figure 2.12 – Expansion of Pointers on Rings

Notice that the worst initial conditions are those such as that presented in Figure 2.12a. For these, the initial rate of convergence is at least cubic, since it closely mirrors that of the ring dynamic. However, the limit distribution gives good expanders w.h.p. (namely $\Phi = \Omega(\log^{-k} N)$ for some k), and the rate of convergence should therefore improve as the dynamic proceeds. Since the rate of convergence of the interchange process over expanders has convergence rate $O(N \log N)$, one can easily conjecture that the rate of convergence starting from an expander is of order $O(KN \log N)$, i.e. the same as for the non-local dynamic.

The convergence rate of the bootstrap dynamic starting from an arbitrary position is therefore somewhere between $\Theta(KN^3 \log N)$ and $\Theta(KN \log N)$. Our conjecture, backed by Theorem 9, is that local bootstrap dynamics are nearly as efficient as non-local dynamics.





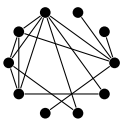
Chapter 3

Rapid Mixing of Dynamic Graphs with Local Evolution Rules

1	Introduction	46
2	Main results	47
3	Proof strategy	49
4	From bounds in expectation to bounds with high probability	52
4.1	Proof of Lemma 1	52
4.2	Proof of expansion at equilibrium	54
5	Controlling the mean	56
5.1	Small sets: from partial expansion to partial spread . . .	57
5.2	Large sets	57
6	Proof of Theorem 9	59

A brief announcement containing the main results of this chapter was presented at DISC 2017.

This chapter was recently submitted to *Random Structures & Algorithms*.



1 Introduction

Distributed systems typically cannot operate efficiently unless their constituting parts are interconnected via a network with suitable properties. In the context of peer-to-peer systems, desirable properties of the interconnection graph between peers include having a small diameter, small node degrees, and requiring many failures to disconnect a sizeable part of the network. Yet another useful property is the ability to obtain at low cost and via a distributed algorithm uniform samples of nodes in the system.

As mentioned in the first chapter, all these properties hold if the interconnection graph is an expander. By definition, the unoriented graph $G = (V, E)$ is a γ -expander for some constant $\gamma > 0$ if each set $S \subset V$ of vertices with size $|S|$ no larger than $\frac{1}{2}|V|$ is such that at least $\gamma|S|$ distinct edges in E connect S to its complement \bar{S} in V . It is an expander if it is a γ -expander for some $\gamma \geq \Omega(1)$.

For instance, when the graph is an expander, close-to-uniform samples are obtained in a distributed manner and at low cost by running a random walk on the peer-to-peer graph: It being an expander then ensures that only few steps (on the order of the logarithm of the system size, measured in number of nodes) of the walk suffice.

Because peer-to-peer systems are volatile, i.e. subject to node arrivals and departures, it is not possible to determine once and for all an expander graph to interconnect participating peers. Instead the graph must constantly evolve, with the aim to preserve or restore the desired expander property. Moreover, the graph evolution must rely on local adjustments, since by design no central controller has knowledge of the whole graph.

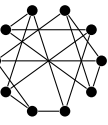
This has prompted research on dynamics for continuous modification of graphs that would (i) rely only on local modifications of the current graph, (ii) produce expander graphs at equilibrium, and (iii) reach equilibrium quickly. More precisely, considering graphs on a set of N nodes, by quickly we mean requiring a number of modifications per node that scales poly-logarithmically in N before equilibrium is attained.

The main contribution of this chapter is to propose a new graph dynamics together with the proof that it meets these three requirements. Its organization is as follows. Section 2 describes our proposed dynamics together with the main result, Theorem 8. It also highlights Theorem 9, our technical result controlling the spread of laws of random walks on graphs at short times. Section 3 explains the proof strategy. Section 4 explains how to deduce bounds with high probability on isoperimetric ratios from bounds on corresponding expectations, using negative dependence properties. Section 5 derives the necessary bounds on expectations, leveraging in particular Theorem 9. The proof of the later constitutes Section 6.

We now review relevant prior work.

Related Work

Markovian local graph dynamics for peer-to-peer systems have been considered in [41, 11, 10, 17]. In all these papers the stationary regime for the proposed dynamics has been identified; in the last three references, loose bounds on the mixing time (defined below), or time to achieve equilibrium, have been obtained.



The tightest bounds available prior to the present article on the time before such local graph dynamics produce an expander graph were obtained in [3]. Specifically, it is shown in [3] that a discrete time Markov chain on the set of connected Δ -regular graphs on N vertices creates with high probability an expander graph after a time of $O(N^2 \Delta^2 \sqrt{\log N})$ with Δ of order $O(\log N)$. This implies the realization of this property after each node has performed a number of updates of order $O(N \log^{\frac{5}{2}} N)$, i.e. a number that is quasi-linear in the system size N .

Graph dynamics have also been considered in different contexts. [47] considers local dynamics for producing so-called cladograms uniformly at random, and bounds their mixing time. [30] considers dynamics of matchings in bipartite graphs and controls their mixing time. The motivation of [30] is the estimation of graph descriptors using a Markov chain Monte-Carlo approach. Finally, non-local graph dynamics together with their mixing time have been considered in [7] in order to sample from so-called exponential random graph distributions.

2 Main results

In the sequel, we consider graphs over vertex set $[N]$, asymptotic results being with respect to N .

Consider the following setting. The vertices in $[N]$ are connected by edges of three distinct types: a fixed cycle, blue edges and red edges. The cycle is constituted of a fixed set of edges $E_o = \{(n, n + 1) \mid i \in [N]\}$, with $N + 1 \equiv 1$. Each node $n \in [N]$ furthermore maintains two pointers, one blue and one red, with respective destinations b_n, r_n in $[N]$. The destinations of the pointers are such that $(b_n)_{n \in [N]}$ and $(r_n)_{n \in [N]}$ are permutations: each node n is the destination of exactly one blue pointer and one red pointer. The blue edges and red edge sets are respectively $E_b = \{(n, b_n) \mid n \in [N]\}$ and $E_r = \{(n, r_n) \mid n \in [N]\}$. All edges are considered to be unoriented.

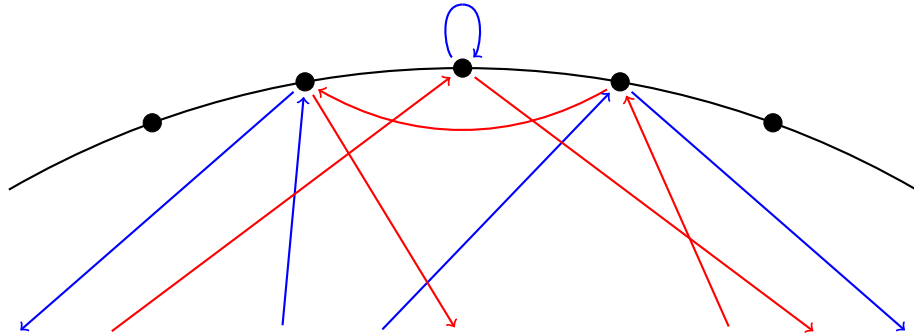


Figure 3.1 – Bi-color pointer interchange model

We now consider the following continuous-time dynamics. The graph evolves through alternating blue and red phases. During each phase, only the edges of a given color evolve, while those of the other color are kept fixed. During a blue phase, for example, blue pointers are swapped along graph G_r constituted of the edges in both E_o and E_r . Note that G_r is a 4-regular multigraph.

The dynamics for a blue phase are defined as follows. Each edge $e = (i, j)$ of G_r maintains an internal clock, in which the time between ticks are exponentially

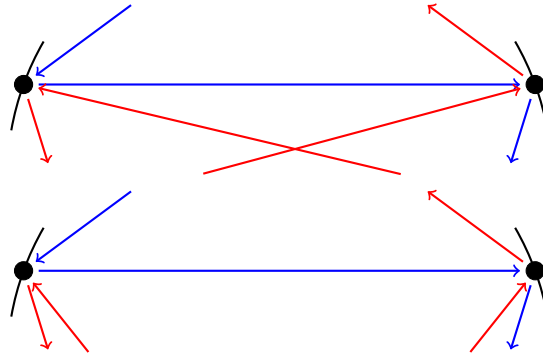
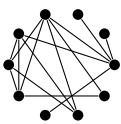


Figure 3.2 – Swapping two red pointers along a blue edge (red phase)

distributed with mean 1 and all independent. At every tick, the two nodes $n, m \in [N]$ such that $b_n = i$ and $b_m = j$ swap their pointers. This effectively boils down to transposing i and j in the permutation $(b_n)_{n \in [N]}$. Such a process has been studied in the literature, where it is known as the *interchange process*. See for instance Jonasson [33] or N. Berestycki [6], where the discrete time version of this process is analyzed.

For the red phases, the roles of blue and red pointers are swapped; the graph containing the edges in E_o and E_b is denoted G_b . To clarify notations, we denote E_f the fixed edge set during phase f , i.e. E_b if f is a red phase, E_r otherwise, and G_f the fixed graph, i.e. containing the edges in E_o and E_f . We finally write $E_f^\circ = E_f \cup E_o$.

Our main result is then as follows

Theorem 8. *Let $T = \log^a N$ where $a > 8$ is a constant. After $F = \log_2 N$ phases of length T , with high probability, the sets of blue and red pointers are uniformly and independently distributed permutations over the symmetric group \mathfrak{S}_N .*

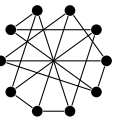
Corollary 2. *After each node has undergone a number of local connectivity modifications that is polylogarithmic in N , the process has produced an expander with high probability.*

Proof. By time $\tau = FT$, a given node $n \in [N]$ has seen under these dynamics a number of connectivity modifications that is at most a Poisson random variable with mean 8τ . Indeed, at any given time, the rate at which a transition might occur is at most 8, corresponding to the rate at which the pointer b_n (for the blue phase) issued from n moves (equal to the number of edges of G_r adjacent to b_n , i.e. 4) plus the rate at which the blue pointer ending at n moves (also 4).

The probability that this number M_n of connectivity modifications exceeds 16τ is then, by Chernoff's bound for deviations of Poisson random variables from their mean, bounded by

$$\mathbf{P}(M_n \geq 16\tau) \leq e^{-8\tau h(\frac{16\tau}{8\tau})} = e^{-8\tau h(2)},$$

where $h(x) = x \log x - x + 1$ is the Cramér transform of a unit mean Poisson random variable. Since τ is at least of order $\log^{a+1} N$ with $a > 0$, the last term is $o(N^{-1})$. Thus the probability that at least one node $n \in [N]$ undergoes more than $16\tau = \text{polylog}(N)$ local modifications by time τ is, by the union bound, no more than $N \cdot o(N^{-1}) = o(1)$.



3. PROOF STRATEGY

The fact that the resulting graph G_F is an expander will be shown in Section 4, in which we introduce the necessary technical lemmas. Note that results in [8, 18] establish for very similar (although not identical) random graph models that these are expanders with high probability. \square

We now state another result, which will be instrumental in Section 5, and proven in Section 6, but which we believe could be of independent interest:

Theorem 9. *Let $G = ([N], E)$ be an undirected multi-graph with maximum degree Δ , and $(X_t)_{t \in \mathbb{R}}$ the continuous time random walk on G , i.e. the Markov jump process on $[N]$ with jump rates q_{ij} equal to the multiplicity of (i, j) in E . The infinitesimal generator of $(X_t)_{t \in \mathbb{R}}$ is $-L$, where L is the Laplacian matrix of G . Let $(\pi_i(t))_{i \in [N]}$ denote its law at time t . For an arbitrary initial distribution of the random walk, for any $k \leq \frac{N}{2}$ and $S \subset [N]$ such that $|S| \leq k$ and any $t \geq 0$, one has:*

$$\sum_{i \in S} \pi_i(t) \leq \frac{|S|}{k+1} + \sqrt{k+1} e^{-\lambda_2^* t}, \quad (3.1)$$

where

$$\lambda_2^* = \frac{\Phi_k(G)^2}{2\Delta}. \quad (3.2)$$

Remark 1. *The quantity λ_2^* is of the same form as the lower bound on the spectral gap λ_2 of the Laplacian that the celebrated Cheeger inequality gives when $k = \frac{N}{2}$. In fact for $k = \frac{N}{2}$, the expression of $\Phi_k(G)$ coincides with this lower bound. In this classical situation, instead of (3.1), one has the conclusion that*

$$\|\pi(t) - \mathcal{U}_{[N]}\|_{\text{TV}} \leq \frac{1}{2} \sqrt{N} e^{-\lambda_2^* t},$$

as shown in Property 5.

3 Proof strategy

To proceed, we first introduce some definitions.

Definition 20. *The collection $(\Phi_k(G))_{1 \leq k \leq \frac{N}{2}}$ of isoperimetric constants of graph G constitutes its isoperimetric profile.*

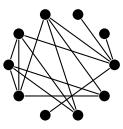
The graph is said to be a (γ, c) -expander if, for all $k \leq \frac{N}{2}$, $\Phi_k(G) \geq \min(\gamma, \frac{c}{k})$.

Note that a graph is a γ -expander according to the classical notion if it is a $(\gamma, \frac{N}{2})$ -expander according to the above definition.

Our proof consists in controlling the evolution of the isoperimetric profile of the graph along which pointers move from one phase to the next, establishing lower bounds on this profile in an iterative manner.

Let β be a constant such that $1 < \beta < \frac{a-4}{4}$. Such β exists by our assumption that $a > 8$. Let $\gamma = \log^{-\beta} N$. We show the following

Lemma 1. *Suppose that, during phase f , the graph G_f is a (γ, c) -expander for some integer c . Then, with probability at least $1 - o(N^{-1})$, the graph G_{f+1} is a $(\gamma, 2c)$ -expander.*



CHAPTER 3. RAPID MIXING OF LOCAL DYNAMIC GRAPHS

Notice that if G_f contained blue edges, G_{f+1} contains red edges, and vice versa.

To prove this, we first show a stronger lower bound on the average number of shortcuts leaving any given set S :

Lemma 2. *If, for a given phase f and integer c , G_f is a (γ, c) -expander, then for all $S \subset [N]$ with $|S| \leq \frac{N}{2}$,*

$$\mathbf{E}[|E_{f+1}(S, \bar{S})|] \geq \frac{1}{2\gamma} \min(\gamma|S|, 2c). \quad (3.3)$$

Lemma 1 is then deduced from Lemma 2 by invoking some concentration inequalities together with union bounds. Details are given in Section 4.2.

An easy consequence of Lemma 1 is the following:

Corollary 3. *After $F = \log_2(N)$ phases, with high probability both $G_{\mathbf{b}}$ and $G_{\mathbf{r}}$ are $(\gamma, \frac{N}{2})$ -expanders.*

Proof. Clearly, G_1 is a $(\gamma, 2)$ -expander. Indeed, any subset $S \subset [N]$ of size $|S| \leq \frac{N}{2}$ is connected by at least two edges (that come from the cycle) to its complement \bar{S} , so that

$$|E_1^\circ(S, \bar{S})| \geq 2 \geq \min(\gamma|S|, 2).$$

Denote by \mathcal{E}_f the event that G_f is a $(\gamma, 2^f)$ -expander. Thus we have just established that event \mathcal{E}_1 holds with certainty, and Lemma 1 entails that, for all $f \geq 1$,

$$\mathbf{P}(\bar{\mathcal{E}}_{f+1} \mid \mathcal{E}_f) \leq o(N^{-1}).$$

Thus

$$\begin{aligned} \mathbf{P}(\bar{\mathcal{E}}_{f+1}) &= \mathbf{P}(\bar{\mathcal{E}}_{f+1} \mid \mathcal{E}_f) \mathbf{P}(\mathcal{E}_f) + \mathbf{P}(\bar{\mathcal{E}}_{f+1} \mid \bar{\mathcal{E}}_f) \mathbf{P}(\bar{\mathcal{E}}_f) \\ &\leq o(N^{-1}) + \mathbf{P}(\bar{\mathcal{E}}_f). \end{aligned}$$

By induction on f , this yields

$$\mathbf{P}(\bar{\mathcal{E}}_{f+1}) \leq o\left(\frac{f}{N}\right).$$

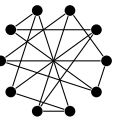
For $F = \log_2(N)$, the right-hand side of this expression is $o(1)$, so that with high probability, the graphs G_{F-1} and G_F are $(\gamma, \frac{N}{2})$ -expanders. As the color of these two graphs differ, this concludes the proof. \square

The proof of Theorem 8 is then concluded as follows:

Proof. By Corollary 3, after $F = \log_2(N)$ phases, the Cheeger constants $\Phi_{\frac{N}{2}}(G_{\mathbf{b}})$ and $\Phi_{\frac{N}{2}}(G_{\mathbf{r}})$ are at least γ .

We show that, for any graph G , $\Phi_{\frac{N}{2}}(G) \geq \gamma$ implies that the interchange processes on G mixes in less than T steps, so that with high probability, in two more phases our process will have reached stationarity.

Our main tool to this end is Theorem 4.3, p. 39 in Berestycki [6], which gives a sufficient condition for the discrete time interchange process on a graph G to mix in time T . The continuous time analogous result reads



3. PROOF STRATEGY

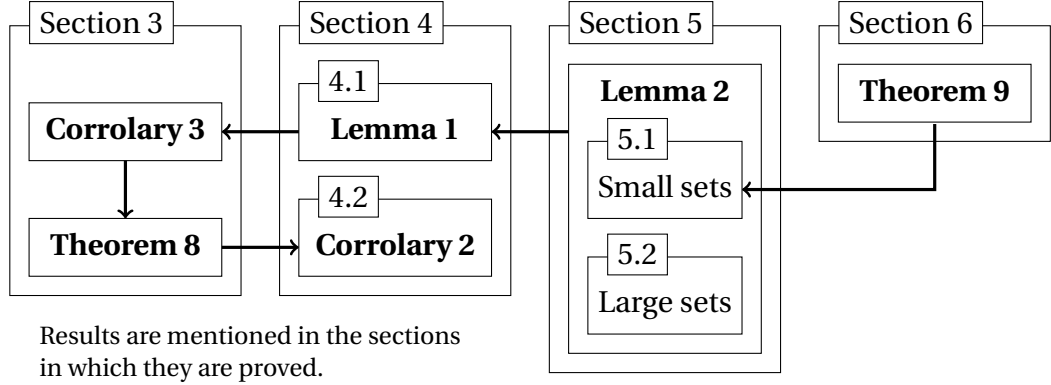


Figure 3.3 – Outline of proof

Theorem 10 (Theorem 4.3, p. 39, [6]). *For each pair of nodes $i, j \in [N]$, define a path γ_{ij} on G connecting i and j , and let $\Gamma = \{\gamma_{ij} \mid i, j \in [N]\}$. Denote Υ the length of the longest path in Γ , and K the supremum over the edges e of G of the number of paths in Γ passing through e .*

The continuous time interchange process on G will have mixed in time T provided

$$T \geq 8\Upsilon K \frac{\log N}{N}. \quad (3.4)$$

According to Lemma 3 below, for a Δ -regular $(\gamma, \frac{N}{2})$ -expander with Δ constant, we can choose these paths such that

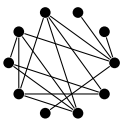
$$\Upsilon = O\left(\frac{\log N}{\gamma^2}\right) \quad \text{and} \quad K = O\left(\frac{N \log^2 N}{\gamma^2}\right).$$

Plugged into (3.4), these evaluations imply that mixing has occurred by time T provided T is large compared to $\left(\frac{\log N}{\gamma}\right)^4$, i.e provided $\log^a N = \omega(\log^{4+4\beta} N)$. This condition holds since $\beta < \frac{a-4}{4}$. \square

Lemma 3. *Let G be a Δ -regular graph with vertex set $[N]$, such that $\Phi_{\frac{N}{2}}(G) \geq \gamma$. One can construct a set of paths $\Gamma = \{\gamma_{ij} \mid i, j \in [N]\}$ such that the γ_{ij} each have length at most $\Upsilon = 2\frac{\Delta^2}{\gamma^2} \log N$, and such that each edge e of G is crossed by at most $18\frac{\Delta^2}{\gamma^2} N \log^2 N$ paths.*

Proof. The celebrated Cheeger's inequality (see e.g. Berestycki [6] Theorem 3.5, p. 30) ensures that the spectral gap for the discrete time random walk on a Δ -regular graph G with $\Phi_{\frac{N}{2}}(G) \geq \gamma$ is at least $\frac{\gamma^2}{2\Delta^2}$. Thus the total variation distance between the distribution of the random walk at time $\Upsilon = 2\frac{\Delta^2}{\gamma^2} \log N$ and the uniform distribution on G is $o(N^{-1})$ (this follows e.g. by Theorem 2.2, p. 18 in [6]). As a result, for any $i \in [N]$, the probability that the walk started at i hits j at time Υ is at least $\frac{1}{2N}$. Consider then the following randomized construction. For each i , create $5N \log N$ independent walks of length Υ started at i . The probability that for some particular $j \in [N]$, no such walk issued from i hits j is then at most

$$\left(1 - \frac{1}{2N}\right)^{5N \log N} \leq e^{-\frac{5}{2} \log N} = o(N^{-2}).$$



Using the union bound, we can therefore conclude that, with high probability, the collection of paths thus created joins every node i to every node j .

Let us now evaluate the number of times a given edge $e = (u, v)$ of G is traversed by this collection of paths. This is no larger than the number of times these paths visit node u . For $t \leq 5N \log N$, denote by $X_i(t)$ the number of visits to u by the t -th path sampled with starting point i . Clearly, $X_i(t) \leq \Upsilon$. Also,

$$\mathbf{E} \left[\sum_{i \in [N]} \sum_{t \leq 5N \log N} X_i(t) \right] = 5N \log N \sum_{i \in [N]} \sum_{l=0}^{\Upsilon} P_{iu}^{(l)},$$

where $P_{iu}^{(l)}$ denotes the transition probability from i to u in l steps of the walk. However the walk is symmetric, so that $P_{iu}^{(l)} = P_{ui}^{(l)}$. The above expectation thus reads

$$\mathbf{E} \left[\sum_{i \in [N]} \sum_{t \leq 5N \log N} X_i(t) \right] = 5(\Upsilon + 1)N \log N.$$

Let $Z = \sum_{i \in [N]} \sum_{t \leq 5N \log N} X_i(t)$ denote the total number of visits to u by all paths. For any $C > 0$, Hoeffding's inequality then gives

$$\mathbf{P}(Z \geq \mathbf{E}[Z] + CN\Upsilon \log N) \leq \exp\left(-\frac{C^2 N^2 \Upsilon^2 \log^2 N}{5\Upsilon^2 N^2 \log N}\right) = e^{-\frac{C^2}{5} \log N}.$$

Taking $C = 3$, the right-hand side is $o(N^{-1})$. Thus, by the union bound, with high probability, no node u is visited more than $9\Upsilon N \log N = 18 \frac{d^2}{\gamma^2} N \log^2 N$ times by the collection of constructed paths. \square

4 From bounds in expectation to bounds with high probability

4.1 Proof of Lemma 1

Assume that G_f is a (γ, c) -expander. By Lemma 2, for each fixed set $S \subset [N]$ with $|S| \leq \frac{N}{2}$, we have that

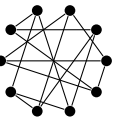
$$\mathbf{E}[|E_{f+1}(S, \bar{S})|] \geq \frac{1}{2\gamma} \min(\gamma|S|, 2c).$$

Fix $k \leq \frac{N}{2}$. We further restrict ourselves to $k > \frac{2}{\gamma}$, since $|E_o(S, \bar{S})| \geq 2$ and therefore one always has that $\Phi_k \geq \gamma$ for $k \leq \frac{2}{\gamma}$.

For some set S of size k , let $l \in \llbracket 1, k \rrbracket$ be the number of contiguous portions of the cycle it is made of. Clearly $|E_o(S, \bar{S})| = 2l$, and therefore

$$|E_{f+1}^\circ(S, \bar{S})| = |E_{f+1}(S, \bar{S})| + 2l.$$

Recall (see e.g. Dubashi and Ranjan [13], and Borcea et al. [9]) that a set of random variables $(X_i)_{i \in I}$ is said to be negatively associated if for any two functions f, g :



4. FROM BOUNDS IN EXPECTATION TO BOUNDS WITH HIGH PROBABILITY

$\mathbb{R}^I \rightarrow \mathbb{R}$ that are non-decreasing in each of their coordinates, and depend on disjoint sets of variables X_i , the two random variables $f((X_i)_{i \in I})$ and $g((X_i)_{i \in I})$ are negatively correlated, i.e.

$$\mathbf{E}[f((X_i)_{i \in I})g((X_i)_{i \in I})] \leq \mathbf{E}[f((X_i)_{i \in I})]\mathbf{E}[g((X_i)_{i \in I})].$$

We will need the following two results.

Lemma 4. *Conditionally on the shortcut configuration at the beginning of the considered phase, the random variable $|E_{f+1}(S, \bar{S})|$ consists of the sum of negatively associated Bernoulli random variables. Consequently, for any $r \in (0, 1)$, it holds that*

$$\mathbf{P}\left(|E_{f+1}(S, \bar{S})| \leq r \frac{1}{2\gamma} \min(\gamma|S|, 2c)\right) \leq e^{-\frac{1}{2\gamma} \min(\gamma|S|, 2c)h(r)}, \quad (3.5)$$

where $h(r) = r \log r - r + 1$.

Proof. Represent the collection of termination points of pointers through the binary variables $\xi_i \in \{0, 1\}$, $i \in [N]$ where $\xi_i = 1$ if and only if one pointer issued from S points towards i . The variables $(\xi_i)_{i \in [N]}$ evolve, under the interchange process dynamics, as a symmetric exclusion process. Given that we condition on the initial configuration, applying Proposition 5.1 in [9] gives that the collection of indicator variables $(\xi_i(t))_{i \in [N]}$ satisfies the *strong Rayleigh property* [37] for all $t \geq 0$. This is namely true for $(\xi_i(T))_{i \in [N]}$, i.e. at the end of the phase. Since the strong Rayleigh property implies negative association (see again [9], Section 2, Figure 1), it then follows from Dubhashi and Ranjan [13] that

$$|E_{f+1}(S, \bar{S})| = \sum_{i \in \bar{S}} \xi_i(T)$$

verifies the same Chernoff bounds that it would if the $\xi_i(T)$ were mutually independent. The announced result (3.5) then follows from classical evaluations of Chernoff bounds. \square

Lemma 5. *The number of sets $S \subset [N]$ of size k that consist of l contiguous portions of the cycle is at most N^{2l} .*

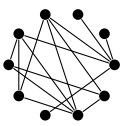
It is also upper-bounded by $N \binom{k-1}{l-1} \binom{N-k-1}{l-1}$.

Proof. We may enumerate such sets S by scanning the cycle $[N]$ starting from 1, and identifying the first time we find a starting point of an interval in S , then the end point of that interval, and so on. Clearly this will produce $2l$ numbers in $\llbracket 1, N \rrbracket$, which characterize S , hence the upper bound N^{2l} .

To obtain the other upper bound, note that the number of strictly positive sequences of l integers x_1, \dots, x_l such that $x_1 + \dots + x_l = k$ equals the number of non-negative sequences of such integers such that $x_1 + \dots + x_l = k - l$, and this number is well known to equal

$$\binom{k-l+l-1}{l-1}.$$

Similarly, the number of strictly positive sequences of integers y_1, \dots, y_l such that $y_1 + \dots + y_l = N - k$ equals $\binom{N-k-1}{l-1}$. Given a set $S \subset [N]$ of size k and made of l



distinct intervals, let $z \in \llbracket 0, N-1 \rrbracket$ be the smallest number of clockwise rotations of the set such that $i = 1$ corresponds exactly to the beginning (in clockwise order) of a contiguous segment of the set. The set is then fully specified by the lengths of its constituting segments, x_1, \dots, x_l , in clockwise order, together with the lengths of the segments separating its own segments, y_1, \dots, y_l . Since this construction forms an injection from the collection of considered sets S to a set of size $N \binom{k-1}{l-1} \binom{N-k-1}{l-1}$, the announced upper bound follows. \square

The union bound gives us, using the first upper bound in this last lemma, the following bound on the probability p_k that for some set S of size k , one does not have the desired property $|E_{f+1}^\circ(S, \bar{S})| \geq \min(\gamma k, 2c)$:

$$p_k \leq \sum_{l=1}^k N^{2l} \mathbf{P}(|E_{f+1}(S, \bar{S})| \leq \min(\gamma k, 2c) - 2l).$$

We now distinguish according to whether $\gamma k \leq 2c$ or not.

Case 1: $\gamma k \leq 2c$ We then have, by (3.5):

$$\begin{aligned} p_k &\leq \sum_{l=1}^{\frac{\gamma k}{2}} N^{2l} \exp\left(-\frac{1}{2\gamma} \gamma k h\left(\frac{\gamma k - 2l}{\frac{1}{2\gamma} \gamma k}\right)\right) \\ &\leq N \exp\left(\gamma k \log N - \frac{1}{2\gamma} \gamma k h(o(1))\right) \\ &= \exp\left(\left(1 + \gamma k - \frac{\gamma k}{2} \log^{\beta-1} N \cdot h(o(1))\right) \log N\right). \end{aligned}$$

The term in the inner parenthesis is asymptotically equivalent to $-\frac{\gamma k}{2} \log^{\beta-1} N$, because $h(o(1))$ tends to 1 and $\beta > 1$. Moreover, since $\gamma k > 2$, the whole exponent is large compared to $\log N$. Thus $p_k = o(N^{-r})$ for any fixed $r > 0$.

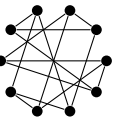
Case 2: $\gamma k > 2c$ We then have

$$\begin{aligned} p_k &\leq \sum_{l=1}^c N^{2l} \exp\left(-\frac{1}{2\gamma} 2c h\left(\frac{2c - 2l}{\frac{1}{2\gamma} 2c}\right)\right) \\ &\leq N \exp\left(2c \log N - \frac{1}{2\gamma} 2c h(o(1))\right) \\ &= \exp\left(\left(1 + 2c - c \log^{\beta-1} N \cdot h(o(1))\right) \log N\right). \end{aligned}$$

We can then conclude as in the previous case.

4.2 Proof of expansion at equilibrium

To complete the proof of Corollary 2, we now show that the graph G_F obtained after $F = \log_2 N$ phases is an expander, i.e. an ϵ -expander for some fixed $\epsilon \geq \Omega(1)$, thereby strengthening the statement that it is a γ -expander. The graph G_F is, with high probability made of a cycle plus a uniform random permutation. We place ourselves in the remainder of this section on the event where it consists precisely of a cycle plus a uniform random permutation.



4. FROM BOUNDS IN EXPECTATION TO BOUNDS WITH HIGH PROBABILITY

For a set $S \subset [N]$ of size k , the number of edges from the permutation starting at nodes in S and ending at nodes in \bar{S} reads $\sum_{i=1}^k \xi_i$, for Bernoulli random variables ξ_i with mean $1 - \frac{k}{N}$. Moreover, these random variables are negatively associated, as follows from [13]. Chernoff bound on their deviation from the mean is then stronger than the corresponding bound obtained assuming they are independent. This entails that, for $r \leq 1 - \frac{k}{N}$,

$$\mathbf{P}\left(\sum_{i=1}^k \xi_i \leq rk\right) \leq e^{-kD(r\|1-\frac{k}{N})},$$

where

$$D(r\|s) = r \log \frac{r}{s} + (1-r) \log \frac{1-r}{1-s}$$

is the Kullback-Leibler divergence between Bernoulli distributions with parameters r and s .

Fix $\epsilon > 0$ a small positive constant, and let $k \leq N^{\frac{1}{3}}$. In particular, one has that $\frac{k}{N} \leq 1 - \epsilon$. We thus have, in view of the first bound in Lemma 5, the upper-bound on the probability p_k that there exists some set $S \subset [N]$ of size k such that $|E_F^\circ(S, \bar{S})| < \epsilon k$:

$$\begin{aligned} p_k &\leq \sum_{l=1}^k N^{2l} \mathbf{P}(|E_f(S, \bar{S})| < \epsilon k - 2l) \\ &\leq \sum_{l=1}^{\lfloor \frac{\epsilon k}{2} \rfloor - 1} N^{2l} e^{-kD(\epsilon - \frac{2l}{k}\|1-\frac{k}{N})}. \end{aligned}$$

Since $D(\epsilon - \frac{2l}{k}\|1-\frac{k}{N})$ increases with l , we may upper-bound each term in this last summation by $N^{k\epsilon} \exp(-kD(\epsilon\|1-\frac{k}{N}))$. Its logarithm C reads

$$\begin{aligned} C &= \epsilon k \log N - kD\left(\epsilon\|1-\frac{k}{N}\right) \\ &= \epsilon k \log N - k\epsilon \log \frac{\epsilon}{1-\frac{k}{N}} - k(1-\epsilon) \log \frac{1-\epsilon}{\frac{k}{N}}. \end{aligned}$$

The second term is $O(k)$, while the third term is for large enough N no larger than $-k(1-\epsilon) \log \sqrt{N} = -k\frac{1-\epsilon}{2} \log N$. It follows that

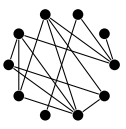
$$C \leq k\left(\left(\epsilon - \frac{1-\epsilon}{2}\right) \log N + O(1)\right)$$

and, assuming $\epsilon < 1/3$, this is no larger than $-r_\epsilon \log N$, where $r_\epsilon = \frac{1-3\epsilon}{4} > 0$. This yields for all $k \leq N^{\frac{1}{3}}$:

$$p_k \leq k e^{-r_\epsilon k \log N}.$$

As $\sum_{k=1}^{N^{\frac{1}{3}}} k e^{-r_\epsilon k \log N} = o(1)$, with high probability no subset $S \subset [N]$ of size $|S| \leq N^{\frac{1}{3}}$ is such that $|E_F^\circ(S, \bar{S})| < \epsilon |S|$.

For $|S| = k \in \left[N^{\frac{1}{3}}, \frac{N}{2} \right]$, we use the second upper bound of Lemma 5 on the number of size k sets made of l segments. Since this bound increases with l for



$l \leq \frac{\epsilon k}{2}$, we obtain

$$p_k \leq kN \binom{k-1}{\frac{\epsilon k}{2}} \binom{N-k-1}{\frac{\epsilon k}{2}} e^{-kD(\epsilon \|1 - \frac{k}{N})}.$$

Stirling's formula implies that the logarithm of this upper bound is no larger than

$$\begin{aligned} & \log(kN) \\ & + \frac{\epsilon k}{2} \log \frac{k-1}{\frac{\epsilon k}{2}} + \left(k-1 - \frac{\epsilon k}{2}\right) \log \frac{k-1}{k-1 - \frac{\epsilon k}{2}} \\ & + \frac{\epsilon k}{2} \log \frac{N-k-1}{\frac{\epsilon k}{2}} + \left(N-k-1 - \frac{\epsilon k}{2}\right) \log \frac{N-k-1}{N-k-1 - \frac{\epsilon k}{2}} \\ & - k\epsilon \log \frac{\epsilon}{1 - \frac{k}{N}} - k(1-\epsilon) \log \frac{1-\epsilon}{\frac{k}{N}}. \end{aligned}$$

The first term is at most $2 \log N$ since $k \leq \frac{N}{2}$. Using the inequality $\log u \leq u - 1$, the last term in the second line is no larger than $\frac{\epsilon k}{2}$. It follows that the sum of the last two terms in the first line, the last term in the second line and the first term in the last line is upper-bounded by $kf(\epsilon)$ for a function f of ϵ such that $f(\epsilon) = O(\epsilon \log \epsilon^{-1})$. Thus,

$$\begin{aligned} \log p_k & \leq 2 \log N + kf(\epsilon) + \frac{\epsilon k}{2} \log \frac{N-k-1}{\frac{\epsilon k}{2}} - k(1-\epsilon) \log \frac{1-\epsilon}{\frac{k}{N}} \\ & = 2 \log N + kf(\epsilon) + \frac{\epsilon k}{2} \log \frac{1 - \frac{k+1}{N}}{\frac{\epsilon}{2}} + \frac{\epsilon k}{2} \log \frac{N}{k} \\ & \quad - k(1-\epsilon) \log(1-\epsilon) - k(1-\epsilon) \log \frac{N}{k} \\ & = 2 \log N + kg(\epsilon) - k\left(1 - \frac{3}{2}\epsilon\right) \log \frac{N}{k} \end{aligned}$$

for some function $g(\epsilon)$ such that $g(\epsilon) = O(\epsilon \log \epsilon^{-1})$. This readily implies that for small enough $\epsilon > 0$, there exists a constant $s_\epsilon > 0$ such that, for $k \in \left[\left\lfloor N^{\frac{1}{3}}, \frac{N}{2} \right\rfloor \right]$, $p_k \leq e^{-s_\epsilon k}$. The corresponding sum is $o(1)$. Therefore, graph G_f is with high probability an ϵ -expander for some fixed constant $\epsilon > 0$.

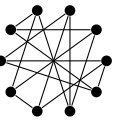
5 Controlling the mean

The goal of this section is to prove Lemma 2. Let G_f be the static graph during phase f . G_f is a 4-regular undirected graph on $[N]$, and we assume it is a (γ, c) -expander:

$$\forall k \leq \frac{N}{2}, \Phi_k(G) \geq \min\left(\gamma, \frac{c}{k}\right).$$

Our goal is to prove that for any fixed set S of size $k \leq \frac{N}{2}$, by the end of phase f (i.e. after T time steps), the expected number of pointers connecting S to \bar{S} satisfies

$$\mathbf{E}[|E_{f+1}(S, \bar{S})|] \geq \frac{1}{2\gamma} \min(\gamma k, 2c).$$



5. CONTROLLING THE MEAN

The proof is divided into two parts, arguing differently depending on the size k of S . Let $k_c = \frac{4c}{\gamma}$; sets of size k with $k \leq k_c$ (respectively $k > k_c$) will be referred to as small sets (respectively large sets).

5.1 Small sets: from partial expansion to partial spread

Let us now use Theorem 9 to prove the conclusion of Lemma 2 for small values of k .

For a fixed set S of size $k \leq k_c$, and a fixed node $i \in S$, let $X_i(t)$ denote the location of the pointer issued from i at time t . Under the dynamics we consider, $X_i(t)$ corresponds to an ordinary random walk on the graph G . Moreover, the assumptions of Lemma 2 guarantee that the graph G_f satisfies

$$\Phi_{3k}(G_f) \geq \min\left(\gamma, \frac{c}{3k}\right) \geq \min\left(\gamma, \frac{c}{3k_c}\right) = \frac{\gamma}{12}.$$

By Theorem 9, one therefore has

$$\mathbf{P}(X_i(T) \in S) \leq \frac{|S|}{3k} + \sqrt{3k+1} e^{-\lambda_2^* T},$$

where $\lambda_2^* = \frac{1}{2\Delta} \Phi_{3k}(G_f)^2 \geq \frac{\gamma^2}{1152}$. Recall that $T = \log^a N$ and that $\gamma = \log^{-\beta} N$. Furthermore, $1 < \beta < \frac{a-4}{4}$, implying that $a - 2\beta > 1$. We then have

$$\mathbf{P}(X_i(T) \in S) \leq \frac{1}{3} + \exp\left(-\frac{\log^{a-2\beta} N}{1152}\right) \leq \frac{1}{2}.$$

Summing over $i \in S$, we obtain that the expected number of pointers issued from S that point into S at the end of the phase is no larger than $\frac{k}{2}$, and therefore that

$$\mathbf{E}[|E_{f+1}(S, \bar{S})|] \geq \frac{k}{2} \geq \frac{1}{2\gamma} \min(\gamma k, 2c).$$

5.2 Large sets

Consider a fixed set S of size k such that $k_c < k \leq \frac{N}{2}$, and define $\pi_i(t)$ to be $\frac{1}{k}$ times the probability that a pointer issued from S targets i , conditionally on the initial configuration of these pointers at the beginning of the phase (corresponding to $t = 0$). Let $\pi_{(i)}(t)$ denote the i -th largest value in $(\pi_j(t))_{j \in [N]}$, and $\pi_{[m]}(t) = \sum_{i=1}^m \pi_{(i)}(t)$ denote the cumulative mass that the probability distribution $\pi(t)$ puts on the m nodes where its mass is the largest.

One clearly has that

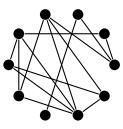
$$\pi_{(i)}(0) = \frac{1}{k} \mathbf{1}_{i \leq k}.$$

We now establish a property of the time derivative $\frac{d}{dt} \pi_{[m]}(t)$:

Lemma 6. *Under the assumptions of Lemma 1 that $\Phi_m(G_f) \geq \min(\gamma, \frac{c}{m})$ for all $m \in \llbracket 1, N \rrbracket$, one has the inequalities*

$$\frac{d}{dt} \pi_{[m]}(t) \leq -4 \sum_{j=1}^{c_m} (\pi_{(m-j+1)} - \pi_{(m-j+1+c_m)}), \quad (3.6)$$

where $c_m = \lfloor \frac{1}{4} \min(\gamma m, c) \rfloor$.



Proof. Assume to simplify notation that the permutation which sorts nodes i in $[N]$ in decreasing order of π_i is the identity, so that $\pi_i(t) = \pi_{(i)}(t)$. The time derivative of $\pi_{[m]}$ then reads

$$\frac{d}{dt} \pi_{[m]}(t) = \sum_{i \leq m} \sum_{j > m} \mathbf{1}_{(i,j) \in E_f^\circ} (\pi_j - \pi_i).$$

Indeed, changes in the mass $\pi_{[m]}$ result from interchange of pointer extremities i, j with $i \leq m$ and $j > m$, which occur at unit rate for $(i, j) \in E_f^\circ$; when one such interchange occurs, the expected change to $\pi_{[m]}$ is precisely $\pi_j - \pi_i$. Now the number of such edges is by assumption at least $\min(\gamma m, c)$. Moreover, the number of such edges adjacent to any node is at most 4, because the graph has degree bounded by 4.

The value of the right-hand side in the above equation, because the π_i are sorted in decreasing order, is minimized when the edges crossing the cut between $[m]$ are adjacent to nodes with index closest to m . The degree constraint then implies the upper bound (3.6). \square

Let $c' = \lfloor \frac{c}{4} \rfloor$ and $I = \llbracket k - \frac{2}{3}k_c, k + \frac{2}{3}k_c \rrbracket$. We now introduce an auxiliary process $(\nu_i(t))_{i \in [N], t \in \mathbb{R}}$ defined for all $i \in [N]$ via:

$$\begin{aligned} \nu_i(0) &= \frac{1}{k} \mathbf{1}_{i \in [k]} \\ \frac{d}{dt} \nu_i(t) &= 4 \mathbf{1}_{i \in I} [(\nu_{i-c'}(t) - \nu_i(t)) \mathbf{1}_{i-c' \in I} + 4(\nu_{i+c'} - \nu_i) \mathbf{1}_{i+c' \in I}]. \end{aligned}$$

The probability distribution $\nu(t)$ is readily interpreted as the law at time t of a random walk started with uniform distribution on $[k]$, that jumps from i to $i + c'$ (resp., $i - c'$) at rate 4, provided both i and the destination $i \pm c'$ lie in I .

Denoting $\nu_{[m]}(t) = \sum_{j=1}^m \nu_j(t)$ for all $m \in \llbracket 1, N \rrbracket$, we then have the following

Lemma 7. *For all $t > 0$, $m \in [N]$, it holds that*

$$\pi_{[m]}(t) \leq \nu_{[m]}(t).$$

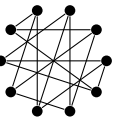
Proof. Introduce the differences $\delta_m(t) := \pi_{[m]}(t) - \nu_{[m]}(t)$. It is readily seen that $\delta_m(0) = 0$ for all $m \in [N]$. Inequality (3.6) of Lemma 6 implies that

$$\frac{d}{dt} \pi_{[m]}(t) \leq -4 \sum_{j=1}^{c'} \mathbf{1}_{m-j+1 \in I} \mathbf{1}_{m-j+1+c' \in I} (\pi_{(m-j+1)} - \pi_{(m-j+1+c')}). \quad (3.7)$$

Indeed, each term in the summation of the right-hand side of (3.6) is non-negative. The j -th term in the summation in the right-hand side of (3.7) is included only if $m - j + 1 \in I$ and $m - j + 1 + c' \in I$. The first condition implies that

$$m - j + 1 \geq k - \frac{2}{3}k_c \geq \frac{1}{3}k_c = \frac{4}{3} \frac{c}{\gamma}.$$

In turn this implies that $\gamma m \geq c$, so that $c_m = c'$. Thus the summation in the right-hand side of (3.7) runs over a subset of indices in the summation in the right-hand side of (3.6), and (3.7) follows from (3.6).



6. PROOF OF THEOREM 9

By definition of $\nu_i(t)$, one has

$$\frac{d}{dt} \nu_{[m]}(t) = -4 \sum_{j=1}^{c'} \mathbf{1}_{m-j+1 \in I} \mathbf{1}_{m-j+1+c' \in I} (\nu_{(m-j+1)} - \nu_{(m-j+1+c')}). \quad (3.8)$$

For all $m \in \llbracket 1, N \rrbracket$, there thus exists an integer $i_m \geq 0$ such that $m - i_m \geq 1$, $m + i_m \leq N$ and

$$\begin{aligned} \frac{d}{dt} \pi_{[m]}(t) &\leq -4(2\pi_{[m]} - \pi_{[m-i_m]} - \pi_{[m+i_m]}), \\ \frac{d}{dt} \nu_{[m]}(t) &= -4(2\nu_{[m]} - \nu_{[m-i_m]} - \nu_{[m+i_m]}), \end{aligned}$$

so that

$$\frac{d}{dt} \delta_m \leq -4(2\delta_m - \delta_{m-i_m} - \delta_{m+i_m}).$$

In the above, as is easily seen, necessarily $i_1 = 0$, so that we have the boundary condition $\delta_1 \leq 0$. Also, since $\pi_{[N]} = \nu_{[N]} = 1$, we have $\delta_N = 0$. The previous equation then implies that necessarily, the supremum over $m \in \llbracket 1, N \rrbracket$ of δ_m cannot become positive, because its derivative is always non-positive. \square

By the previous lemma, an upper bound on $\pi_{[k]}(T)$ is provided by $\nu_{[k]}(T)$. However the latter quantity is simpler to analyze. It can be interpreted as $\frac{1}{k}$ times the average number of points of $\frac{2}{3}k_c$ random walks initialized at each point in $\llbracket k - \frac{2}{3}k_c, k \rrbracket$ which fall within $[k]$ at time T . These walks proceed with jumps of size $\pm c'$ at rate 4, constrained to not leave interval $I = \llbracket k - \frac{2}{3}k_c, k + \frac{2}{3}k_c \rrbracket$.

For a given initial condition $i \in [k - \frac{2}{3}k_c]$, the number of sites it can visit is of the order of $\frac{4}{3} \frac{k_c}{c'} = \Theta(\log^\beta N)$. Classical results on the nearest neighbor random walk on an interval $\llbracket 1, M \rrbracket$ state that it mixes in time of the order of M^2 [36]. Thus each of the random walks just introduced mix in time $O(\log^{2\beta} N) = o(T)$, because $2\beta < a$. We therefore have the following evaluation:

$$\pi_{[k]}(T) \leq \nu_{[k]}(T) \leq 1 - \frac{\frac{2}{3}k_c}{k} \left(\frac{1}{2} - o(1) \right).$$

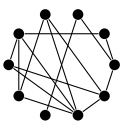
The expected number $\mathbf{E}[|E_{f+1}(S, \bar{S})|]$ is then lower-bounded by

$$\mathbf{E}[|E_{f+1}(S, \bar{S})|] \geq \frac{2}{3}k_c \left(\frac{1}{2} - o(1) \right) = 4 \frac{c}{\gamma} \left(\frac{1}{3} - o(1) \right) \geq \frac{1}{2\gamma} 2c.$$

The announced result follows.

6 Proof of Theorem 9

Proof. In vector form the law $\pi(t)$ of the random walk on G at time t reads $\pi(t) = e^{-tL} \pi(0)$, where L is the graph's Laplacian. Its entries $\pi_i(t)$ are thus linear combinations of n functions of the form $e^{-\lambda_j t}$, where λ_j are the eigenvalues of L , and so is the difference $\pi_i(t) - \pi_j(t)$. It can be shown by induction on N



CHAPTER 3. RAPID MIXING OF LOCAL DYNAMIC GRAPHS

that such linear combinations of N distinct exponential functions are either identically zero in t , or admit at most $N - 1$ distinct roots in t . Thus for any $i \neq j$, either $\pi_i(t) \neq \pi_j(t)$ except perhaps for finitely many t , or else $\pi_i(t) = \pi_j(t)$ for all $t \geq 0$.

We can thus split \mathbb{R}_+ into finitely many intervals $I^{(1)} = [0, t_1)$, $I^{(2)} = [t_1, t_2)$, \dots , and on each interval $I^{(j)}$ determine a particular permutation $\sigma^{(j)} \in \mathfrak{S}_N$ such that for all j , and all $t \in I^{(j)}$, one has

$$\pi_{\sigma^{(j)}(1)}(t) \geq \pi_{\sigma^{(j)}(2)}(t) \geq \dots \geq \pi_{\sigma^{(j)}(N)}(t).$$

For t in any given interval $I^{(j)}$, we will maintain an auxiliary probability distribution on $\llbracket 1, k+1 \rrbracket$, denoted $(\nu_i(t))_{i \in \llbracket k+1 \rrbracket}$. This distribution can be interpreted as that of a random walk on a graph $G^{(j)}$ with node set $\llbracket k+1 \rrbracket$, obtained from G as follows. We identify node $\sigma^{(j)}(i)$ in G with node i in $G^{(j)}$ for all $i \in \llbracket k \rrbracket$, and collapse all nodes $\sigma^{(j)}(u)$, $u > k$ to form node $k+1$. All edges are then preserved, so that the adjacency matrix $A^{(j)}$ of $G^{(j)}$ is given by

$$\begin{aligned} A_{u,v}^{(j)} &= A_{\sigma^{(j)}(u), \sigma^{(j)}(v)}, & u, v \in \llbracket k \rrbracket, \\ A_{u, k+1}^{(j)} &= \sum_{v=k+1}^N A_{\sigma^{(j)}(u), \sigma^{(j)}(v)}, & u \in \llbracket k \rrbracket, \end{aligned}$$

where A is the adjacency matrix of G . For convenience, we denote by $\pi_{(i)}(t)$ the i -th largest entry of distribution $\pi(t)$. Thus for $t \in I^{(j)}$, $\pi_{(i)}(t) = \pi_{\sigma^{(j)}(i)}(t)$.

The result of the theorem will then follow from the combination of two ingredients. We first show in Lemma 8 below that, for all t , one has the following bound:

$$\pi_{(i)}(t) \leq \nu_{\min(i, k+1)}(t), \quad i \in \llbracket N \rrbracket, t \geq 0. \quad (3.9)$$

We then establish in Lemma 9 below that for all j , the second smallest eigenvalue $\lambda_2^{(j)}$ of the Laplacian of $G^{(j)}$ is lower-bounded by λ_2^* given in (3.2), where crucially Δ is the largest node degree in G , not in $G^{(j)}$.

This readily implies the L^2 control

$$\sum_{i \in \llbracket k+1 \rrbracket} \left| \nu_i(t) - \frac{1}{k+1} \right|^2 \leq e^{-2\lambda_2^* t}.$$

Cauchy-Schwarz inequality then gives the following control on variation distance:

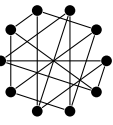
$$\sum_{i \in \llbracket k+1 \rrbracket} \left| \nu_i(t) - \frac{1}{k+1} \right| \leq \sqrt{k+1} e^{-\lambda_2^* t}.$$

Together, these two results entail that for all $s \leq k$,

$$\sum_{i \in \llbracket s \rrbracket} \pi_{(i)}(t) \leq \frac{s}{k+1} + \sqrt{k+1} e^{-\lambda_2^* t}, \quad (3.10)$$

which is the announced result. \square

Lemma 8. *The distributions $\pi(t)$ and $\nu(t)$ verify bound (3.9).*



6. PROOF OF THEOREM 9

Proof. The bound trivially holds at $t = 0$. We can establish it by induction on each interval $I^{(j)}$. Let us consider one such interval, and assume that the property holds at its left end. For notational simplicity we will assume that $\sigma^{(j)}$ is the identity, so that on this interval $\pi_i(t) = \pi_{(i)}(t)$. Introduce the notation

$$\delta_i(t) = \pi_i(t) - \nu_{\min(i, k+1)}(t), \quad i \in [N].$$

For any pair of vertices (i, j) in $[N]$, write $i \sim j$ if i and j are neighbors in G . One has the following time derivatives

$$\begin{aligned} \frac{d}{dt} \pi_i &= \sum_{\substack{j \in [k] \\ j \sim i}} (\pi_j - \pi_i) + \sum_{\substack{j \notin [k] \\ j \sim i}} (\pi_j - \pi_i) & i \in [N] \\ \frac{d}{dt} \nu_i &= \sum_{\substack{j \in [k] \\ j \sim i}} (\nu_j - \nu_i) + \sum_{\substack{j \notin [k] \\ j \sim i}} (\nu_{k+1} - \nu_i) & i \in [k] \\ \frac{d}{dt} \nu_{k+1} &= \sum_{i \notin [k]} \sum_{\substack{j \in [k] \\ j \sim i}} (\nu_j - \nu_{k+1}). \end{aligned}$$

By the previous display one has for $i \in [k]$:

$$\frac{d}{dt} \delta_i = \sum_{\substack{j \in [n] \\ j \sim i}} (\delta_j - \delta_i). \quad (3.11)$$

Note that, because the values π_i are sorted, for all $j \notin [k]$, $\pi_j - \pi_{k+1} \leq 0$. This together with the expression for the time derivative of π_{k+1} yield

$$\frac{d}{dt} \pi_{k+1} \leq \sum_{\substack{j \in [k] \\ j \sim k+1}} (\pi_j - \pi_{k+1}).$$

Thus

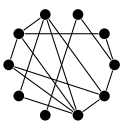
$$\begin{aligned} \frac{d}{dt} \delta_{k+1} &\leq \sum_{\substack{j \in [k] \\ j \sim k+1}} (\pi_j - \pi_{k+1}) - \sum_{i \notin [k]} \sum_{\substack{j \in [k] \\ j \sim i}} (\nu_j - \nu_{k+1}) \\ &= \sum_{\substack{j \in [k] \\ j \sim k+1}} (\delta_j - \delta_{k+1}) - \sum_{i \notin [k+1]} \sum_{\substack{j \in [k] \\ j \sim i}} (\nu_j - \nu_{k+1}). \end{aligned} \quad (3.12)$$

$$(3.13)$$

Let us argue by contradiction, and assume that there exists $t \in \mathbb{R}_+$ and $i \in [N]$ for which $\delta_i(t) > 0$. Let $\delta(t) = \sup\{\delta_j(t) \mid j \in [N]\}$. As the π_j are sorted in decreasing order, one also has $\delta(t) = \sup\{\delta_j(t) \mid j \in [k+1]\}$.

Since the $\delta_j(t)$ are linear combinations of finitely many exponentials, we can then identify an interval $J = [a, b]$ such that on J , for some $i \in [k+1]$, $\delta(t) = \delta_i(t)$, and moreover $\delta(a) = 0$ and $\delta(t) > 0$ for $t \in (a, b]$.

Assume that $i \in [k]$. From expression (3.11), we see that on J , $\frac{d}{dt} \delta = \frac{d}{dt} \delta_i \leq 0$. This contradicts the fact that $\delta > 0$ on $(a, b]$.



CHAPTER 3. RAPID MIXING OF LOCAL DYNAMIC GRAPHS

Assume then that $i = k + 1$. Then on J one has that, for all $j \in [k]$, since the π_j are sorted,

$$\nu_{k+1} = \pi_{k+1} - \delta_{k+1} \leq \pi_{k+1} \leq \pi_j = \nu_j + \delta_j \leq \nu_j + \delta_{k+1}.$$

Thus for all $j \in [k]$, $\nu_{k+1} - \nu_j \leq \delta_{k+1}$. It then follows from (3.12) that

$$\frac{d}{dt} \delta_{k+1} \leq 0 + \alpha \delta_{k+1},$$

where

$$\alpha = \sum_{i \notin [k+1]} |\{j \in [k] \mid j \sim i\}|.$$

Gronwall's lemma (see e.g. [35]) then implies that $\delta_{k+1} \leq 0$ on J , a contradiction. \square

Remark 2. When we move from interval $I^{(j)}$ to $I^{(j+1)}$ one can check that the meaning of distribution ν is preserved: we may change the permutation sorting the entries π_i , which results in a change in the graph used to define the evolution of ν , but while the vertex to which ν_i refers may change, in that case the corresponding mass does not change.

Lemma 9. Given a graph G on vertex set $[N]$ with maximal degree Δ and for fixed $k \leq \frac{N}{2}$, associated isoperimetric constant $\Phi_k(G)$, consider the graph G' obtained by collapsing $N - k$ nodes into a single node as previously described. Then the resulting Laplacian matrix L has spectral gap at least $\lambda_2 \geq \lambda_2^*$, where $\lambda_2^* := \frac{\Phi_k(G)^2}{2\Delta}$.

Proof. Without loss of generality we assume nodes $k + 1, \dots, N$ of G have been collapsed into node $k + 1$ of G' . Let f be an eigenvector of L associated with its second smallest eigenvalue λ_2 . We can always choose f such that $f_{k+1} \leq 0$.

For $v \in [k + 1]$, define $g_v = \max(f_v, 0)$, and thus $g_{k+1} = 0$. Let

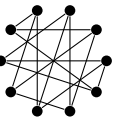
$$W = \{v \in [k + 1] \mid f_v > 0\}.$$

Letting $(a_{uv})_{u,v \in [N]}$ denote the adjacency matrix of graph G , one has

$$\begin{aligned} \lambda_2 \sum_{u \in W} f_u^2 &= \sum_{u \in W} (Lf)_u f_u \\ &= \sum_{u \in W} \left(d_u f_u - \sum_{v \in [k+1]} a_{uv} f_v \right) f_u \\ &= \sum_{u \in W} \sum_{v \in [k+1]} a_{uv} (f_u - f_v) f_u \\ &= \sum_{u \in W} \sum_{v \in W} a_{uv} (f_u - f_v) f_u + \sum_{u \in W} \sum_{v \notin W} a_{uv} (f_u - f_v) f_u \\ &\geq \sum_{u \in W} \sum_{v \in W} a_{uv} (f_u - f_v) f_u + \sum_{u \in W} \sum_{v \notin W} a_{uv} f_u^2 \\ &= \langle Lg, g \rangle \end{aligned}$$

Thus

$$\lambda_2 \geq \frac{\langle Lg, g \rangle}{\langle g, g \rangle} = K.$$



6. PROOF OF THEOREM 9

On the other hand,

$$\begin{aligned} \sum_{(uv) \in E} a_{uv} (g_u + g_v)^2 &= 2 \sum_{(uv) \in E} a_{uv} (g_u^2 + g_v^2) - \sum_{(uv) \in E} a_{uv} (g_u - g_v)^2 \\ &\leq 2 \sum_{v \in V} d_v g_v^2 \\ &\leq 2\Delta \langle g, g \rangle, \end{aligned}$$

where we have used the fact that $g_{k+1} = 0$ to upper bound each product $d_v g_v^2$ by Δg_v^2 .

By Cauchy-Schwarz inequality,

$$\left(\sum_{(uv) \in E} a_{uv} |g_u^2 - g_v^2| \right)^2 \leq \left(\sum_{(uv) \in E} a_{uv} (g_u - g_v)^2 \right) \left(\sum_{(uv) \in E} a_{uv} (g_u + g_v)^2 \right).$$

Combined, these bounds give

$$\begin{aligned} K &= \frac{(\sum_{(uv) \in E} a_{uv} (g_u - g_v)^2) (\sum_{(uv) \in E} a_{uv} (g_u + g_v)^2)}{\langle g, g \rangle \sum_{(uv) \in E} a_{uv} (g_u + g_v)^2} \\ &\geq \frac{(\sum_{(uv) \in E} a_{uv} |g_u^2 - g_v^2|)^2}{2\Delta \langle g, g \rangle^2}. \end{aligned}$$

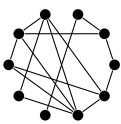
Let $0 = t_0 < t_1 < \dots < t_m$ be the distinct values taken by the g_v . For $i = 0, \dots, m$, let $V_i = \{v \in V \mid g_v \geq t_i\}$. Thus for $i > 0$, $k+1 \notin V_i$. Let

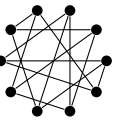
$$\begin{aligned} M &:= \sum_{(uv) \in E} a_{uv} |g_u^2 - g_v^2| \\ &= \sum_{i=1}^m \sum_{\substack{(uv) \in E \\ g_v < g_u = t_i}} a_{uv} (g_u^2 - g_v^2) \\ &= \sum_{i=1}^m \sum_{u: g_u = t_i} \sum_{\substack{v: g_v = t_j \\ j < i}} a_{uv} (t_i^2 - t_{i-1}^2 + \dots - t_{j+1}^2 + t_{j+1}^2 - t_j^2) \\ &= \sum_{i=1}^m \sum_{u \in V_i} \sum_{v \notin V_i} a_{uv} (t_i^2 - t_{i-1}^2) \\ &= \sum_{i=1}^m |E(V_i, \bar{V}_i)| (t_i^2 - t_{i-1}^2) \\ &\geq \Phi_k(G) \sum_{i=1}^m |V_i| (t_i^2 - t_{i-1}^2) \\ &= \Phi_k(G) \sum_{i=1}^m t_i^2 (|V_i| - |V_{i+1}|) \\ &= \Phi_k(G) \langle g, g \rangle. \end{aligned}$$

Combined, these results yield

$$\lambda_2 \geq K \geq \frac{(\Phi_k(G) \langle g, g \rangle)^2}{2\Delta \langle g, g \rangle^2} = \lambda_2^*.$$

□

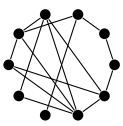




Chapter 4

Oracle Sampling

1	Introduction	66
2	Coupling From the Past	67
3	Oracle Sampling	69
3.1	Oracle Sampling with Bounding Chains	70
3.2	Oracle Sampling in CFTP	70
3.3	Proof of Correctness	74
3.4	Complexity	79
3.5	Variants of Oracle Sampling	81
4	Independent Sets	82
4.1	Sampling algorithms	82
4.2	Star graph	84
4.3	Numerical experiments	86



1 Introduction

The Monte Carlo Markov chain (MCMC) scheme is used in many domains, from statistical physics [19] to combinatorial optimization [28], to telecommunication networks [48, 32] and information theory [15]. The method often requires solving two problems. The first consists in constructing a Markov chain whose stationary distribution is the target sampling distribution. There exist many techniques to achieve this goal, such as Metropolis-Hastings [43, 25] and Gibbs sampling [20].

The second is termination. With the classical MCMC algorithm [4], the Markov chain is simulated for a predefined number of iterations, after which the current state is returned. Determining the required number iterations consists in computing the *mixing time* of the Markov chain, something which has proven to be very difficult to do. Other methods, such as the Coupling From The Past (CFTP) algorithm by Propp and Wilson [45], are perfect sampling algorithms: they terminate without the need for any additional parameter, and the output follows the exact target distribution.

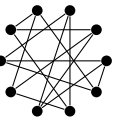
One of the key features that makes CFTP a viable algorithm is monotonicity [45]. By using a monotone Markov chain and period doubling, we remove the linear dependency of the running time of the algorithm on the size of the state space. The resulting complexity is then comparable to the mixing time of the Markov chain.

For non-monotonous Markov chains, a similar technique can be used, based on bounding chains [28]. By creating a simplified envelope that encompasses all possible trajectories, the above technique can still be used. The efficiency of the resulting algorithm depends on the complexity and tightness of the bounding chain.

We present a new method for reducing the running time of the CFTP algorithm: oracle sampling. With bounding chains, many events do not actually change the state of the chain. In an approach similar to importance sampling, we remove these transitions, focusing only on those that participate in the evolution of this bounding state. In order to preserve the correct stationary distribution, the algorithm must dynamically add random events so as to prevent the appearance of any bias.

Oracle sampling is most efficient when addressing local dynamics, for example when using Gibbs samplers. These are situations in which sampling transitions is computationally efficient. We test the algorithm on one such model, independent sets, and compare its performance with that of the original CFTP algorithm.

This chapter is organized as follows. Section 2 gives an overview of the basic CFTP algorithm and of the use of bounding chains. Our main contribution is presented in Section 3. We start in Section 3.1 by introducing the notion of active and passive events. Section 3.2 contains a detailed explanation on how to correctly skip events in the context of CFTP, resulting in Algorithm 8. The validity of the algorithm is provided by Theorem 12, which constitutes the main result of this paper. We apply this algorithm in the context of random independent sets in Section 4.



2 Coupling From the Past

In this section, we give the original coupling from the past algorithm, as introduced by Propp and Wilson [45], as well as some usual variations.

Since CFTP relies on the ability to couple Markov chains, we use a Markov automata representation.

The usual notations are used: \mathcal{A}^* for the set of finite words, \mathcal{A}^ω for the set of infinite words, and \mathcal{A}^∞ for all finite and infinite words. Words are denoted $u_{i \rightarrow j} = (u_i, \dots, u_j)$, or the empty word ϵ if $i > j$, and the length of a word u is $|u|$. To simplify notation, we write $x \cdot u_{i \rightarrow j}$ for $x \cdot u_i \cdot \dots \cdot u_j$. The concatenation of two words u and v is $u \odot v$, and, for $i \leq j$, we denote

$$\bigcirc_{k=i}^j u^{(k)} = u^{(i)} \odot u^{(i+1)} \odot \dots \odot u^{(j)} \quad \text{and} \quad \bigcirc_{k=j}^i u^{(k)} = u^{(j)} \odot u^{(j-1)} \odot \dots \odot u^{(i)}.$$

Coupling from the past Recall that, for a grand coupling $((X_t^x)_{t \in \mathbb{N}})_{x \in \mathcal{S}}$ with coupling time τ , X_τ^x is not distributed according to the stationary distribution of the Markov automaton. Coupling from the past consists in constructing a similar sequence which does yield the correct distribution.

Consider a negatively indexed infinite word $U_{-\infty \rightarrow -1}$ distributed according to $D^{\otimes \mathbb{N}}$. For all $x \in \mathcal{S}$ and $i \in \mathbb{N}$, let

$$Y_i^{(x)} = x \cdot U_{-i} \cdot \dots \cdot U_{-1} = x \cdot U_{-i \rightarrow -1}$$

and

$$S_{-i} = \{Y_i^{(x)} \mid x \in \mathcal{S}\},$$

which we also denote $S_{-i} = \mathcal{S} \cdot U_{-i \rightarrow -1}$.

Call τ_b the smallest index such that $S_{-\tau_b}$ is a singleton. τ_b is called the *backward coupling time*.

For all $x \in \mathcal{S}$ and $i \in \mathbb{N}$, $Y_i^{(x)}$ can be interpreted as the state at time 0 of the Markov chain starting in x at time $-i$. τ_b is the first instant i such that the grand coupling started at time $-i$ has coupled at time 0.

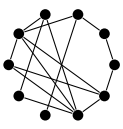
The CFTP algorithm is based on the following result.

Theorem 11 ([45]). *The unique element of $S_{-\tau_b}$ is almost surely distributed according to π , and $\mathbf{E}[\tau_b] = \mathbf{E}[\tau]$.*

Notice also that for all $j \geq i$, $S_{-j} \subseteq S_{-i}$, and namely that the S_{-i} are therefore all equal to $S_{-\tau_b}$ for $i \geq \tau_b$.

This result can be used to generate samples distributed according π , as illustrated in Algorithm 4. Note that it is important to reuse the previous sequence of letters at each iteration, prepending the newly generated letter; if the whole word is re-sampled, the output of the algorithm is incorrect, as detailed in [24].

The complexity of this algorithm is $O(\mathbf{E}[\tau] |\mathcal{S}| \Gamma(\cdot))$, where $\Gamma(\cdot)$ is the complexity of the transition function \cdot .



Algorithm 4 Coupling From the Past (CFTP)

```

function CFTP( $(\mathcal{S}, \mathcal{A}, D, \cdot)$ )
  for  $x \in \mathcal{S}$  do
     $Y(x) \leftarrow x$  Initialize  $Y_0$ , which is the identity vector
  end for
  repeat
     $U_{-(i+1)} \leftarrow \text{DRAW}(D)$  Draw a new random letter  $U_{-(i+1)}$ 
    for  $x \in \mathcal{S}$  do
       $Z(x) \leftarrow Y(x \cdot U_{-(i+1)})$  Compute  $Y_{i+1}$  from  $Y_i$  and  $U_{-(i+1)}$ 
    end for
     $Y \leftarrow Z$ 
  until  $|Y(\mathcal{S})| = 1$  Repeat until all  $Y^{(x)}$  are equal
  return  $\text{UNIQUEELEMENTOF}(Y(\mathcal{S}))$ 
end function

```

Bounding chains The above complexity is often limiting: even though $\Gamma(\cdot)$ is generally a constant, and $\mathbf{E}[\tau]$ is of order t_{mix} , the size of \mathcal{S} can be exponentially large. Variants of the CFTP algorithm have been introduced to overcome this dependence on $|\mathcal{S}|$. The most common of these is the use of bounding chains [28].

Formally, consider a pair (\mathcal{B}, \circ) such that \mathcal{B} is a subset of the power set of \mathcal{S} containing \mathcal{S} , and $\circ : \mathcal{B} \times \mathcal{A} \rightarrow \mathcal{B}$ is a transition function which bounds \cdot , in the sense that

$$\forall x \in \mathcal{S}, \forall B \in \mathcal{B}, \forall a \in \mathcal{A}, \quad x \in B \Rightarrow x \cdot a \in B \circ a.$$

Such a pair is called a bounding chain for \mathbf{A} .

As with $(S_{-i})_{i \in \mathbb{N}}$, define $(B_{-i})_{i \in \mathbb{N}}$ as

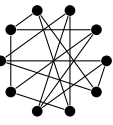
$$B_{-i} = \mathcal{S} \circ U_{-i} \circ \dots \circ U_{-1} = \mathcal{S} \circ U_{-i \rightarrow -1}.$$

Notice that, for all $i \in \mathbb{N}$, $S_{-i} \subseteq B_{-i}$, which implies that if B_{-i} is a singleton, so is S_{-i} . We can therefore adapt the CFTP algorithm to keep track of the bounding chain $(B_{-i})_{i \in \mathbb{N}}$ instead of $(S_{-i})_{i \in \mathbb{N}}$.

The complexity of this new algorithm is $O(\mathbf{E}[\tau_B^2] \Gamma(\circ))$, where $\Gamma(\circ)$ is the complexity of the transition function \circ , and is generally either constant or logarithmic in the size of the state space \mathcal{S} . The random variable τ_B is the forward coupling time of the bounding chain, i.e. the smallest i such that $\mathcal{S} \circ U_{-i \rightarrow -1}$ is a singleton. The square factor is due to the fact that $B_{-(i+1)}$ cannot be computed from B_{-i} and $U_{-(i+1)}$, and the whole chain must therefore be recomputed at each iteration.

This square factor can be overcome by starting from more than one step back in time at each iteration: if the number of prepended letters is doubled at each iteration, rather than adding one unique letter, then the complexity drops down to $O(\mathbf{E}[\tau_B] \Gamma(\circ))$. This technique is referred to as *period doubling* [45], and its implementation is given in Algorithm 5.

Note that, in doing this, the final word $U_{-i \rightarrow -1}$ may be such that $i > \tau_B$. Theorem 11 states that this does not change the output, which is therefore still distributed according to π .



3. ORACLE SAMPLING

Algorithm 5 CFTP with Bounding Chains

```

function BOUNDED-CFTP( $(\mathcal{S}, \mathcal{A}, D, \cdot), (\mathcal{B}, \circ)$ )
   $k \leftarrow 1$ 
   $U \leftarrow \epsilon$ 
   $B \leftarrow \mathcal{S}$ 
  repeat
     $U \leftarrow \text{DRAW}(D^{\otimes k}) \circ U$  Prepend  $k$  letters
     $B \leftarrow \mathcal{S} \circ U$  Recompute the whole chain
     $k \leftarrow 2k$  Double the period
  until  $|B| = 1$ 
  return UNIQUEELEMENTOF( $B$ )
end function

```

Random increments The period doubling technique can be generalized. One possible modification is to randomize the number of new letters at each iteration. We present here the case of geometrically distributed increments, as it will serve for the basis of our own algorithm.

Fix a probability p . We extend the alphabet \mathcal{A} by adding a new letter, \sharp , which serves as a delimiter. The new alphabet is denoted \mathcal{A}_\sharp . Call D_p the distribution over \mathcal{A}_\sharp such that $D_p(\sharp) = p$ and, for all $a \in \mathcal{A}$, $D_p(a) = (1-p)D(a)$. Enhance the transition operators \cdot and \circ so that $x \cdot \sharp = x$ and $B \circ \sharp = B$. Finally, let $|u|_\sharp$ denote the number of occurrences of the letter \sharp in u .

Drawing letters i.i.d. according to D_p until a \sharp is drawn gives a word of geometric length, whose letters are i.i.d. according to D (with the exception of the final \sharp). The distribution of this word is denoted G_p .

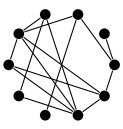
We can prepend such words during our algorithm, halving p at each iteration, to obtain a CFTP algorithm with geometric increments. The word used at the n -th iteration of the algorithm is therefore of the form $U^{(n)} = \odot_{i=n}^1 V^{(i)}$, with the $V^{(i)}$ independently distributed according to $G_{2^{-i}}$, and its distribution is denoted R_n .

If the \sharp are omitted in the increments, then the sequence of letters at each iteration is i.i.d. according to D , so the final output is distributed according to π . Since the \sharp do not change the state of the chain, including them in the increments does not change the output of the algorithm, nor its distribution. Though including these \sharp may seem unnecessary, their presence is essential in our own algorithm, so we include them here.

The CFTP algorithm with geometric increments is given in Algorithm 6. Its complexity is the same as that for deterministic increments: $O(\mathbf{E}[\tau_B] \Gamma(\circ))$.

3 Oracle Sampling

In this section, we present our contribution: a variation of the CFTP algorithm based on a technique similar to that introduced in [44], called *oracle sampling*. Just like regular CFTP, this method allows us to sample random variables according to a stationary distribution π . However, the overall complexity of the algorithm can be arbitrarily smaller, depending on the Markov automaton. This is illustrated in Section 4.



Algorithm 6 Geometric CFTP

```

function GEOMETRIC-CFTP( $(\mathcal{S}, \mathcal{A}, D, \cdot), (\mathcal{B}, \circ)$ )
   $p \leftarrow \frac{1}{2}$ 
   $U \leftarrow \epsilon$ 
   $B \leftarrow \mathcal{S}$ 
  repeat
     $V \leftarrow \epsilon$ 
    repeat
       $A \leftarrow \text{DRAW}(D_p)$ 
       $V \leftarrow V \odot A$ 
    until  $A = \#$  Adds up to  $V \leftarrow \text{DRAW}(G_p)$ 
     $U \leftarrow V \odot U$ 
     $B \leftarrow \mathcal{S} \circ U$ 
     $p \leftarrow \frac{1}{2}p$ 
  until  $|B| = 1$ 
  return UNIQUEELEMENTOF( $B$ )
end function

```

This gain is achieved by sampling only “active” letters throughout the algorithm, i.e. letters which modify the state of the bounding chain.

3.1 Oracle Sampling with Bounding Chains

We first consider a simple “forward” bounding chain, outside of the context of the CFTP algorithm: let $U_{1 \rightarrow +\infty}$ be distributed according to $D^{\otimes \mathbb{N}}$, and $(B_i)_{i \in \mathbb{N}}$ be the bounding chain such that

$$\forall i \in \mathbb{N}, B_i = \mathcal{S} \circ U_{1 \rightarrow i}.$$

We say a letter a is *passive* with respect to B_i if $B_i \circ a = B_i$, and that it is *active* otherwise. Let $\mathcal{A}_+^{B_i}$ be the set of active letters with respect to B_i , and $\mathcal{A}_-^{B_i}$ the set of passive letters.

Consider a new sequence $V_{1 \rightarrow +\infty}$ such that the V_i are distributed according to

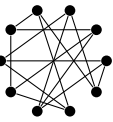
$$D(\cdot \mid \mathcal{A}_+^{\mathcal{S} \circ V_{1 \rightarrow i-1}}),$$

i.e. D conditioned on being active letters. For all $i \in \mathbb{N}$, let $B_i^0 = \mathcal{S} \circ V_{1 \rightarrow i}$. The new bounding chain $(B_i^0)_{i \in \mathbb{N}}$ behaves similarly to $(B_i)_{i \in \mathbb{N}}$, but has a smaller coupling time τ_0 . This gain in speed comes at a price, however, as the limit distribution of the asymptotically unique element of $(B_i^0)_{i \in \mathbb{N}}$ is no longer π .

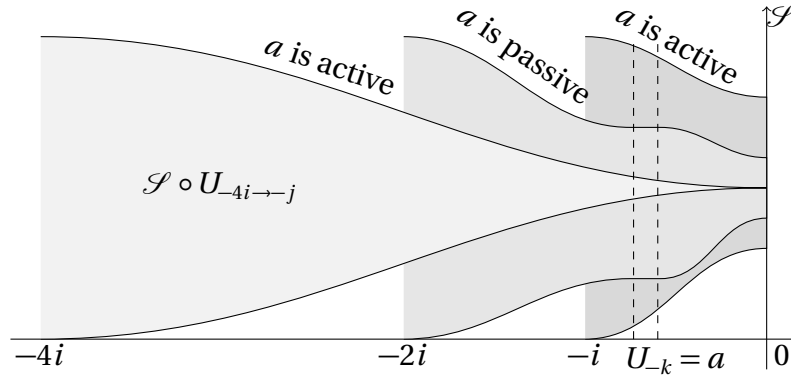
This method of speeding up convergence is referred to as *oracle sampling*. The term “oracle” is due to the fact that we consider sampling active letters to have constant complexity, a strong assumption requiring namely that the set of active letters be known at each step.

3.2 Oracle Sampling in CFTP

We now adapt the CFTP algorithm to make use of oracle sampling.



3. ORACLE SAMPLING



Notice that the letter $U_{-k} = a$, is initially active when the bounding chain starts from $-i$, then passive when starting from $-2i$, and active again when starting from $-4i$.

Figure 4.1 – Letters can change between being active and passive

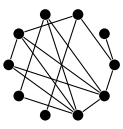
The process is not as straightforward as for the forward case, since the state of the bounding chain at a given moment $-i$ changes each time the algorithm starts from further back in time. The letter U_{-i} can therefore be active at first, then become passive later on in the unwinding of the algorithm, then active again, etc. This is illustrated in Figure 4.1.

Removing letters as they become passive is simple enough, but the reverse is not true: letters that were removed or initially skipped must be accounted for when starting the bounding chain from further back in time. Keeping these letters in memory rather than discarding them is not viable, as it would require actually drawing the skipped letters, thereby defeating the purpose of this entire method. We must therefore reinsert such letters back in at random to compensate for their removal.

Recall, however, that re-sampling previously generated letters in the classical CFTP algorithm yields incorrect results in most cases. We must therefore be careful when inserting new letters, taking care to compensate for the inherent bias this introduces so as to ensure that the output is still distributed according to π .

Geometric increments To achieve this, we rely heavily of the use of geometric increments for the word lengths. The memoryless property of the geometric distribution is vital due to the following observation: suppose that the initial word $U_{-i \rightarrow -1}$ was distributed according to G_p ; the probability of a letter having been removed after a given position $-j$ only depends on $S \circ U_{-i \rightarrow -j}$ and p , *not* on the actual letters drawn before or after, nor on the length of the word.

One caveat is that, when running the geometric CFTP algorithm, the parameter p is not the same throughout the word used to generate the bounding chain. In order to be able to keep track of which parameter was used to generate each portion of this word, we rely on the presence of the letter \sharp , which delimits portions of the word that were generated using a same parameter. Note that these \sharp must never be removed or reinserted once added to the word. They are therefore treated as active letters throughout the algorithm, despite the fact that they do not modify the state of the bounding chain: $\forall B \in \mathcal{B}, \sharp \in \mathcal{A}_+^B$.



We now formally define the removal and insertion operations, referred to as *contracting* and *expanding*.

Contracting Contracting a word consists in removing all the passive letters from that word. Given a state of the bounding chain $B \in \mathcal{B}$ and a word $u \in \mathcal{A}_\#^\infty$, the contraction of u from B is the word $[u]^B$ defined recursively by

- $[\epsilon]^B = \epsilon$;
- $[u_{i \rightarrow j}]^B = u_i \odot [u_{i+1 \rightarrow j}]^{B \circ u_i}$ if $u_i \in \mathcal{A}_+^B$;
- $[u_{i \rightarrow j}]^B = [u_{i+1 \rightarrow j}]^B$ if $u_i \in \mathcal{A}_-^B$.

Contraction from B is idempotent: $[[u]^B]^B = [u]^B$. A word u such that $[u]^B = u$ is called a *contracted word*.

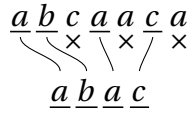
Expanding Expansion is the key feature of this paper. It inserts letters at random in a contracted word *in a way that preserves the stationary distribution of the output of the CFTP algorithm*.

Consider a state $B \in \mathcal{B}$ and a word $u \in \mathcal{A}_\#^*$ ending with $\#$. The expansion of u from B is the random word $\langle u \rangle^B$ defined recursively by

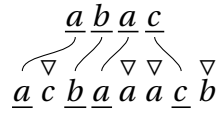
- $\langle \epsilon \rangle^B = \epsilon$;
- If $u_{i \rightarrow j} \neq \epsilon$, let $k = |u_{i \rightarrow j}|_\#$ and A be distributed according to D_{2-k} . Then
 - $\langle u_{i \rightarrow j} \rangle^B = A \odot \langle u_{i \rightarrow j} \rangle^B$ if $A \in \mathcal{A}_-^B$;
 - $\langle u_{i \rightarrow j} \rangle^B = u_i \odot \langle u_{i+1 \rightarrow j} \rangle^{B \circ u_i}$ if $A \in \mathcal{A}_+^B$.

Contraction and expansion are illustrated in Figure 4.2. Notice that both of these operations preserve transitions from B :

$$B \circ u = B \circ [u]^B = B \circ \langle u \rangle^B. \quad (4.1)$$



(a) Contracting

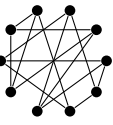


(b) Expanding

Active letters are underlined.

Figure 4.2 – Contracting and Expanding

The geometric CFTP algorithm enhanced with these features is given in Algorithm 7. At each iteration of the outer loop, a contracted prefix V is drawn, to which we attach an expansion of the previous word U . The result is then contracted. The proof that this algorithm does indeed return a sample distributed according to π is not straightforward, and constitutes the main result of this paper, stated in Theorem 12.



3. ORACLE SAMPLING

Algorithm 7 Geometric CFTP with Contraction and Expansion

```

function CONTRACTION-EXPANSION-CFTP(( $\mathcal{S}, \mathcal{A}, D, \cdot$ ), ( $\mathcal{B}, \circ$ ))
   $p \leftarrow \frac{1}{2}$ 
   $U \leftarrow \epsilon$ 
   $B \leftarrow \mathcal{S}$ 
  repeat
     $V \leftarrow \epsilon$ 
    repeat
       $A \leftarrow \text{DRAW}(D_p)$ 
       $V \leftarrow V \circ A$ 
    until  $A = \#$ 
     $U \leftarrow \langle U \rangle^{\mathcal{S}}$  Expand U, as passive letters may become active
     $U \leftarrow V \circ U$ 
     $U \leftarrow [U]^{\mathcal{S}}$  Now contract again
     $B \leftarrow \mathcal{S} \circ U$ 
     $p \leftarrow \frac{1}{2}p$ 
  until  $|B| = 1$ 
  return UNIQUEELEMENTOF( $B$ )
end function
  
```

Rerooting One downside of Algorithm 7 is that the word U is expanded at each iteration. However, in order to have oracle sampling actually reduce the overall complexity of the algorithm, it is important to always work with contracted words. We therefore introduce one last operation, *rerooting*, which combines contraction and expansion so as to never fully expand the word being processed.

Formally, consider two states $B, B' \in \mathcal{B}$. Rerooting a word $u \in \mathcal{A}_{\#}^{\infty}$ ending in $\#$ from B to B' consists in computing

$$\llbracket u \rrbracket^{B \rightsquigarrow B'} = \llbracket \langle u \rangle^B \rrbracket^{B'}$$

so as to never fully expand u . One way to do this is to define $\llbracket u \rrbracket^{B \rightsquigarrow B'}$ recursively as follows:

- $\llbracket \epsilon \rrbracket^{B \rightsquigarrow B'} = \epsilon$;
- If $u_{i \rightarrow j} \neq \epsilon$, let $k = |u_{i \rightarrow j}|_{\#}$ and A be distributed according to

$$D_{2-k}(\cdot \mid \mathcal{A}_+^B \cup \mathcal{A}_+^{B'}).$$

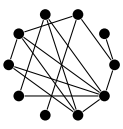
Then

- If $A \in \mathcal{A}_-^B$ (and therefore $A \in \mathcal{A}_+^{B'}$), we expand:

$$\llbracket u_{i \rightarrow j} \rrbracket^{B \rightsquigarrow B'} = A \circ \llbracket u_{i \rightarrow j} \rrbracket^{B \rightsquigarrow B' \circ A};$$

- Otherwise, if $u_i \in \mathcal{A}_+^{B'}$, u_i is active for both chains, so we keep it:

$$\llbracket u_{i \rightarrow j} \rrbracket^{B \rightsquigarrow B'} = u_i \circ \llbracket u_{i+1 \rightarrow j} \rrbracket^{B \circ u_i \rightsquigarrow B' \circ u_i};$$



- Finally, if $u_i \in \mathcal{A}_-^{B'}$, u_i is no longer active, so we contract:

$$\llbracket u_{i \rightarrow j} \rrbracket^{B \rightsquigarrow B'} = \llbracket u_{i+1 \rightarrow j} \rrbracket^{B \circ u_i \rightsquigarrow B'}.$$

Notice that letters passive for both B and B' are not drawn. This would correspond to adding and removing the same letter, as is the case when we expand and contract separately. The conditional distribution ensures this does not happen, and that the complexity is linear in the size of the contracted word (the first case can arise at most once per letter in $\llbracket u \rrbracket^{B \rightsquigarrow B'}$, the other two at most once per letter in u).

Once more, we make the strong assumption that we can draw random variables from distributions such as $D_{2-k}(\cdot \mid \mathcal{A}_+^B \cup \mathcal{A}_+^{B'})$ in constant time. This is not unreasonable, especially in cases such as Gibbs sampling [20], where at most a constant number of letters can change between being active and passive at each step.

The final algorithm is given in Algorithm 8.

Algorithm 8 Oracle CFTP

```

function ORACLE-CFTP( $(\mathcal{S}, \mathcal{A}, D, \cdot), (\mathcal{B}, \circ)$ )
   $p \leftarrow \frac{1}{2}$ 
   $U \leftarrow \epsilon$ 
   $B \leftarrow \mathcal{S}$ 
  repeat
     $V \leftarrow \epsilon$ 
    repeat
       $A \leftarrow \text{DRAW}(D_p(\cdot \mid \mathcal{A}_+^{\mathcal{S} \circ V}))$            V is drawn contracted
       $V \leftarrow V \circ A$ 
    until  $A = \#$ 
     $U \leftarrow V \circ \llbracket U \rrbracket^{\mathcal{S} \rightsquigarrow \mathcal{S} \circ V}$            The rerooting occurs here
     $B \leftarrow \mathcal{S} \circ U$ 
     $p \leftarrow \frac{1}{2}p$ 
  until  $|B| = 1$ 
  return UNIQUEELEMENTOF( $B$ )
end function

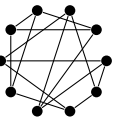
```

3.3 Proof of Correctness

We now prove the correctness of Algorithm 8.

Equivalence with Respect to B One of the key tools of this proof is the use of the following equivalence: given a bounding state $B \in \mathcal{B}$, two random words U and V are considered *equivalent with respect to B* if they have the same distribution and the same contractions from B (i.e. $[U]^B = [V]^B$). Such an equivalence is denoted $U \stackrel{B}{\equiv} V$.

The motivation behind the precise definition of expansion is the following: given a word U , distributed according to R_n , and a bounding state $B \in \mathcal{B}$, con-



3. ORACLE SAMPLING

tracting then expanding U gives an equivalent word with respect to B :

$$\langle [U]^B \rangle_p^B \equiv U.$$

We begin by showing a similar equivalence when U is distributed according to $D_p^{\otimes \mathbb{N}}$, for any fixed p . From there, we extend the result to G_p , and finally to R_n .

The definition of expansion in the previous section was specifically tailored to work with the distribution R_n . To show the first two results, we therefore generalize expansion to operate on words with a fixed rate of occurrences of the letter \sharp , as is the case when sampling from $D_p^{\otimes \mathbb{N}}$.

For any probability p and bounding state B , the p -expansion $\langle u \rangle_p^B$ of $u \in \mathcal{A}_\sharp^\infty$ from B is defined recursively by

- $\langle \epsilon \rangle_p^B = \epsilon$;
- If $u_{i \rightarrow j} \neq \epsilon$, let A be distributed according to D_p . Then
 - $\langle u_{i \rightarrow j} \rangle_p^B = A \odot \langle u_{i \rightarrow j} \rangle_p^B$ if $A \in \mathcal{A}_-^B$;
 - $\langle u_{i \rightarrow j} \rangle_p^B = u_i \odot \langle u_{i+1 \rightarrow j} \rangle_p^{B \circ u_i}$ if $A \in \mathcal{A}_+^B$.

We now show the following result:

Property 12 (Equivalence in the case of $D_p^{\otimes \mathbb{N}}$). *Let p be a fixed probability, $B \in \mathcal{B}$ be a bounding state, and U be distributed according to $D_p^{\otimes \mathbb{N}}$. Contracting then p -expanding from B yields an equivalent word with respect to B :*

$$\langle [U]^B \rangle_p^B \equiv U.$$

Proof. Let $V = \langle [U]^B \rangle_p^B$, $n \in \mathbb{N}$ and $u_{1 \rightarrow n} \in \mathcal{A}_\sharp^n$. We have that

$$\mathbf{P}(V_{1 \rightarrow n} = u_{1 \rightarrow n}) = \prod_{i=1}^n \mathbf{P}(V_i = u_i \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}).$$

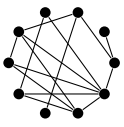
If we can show that, for all $i \in \mathbb{N}$,

$$\mathbf{P}(V_i = u_i \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) = D_p(u_i), \quad (4.2)$$

then we have that, for all $n \in \mathbb{N}$, $V_{1 \rightarrow n}$ is distributed according to $D_p^{\otimes n}$, and we can therefore conclude that V is distributed according to $D_p^{\otimes \mathbb{N}}$.

We now show (4.2) by conditioning on whether V_i is or is not in $\mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}}$:

$$\begin{aligned} & \mathbf{P}(V_i = u_i \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ &= \mathbf{P}(V_i = u_i \mid V_i \in \mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}} \wedge V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ & \quad \times \mathbf{P}(V_i \in \mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}} \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ & \quad + \mathbf{P}(V_i = u_i \mid V_i \in \mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}} \wedge V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ & \quad \times \mathbf{P}(V_i \in \mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}} \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}). \end{aligned}$$



CHAPTER 4. ORACLE SAMPLING

Notice that V_i is in $\mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}}$ if and only if it was added during expansion. By definition, this occurs with probability $D_p(\mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}})$, so we have that

$$\mathbf{P}(V_i \in \mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}} \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) = D_p(\mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}})$$

and

$$\mathbf{P}(V_i \in \mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}} \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) = D_p(\mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}}).$$

Consider the case where V_i is passive. It was inserted during expansion, and its distribution is therefore the restriction of D_p to $\mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}}$:

$$\begin{aligned} \mathbf{P}(V_i = u_i \mid V_i \in \mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}} \wedge V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ = D_p(u_i \mid \mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}}). \end{aligned}$$

Similarly, if V_i is known to be active, then it was already in U and was not removed during contraction. Its distribution was therefore D_p , conditioned on being in $\mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}}$:

$$\begin{aligned} \mathbf{P}(V_i = u_i \mid V_i \in \mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}} \wedge V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ = D_p(u_i \mid \mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}}). \end{aligned}$$

Combining all these results gives that

$$\begin{aligned} \mathbf{P}(V_i = u_i \mid V_{1 \rightarrow i-1} = u_{1 \rightarrow i-1}) \\ = D_p(u_i \mid \mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}}) \times D_p(\mathcal{A}_-^{B \circ u_{1 \rightarrow i-1}}) \\ + D_p(u_i \mid \mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}}) \times D_p(\mathcal{A}_+^{B \circ u_{1 \rightarrow i-1}}) \\ = D_p(u_i). \end{aligned}$$

This concludes the proof that V is distributed according to $D_p^{\otimes \mathbb{N}}$, i.e. has the same distribution as U .

To finish, notice that $[V]^B = \left[\langle [U]^B \rangle_p^B \right]^B$. Since the outer contraction removes all the letters added during expansion (they are by definition passive), and none of the letters present before that ($[U]^B$ contains only active letters), we have that $[V]^B = [U]^B$. We can therefore conclude that $V \stackrel{B}{\equiv} U$. \square

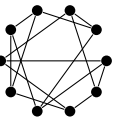
We now extend this result to words of geometric lengths.

Property 13 (Equivalence in the case of G_p). *Let p be a fixed probability, $B \in \mathcal{B}$ be a bounding state, and U be distributed according to G_p . Contracting then p -expanding from B yields an equivalent word with respect to B :*

$$\langle [U]^B \rangle_p^B \stackrel{B}{\equiv} U.$$

Proof. For any word u , denote $[u]_{\sharp}$ the truncation of u after the first occurrence of the letter \sharp . Notice that U can be constructed by taking an infinite random word V distributed according to $D_p^{\otimes \mathbb{N}}$, and setting $U = [V]_{\sharp}$.

Since both contracting and p -expanding operate recursively from left to right, what letters are removed or added before the first \sharp when going from V to $\langle [V]^B \rangle_p^B$



3. ORACLE SAMPLING

is independent of what comes after this \sharp . We can therefore construct a coupling between $\langle [U]^B \rangle_p^B$ and $\langle [V]^B \rangle_p^B$ such that $\langle [U]^B \rangle_p^B = \left[\langle [V]^B \rangle_p^B \right]_{\sharp}$.

Using Property 12, we have that $\langle [V]^B \rangle_p^B$ is distributed according to $D_p^{\otimes \mathbb{N}}$, and $\langle [U]^B \rangle_p^B$ is therefore distributed according to G_p . Finally, using the same argument as in the previous proof,

$$[U]^B = \left[[V]^B \right]_{\sharp} = \left[\left[\langle [V]^B \rangle_p^B \right]_{\sharp} \right] = \left[\langle [U]^B \rangle_p^B \right]_{\sharp},$$

from which we can conclude that $U \stackrel{B}{\equiv} \langle [U]^B \rangle_p^B$. \square

Property 14 (Equivalence in the case of R_n). *Let $n \in \mathbb{N}$, $B \in \mathcal{B}$ be a bounding state, and U be distributed according to R_n . Contracting then expanding from B yields an equivalent word with respect to B :*

$$\langle [U]^B \rangle^B \stackrel{B}{\equiv} U.$$

Proof. We first point out some simple preliminary results. Let $B \in \mathcal{B}$, $n \in \mathbb{N}$ and $(u^{(i)})_{n \geq i \geq 1}$, $(v^{(i)})_{n \geq i \geq 1}$ be two sequences of words in $\mathcal{A}_{\sharp}^{\infty}$.

1. For all m such that $n > m \geq 1$, contraction and concatenation can be interchanged as follows:

$$\left[\bigcirc_{i=n}^m u^{(i)} \right]^B = \bigcirc_{i=n}^m \left[u^{(i)} \right]^{B \circ \bigcirc_{j=n}^{i+1} u^{(j)}}. \quad (4.3)$$

2. If each $u^{(i)}$ ends with its only \sharp , then expansion and concatenation can also be swapped by using 2^{-i} -expansions:

$$\left\langle \bigcirc_{i=n}^1 u^{(i)} \right\rangle^B = \bigcirc_{i=n}^1 \left\langle u^{(i)} \right\rangle_{2^{-i}}^{B \circ \bigcirc_{j=n}^{i+1} u^{(j)}}. \quad (4.4)$$

3. Finally, if, for all $n \geq i \geq 1$,

$$u^{(i)} \stackrel{B \circ \bigcirc_{j=n}^{i+1} u^{(j)}}{\equiv} v^{(i)},$$

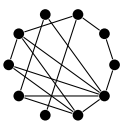
then the concatenations are equivalent:

$$\bigcirc_{i=n}^1 u^{(i)} \stackrel{B}{\equiv} \bigcirc_{i=n}^1 v^{(i)}. \quad (4.5)$$

Consider the unique decomposition

$$U = \bigcirc_{i=n}^1 U^{(i)}$$

such that, for all i , $U^{(i)}$ ends with its only \sharp . By construction of R_n , each $U^{(i)}$ is distributed according to $G_{2^{-i}}$. Using the above results and Property 13, we have that



$$\begin{aligned}
& \langle [U]^B \rangle^B = \left\langle \left[\begin{array}{c} 1 \\ \odot \\ [U^{(i)}] \end{array} \right]_{i=n}^B \right\rangle^B \\
& \text{Eq. (4.3)} \left(\right. \\
& \qquad = \left\langle \begin{array}{c} 1 \\ \odot \\ [U^{(i)}]^{B \circ \odot_{j=n}^{i+1} U^{(j)}} \end{array} \right\rangle^B \\
& \text{Eq. (4.4)} \left(\right. \\
& \qquad = \bigodot_{i=n}^1 \left\langle [U^{(i)}]^{B \circ \odot_{j=n}^{i+1} U^{(j)}} \right\rangle_{2^{-i}}^{B \circ \odot_{k=n}^{i+1} [U^{(k)}]^{B \circ \odot_{j=n}^{k+1} U^{(j)}}} \\
& \text{Eq. (4.3)} \left(\right. \\
& \qquad = \bigodot_{i=n}^1 \left\langle [U^{(i)}]^{B \circ \odot_{j=n}^{i+1} U^{(j)}} \right\rangle_{2^{-i}}^{B \circ [\odot_{k=n}^{i+1} U^{(k)}]^B} \\
& \text{Eq. (4.1)} \left(\right. \\
& \qquad = \bigodot_{i=n}^1 \left\langle [U^{(i)}]^{B \circ \odot_{j=n}^{i+1} U^{(j)}} \right\rangle_{2^{-i}}^{B \circ \odot_{k=n}^{i+1} U^{(k)}} \\
& \text{Prop. 13} \left(\right. \\
& \qquad = \bigodot_{i=n}^1 V^{(i)} \quad \text{where} \quad V^{(i)} \stackrel{B \circ \odot_{j=n}^{i+1} U^{(j)}}{\equiv} U^{(i)} \quad \text{for all } i \\
& \text{Eq. (4.5)} \left(\right. \\
& \qquad \equiv \bigodot_{i=n}^B U^{(i)} \\
& \qquad = U,
\end{aligned}$$

which concludes the proof. \square

Convergence and distribution We now go back to proving the correctness of Oracle CFTP. Recall that the algorithm operates by iteratively computing

$$U^{(n+1)} = [V^{(n+1)}]^\mathcal{S} \circ [U^{(n)}]^\mathcal{S} \rightsquigarrow \mathcal{S} \circ V^{(n+1)}, \quad (4.6)$$

where the $V^{(i)}$ are independent and distributed according to $G_{2^{-i}}$. The algorithm terminates when $U^{(N)}$ is such that $\mathcal{S} \circ U^{(N)}$ is a singleton $\{X\}$, returning X .

Theorem 12. *If there exists $u \in \mathcal{A}^*$ such that $\mathcal{S} \circ u$ is a singleton, then*

1. *Oracle CFTP terminates: N is a.s. finite and of finite expectation; and*
2. *Oracle CFTP is correct: X is distributed according to π .*

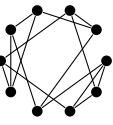
Proof. We begin by proving that Oracle CFTP terminates.

Let $u \in \mathcal{A}^*$ be such that $\mathcal{S} \circ u$ is a singleton. Notice that, for any $n \in \mathbb{N}$, if $V^{(n)}$ begins with u , then $\mathcal{S} \circ U^{(n)}$ is a singleton, and therefore $N \leq n$. The probability that u is a prefix of $V^{(n)}$ is equal to the probability that there are no \sharp in the first $|u|$ letters, i.e. at least $2^{-|u|}$, times the probability of these letters being those of u knowing they are not \sharp , $D^{\otimes |u|}(u)$:

$$\mathbf{P}(u \text{ is a prefix of } V^{(n)}) \geq 2^{-|u|} D^{\otimes |u|}(u).$$

Such an event will necessarily happen, and N is therefore a.s. finite. Its expectation is furthermore upper-bounded by

$$\frac{2^{|u|}}{D^{\otimes |u|}(u)}.$$



3. ORACLE SAMPLING

We now move on to prove the correctness of the algorithm.

A rapid induction shows that the $\langle U^{(n)} \rangle^{\mathcal{S}}$ are distributed according to R_n :

$$\begin{aligned}
\langle U^{(n+1)} \rangle^{\mathcal{S}} &= \left\langle [V^{(n+1)}]^{\mathcal{S}} \odot \left[\langle U^{(n)} \rangle^{\mathcal{S} \circ V^{(n+1)}} \right]^{\mathcal{S}} \right\rangle \\
&= \left\langle [V^{(n+1)}]^{\mathcal{S}} \right\rangle_{2^{-(n+1)}}^{\mathcal{S}} \odot \left\langle \left[\langle U^{(n)} \rangle^{\mathcal{S}} \right]^{\mathcal{S} \circ V^{(n+1)}} \right\rangle^{\mathcal{S} \circ V^{(n+1)}} \\
&\stackrel{\mathcal{S}}{=} \underbrace{V^{(n+1)}}_{G_{2^{-(n+1)}}} \odot \underbrace{\langle U^{(n)} \rangle^{\mathcal{S}}}_{R_n}
\end{aligned}$$

The two elements in the last line being independent, $\langle U^{(n+1)} \rangle^{\mathcal{S}}$ is distributed according to R_{n+1} . Notice however that it is *not* equivalent to $\bigodot_{i=n+1}^1 V^{(i)}$, since $U^{(n)}$ is expanded from \mathcal{S} , not $\mathcal{S} \circ V^{(n+1)}$. More importantly, $\langle U^{(N)} \rangle^{\mathcal{S}}$ itself is not distributed according to R_N , as N is defined in such a way that $\mathcal{S} \circ \langle U^{(N)} \rangle^{\mathcal{S}}$ is necessarily a singleton.

Let $\varepsilon > 0$, and n_ε be an integer such that $\mathbf{P}(N > n_\varepsilon) < \varepsilon$. Since $\langle U^{(n_\varepsilon)} \rangle^{\mathcal{S}}$ is distributed according to R_{n_ε} , removing the \sharp from $\langle U^{(n+1)} \rangle^{\mathcal{S}}$ results in a random word W whose letters are i.i.d. according to D , and whose length is independent of these letters.

Consider a state Y distributed according to π . Since every state is invariant under $x \mapsto x \cdot \sharp$, we have that $Y \cdot \langle U^{(n_\varepsilon)} \rangle^{\mathcal{S}} = Y \cdot W$, which is distributed according to π . We also have that $Y \cdot \langle U^{(n_\varepsilon)} \rangle^{\mathcal{S}} \in \mathcal{S} \circ \langle U^{(n_\varepsilon)} \rangle^{\mathcal{S}} = \mathcal{S} \circ U^{(n_\varepsilon)}$. As such, if ever $\mathcal{S} \circ U^{(n_\varepsilon)}$ is a singleton, then its unique element is distributed according to π .

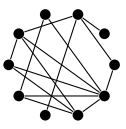
For any $n \in \mathbb{N}$, notice that

$$\begin{aligned}
\mathcal{S} \circ U^{(n+1)} &= \mathcal{S} \circ V^{(n+1)} \circ \langle U^{(n)} \rangle^{\mathcal{S}} \\
&\subseteq \mathcal{S} \circ \langle U^{(n)} \rangle^{\mathcal{S}} && (\mathcal{S} \circ V^{(n+1)} \subseteq \mathcal{S}) \\
&= \mathcal{S} \circ U^{(n)}.
\end{aligned}$$

A direct consequence of this is that, whenever $N \leq n_\varepsilon$, $\mathcal{S} \circ U^{(n_\varepsilon)} \subseteq \mathcal{S} \circ U^{(N)}$. Since $\mathcal{S} \circ U^{(N)}$ is a singleton, and $\mathcal{S} \circ U^{(n_\varepsilon)}$ cannot be the empty set, we even have equality. Since $N \leq n_\varepsilon$ holds with probability at least $1 - \varepsilon$, and since the unique element of $\mathcal{S} \circ U^{(n_\varepsilon)}$ is distributed according to π , we then have that the output of the Oracle CFTP algorithm is correctly distributed with probability at least $1 - \varepsilon$. This being true for all $\varepsilon > 0$, the proof is complete. \square

3.4 Complexity

When considering a specific Markov automaton, it is a common practice to bound the complexity of the CFTP algorithm for that specific usage. For the base CFTP algorithm, Theorem 11 states that $\mathbf{E}[\tau_b] = \mathbf{E}[\tau]$. This makes it possible to bound the complexity of the original algorithm by bounding the *forward* coupling time of the Markov automaton. While this still holds when using bounding chains with constant or geometric increments (so long as the forward coupling time is computed for the transition function \circ rather than \cdot), it is no longer valid when considering oracle sampling.



CHAPTER 4. ORACLE SAMPLING

To better understand the extra difficulty, consider the following toy example: a Markov automaton with $\mathcal{S} = \{x_0, x_1\}$ and $\mathcal{A} = \{s, z\}$, such that

- $x_i \cdot s = x_{1-i}$ (swap states);
- $x_i \cdot z = x_0$ (go to zero);
- $D(z) = \varepsilon$ (z is a rare letter).

The bounding chain is defined as

- $\{x_0, x_1\} \circ s = \{x_0, x_1\}$ (s is passive so long as we have not coupled);
- $\{x_0, x_1\} \circ z = \{x_0\}$ (z means coupling);
- $\{x_i\} \circ s = \{x_{1-i}\}$ (s is active once we have coupled);
- $\{x_i\} \circ z = \{x_0\}$.

With oracle sampling, the forward coupling time is 1: since s is passive, we draw z , and we have automatically coupled. For Oracle CFTP, i.e. the backward coupling time, we have to go back until we can find a z . At this point, s becomes active, and the expansion phase adds approximately ε^{-1} occurrences of s . The complexity is therefore of order ε^{-1} . This illustrates how there can be an arbitrarily large factor between forward and backward coupling times for oracle sampling.

The key observation is the above example is that, whereas the forward coupling time only depends on the quality of contraction for uncoupled states ($|B| > 1$), the backward coupling time must also take into account the quality of contraction after coupling (when B is a singleton).

Property 15. *Let $\gamma(n) \geq \max_{x \in \mathcal{S}} \mathbf{E}[|U_{1 \rightarrow n}|^{x}]$ be an upper bound on the expected contracted size of a random word of length n . The running time Γ_O of the algorithm satisfies*

$$\mathbf{E}[\Gamma_O] = O\left(\left(\mathbf{E}[\tau_O] + \mathbf{E}[\gamma(\tau_B)] + \log(\mathbf{E}[\tau_B])\right) \times \log(\mathbf{E}[\tau_B])\right),$$

where τ_O and τ_B are the forward coupling times of the bounding chains with and without oracle sampling.

Sketch of proof. Notice that the complexity at each iteration n of the algorithm is linear in the number of letters of $V^{(n)}$: each letter is read at most once, as well as being possibly added or removed only once. We can therefore state that the overall complexity Γ_O of the algorithm satisfies

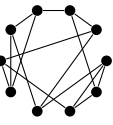
$$\mathbf{E}[\Gamma_O] = O\left(\mathbf{E}\left[\sum_{n=1}^N |V^{(n)}|\right]\right).$$

The expected length of the $V^{(n)}$ is increasing, and it can be shown that

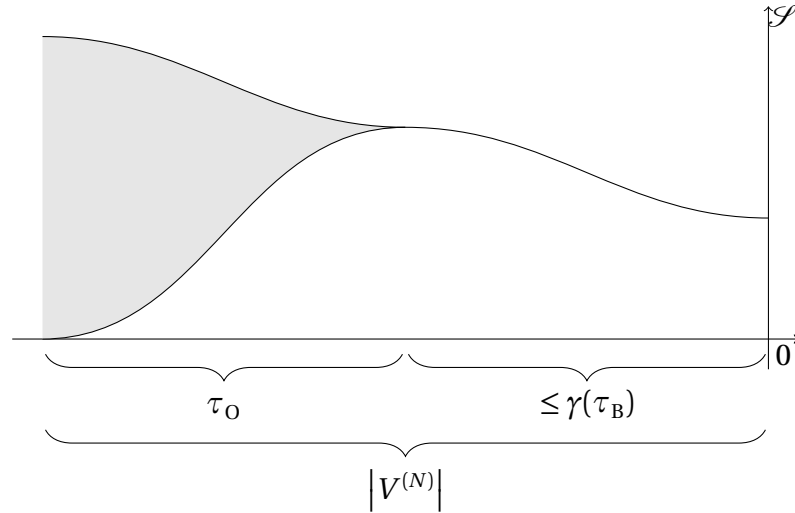
$$\mathbf{E}[N] = \mathbf{E}\left[|V^{(N)}|_{\#}\right] \leq \log(\mathbf{E}[\tau_B]),$$

from which we deduce that

$$\mathbf{E}[\Gamma_O] = O\left(\mathbf{E}\left[|V^{(N)}|\right] \times \log(\mathbf{E}[\tau_B])\right).$$



3. ORACLE SAMPLING



The final coupling word $V^{(N)}$ can be split into a coupling subword (left) and a coupled subword (right).

Figure 4.3 – Bounding Complexity

Consider Figure 4.3. $V^{(N)}$ can be split into a coupling subword, whose length is upper-bounded by τ_O , and a coupled subword, whose length is at most $\gamma(\tau_B)$. Adding the number $|V^{(N)}|_{\#}$ of occurrences of the letter $\#$, which we have omitted so far, we get that

$$\mathbf{E}[|V^{(N)}|] \leq \mathbf{E}[\tau_O] + \mathbf{E}[\gamma(\tau_B)] + \log(\mathbf{E}[\tau_B]).$$

□

Note the implications of the term $\log(\mathbf{E}[\tau_B])$: though this is only an upper bound, it suggests that the complexity of the oracle CFTP algorithm is at least logarithmic in the complexity of the initial bounding chain algorithm.

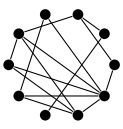
3.5 Variants of Oracle Sampling

Incremental Sampling Though oracle sampling is a convenient model from a theoretical standpoint, we have underlined the issue of sampling active letters when implementing such a method. A more convenient approach, similar to what was introduced in [44], is to discover passive events as they are drawn; this is referred to as *incremental sampling*.

The basis for incremental sampling is to construct the distribution D_i from which V_i is to be drawn using what we have learned from the previous letters:

- If V_{i-1} was active, D_i is reset to D , since the state of the bounding chain has changed and we do not know which letters are active or passive anymore;
- If V_{i-1} was passive, D_i is taken to be D_{i-1} conditioned on not being V_{i-1} , i.e. V_{i-1} is removed from the set of known passive letters.

The new bounding chain $(B_i^I)_{i \in \mathbb{N}}$ defined by $B_i^I = \mathcal{S} \circ V_{1 \rightarrow i}$ for all $i \in \mathbb{N}$ is no longer a Markov chain, but it can nonetheless be used in CFTP to sample from π . Its



coupling time τ_1 is a trade-off between that of the usual bounding chains and that of oracle sampling:

$$\mathbf{E}[\tau_0] \leq \mathbf{E}[\tau_1] \leq \mathbf{E}[\tau].$$

Hybrid Sampling It is possible to combine oracle and incremental sampling to improve performance or simplify implementations. For example, if it is easy to acquire a subset of the passive letters for a given state, then these can be immediately removed from the distribution when the bounding chain changes to that state. On the contrary, if a letter is known to rarely be passive, we can choose to never check whether it is active or not and always keep it in the distribution.

4 Independent Sets

Let $G = (V, E)$ be a simple undirected graph. Recall that a subset I of V is called an independent set if no two vertices in I are connected by an edge, i.e. if

$$\forall x, y \in I, (x, y) \notin E.$$

Let \mathcal{I} be the set of independent sets of G and, for any vertex $v \in V$, denote $N(v)$ the set of neighbors of v , that is to say the $w \in V$ such that $(v, w) \in E$.

We study the performance of the CFTP algorithm with oracle sampling when generating independent sets according to the distribution

$$P_\lambda(I) = \frac{\lambda^{|I|}}{Z_\lambda}, \lambda \in \mathbb{R},$$

where Z_λ is a normalizing constant, often called the *partition function*. We focus on the case where λ is very large. In the spirit of Property 15, we restrict our analysis to the complexity for the forward coupling.

4.1 Sampling algorithms

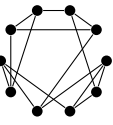
We compare the coupling time our sampling algorithm with oracle sampling with two other approaches described in [28]: Gibbs sampling and the Dyer-Greenhill chain [14].

Gibbs sampling

Let us first define a Gibbs sampler for P_λ . At each iteration, independently draw a vertex v uniformly at random and u uniformly over $[0, 1]$.

- If $u > \frac{\lambda}{\lambda+1}$, then remove v from I if $v \in I$, otherwise do nothing.
- If $0 \leq u \leq \frac{\lambda}{\lambda+1}$, then add v to I if $N(v) \cap I = \emptyset$, otherwise do nothing.

This dynamics allows us to use Monte Carlo and CFTP methods to generate independent sets according to P_λ . The CFTP approach can be greatly improved by using the following bounding chain for the Glauber dynamics defined in [28].



4. INDEPENDENT SETS

Consider a family of independent sets $A \subseteq \mathcal{I}$. Set

$$B = \bigcap_{I \in A} I, \quad D = \left(\bigcup_{I \in A} I \right) \setminus B \quad \text{and} \quad C = \bigcap_{I \in A} (V \setminus I) = V \setminus (B \cup D).$$

Denoting

$$\langle B, D \rangle = \{I \in \mathcal{I} \mid B \subseteq I \subseteq B \cup D\},$$

we have that $A \subseteq \langle B, D \rangle$. In other words, B is the set of vertices common to every independent set in A , C is the set of vertices that are in none of the independent sets of A , and D is the set of vertices that are in some but not all of the independent sets of A . The couples $(\langle B_i, D_i \rangle)_{i \in \mathbb{N}}$ define a bounding chain for the Glauber dynamics $(A_i)_{i \in \mathbb{N}}$. The initial state of this bounding chain is $\langle B_0, D_0 \rangle = \langle \emptyset, V \rangle$.

The Gibbs sampler for the bounding chain is defined as follows: at each iteration, independently draw a vertex v uniformly at random and u uniformly over $[0, 1]$. Suppose the initial state is $\langle B, D \rangle$, and write $B + v$ for $B \cup \{v\}$ and $B - v$ for $B \setminus \{v\}$; the arrival state $\langle B', D' \rangle$ is constructed as follows:

- If $u > \frac{\lambda}{\lambda+1}$, we remove v from all independent sets:

$$B' = B - v \quad D' = D - v \quad C' = C + v$$

- If $0 \leq u \leq \frac{\lambda}{\lambda+1}$, we attempt to add v to the independent sets:

- if $N(v) \cap B \neq \emptyset$, then we do nothing (we necessarily have that $v \in C$);
- if $N(v) \subseteq C$, we can add v to all independent sets:

$$B' = B + v \quad D' = D - v \quad C' = C - v$$

- otherwise, we can only add v to some independent sets:

$$D' = D + v \quad C' = C - v$$

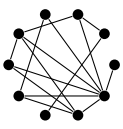
The Dyer-Greenhill scheme

The coupling time of the above bounding chain can be reduced through the Dyer-Greenhill scheme. The main idea is to allow one element to slide to an adjacent feasible vertex. Given $p_s \in [0, 1]$, if, in the Gibbs sampler, an attempt to add v to the independent set I fails due to the presence of a *unique* neighbor u already in I , then with probability p_s , the independent set becomes $I + v - u$. A bounding chain can easily be defined for this new scheme.

Oracle sampling scheme

Now consider oracle sampling for the bounding chain of the Gibbs sampler. For each vertex v , we have two events: adding v to I , denoted a_v , and removing v from I , denoted r_v . The active events are:

- the r_v for which $v \notin C$,
- the a_v for which $v \in C$ and $N(v) \cap B = \emptyset$,



- the a_v for which $v \in D$ and $N(v) \subseteq C$.

Let V_r and V_a be the set of vertices for which removal and addition are respectively active in $\langle B, D \rangle$. For the Gibbs sampler, events are drawn according to the conditional distribution by picking an event uniformly at random in V_z , where $z = a$ with probability

$$\frac{\lambda|V_a|}{\lambda|V_a| + |V_r|},$$

and $z = r$ otherwise.

For a vertex $v \in V$, the fact that a_v and r_v are active is only modified when v , or one of its neighbors, is modified. It is therefore possible to locally update the conditional distribution at each iteration by simply updating the “activeness” of events for the modified vertex and its neighbors. This justifies using oracle sampling rather than incremental sampling in this context.

Note that those three samplers can be adapted to the case of weighted vertices and product-form stationary processes of the form

$$P_\Lambda(I) = \frac{1}{Z_\Lambda} \prod_{v \in I} \lambda(v),$$

where $\Lambda = (\lambda_v)_{v \in V}$ is a weight-vector of the vertices. For the Gibbs sampler, λ is replaced by the $\lambda(v)$ of the selected vertex. The other samplers are modified accordingly.

4.2 Star graph

In this paragraph, we study the graph

$$G_n = (\llbracket 0, n \rrbracket, \{(0, i) \mid i \in \llbracket 1, n \rrbracket\}),$$

called *star graph*. We focus mainly on the performance of the oracle sampling scheme for large values of λ , such as when $\lambda \gg n$. The independents of this graph are

$$\mathcal{I} = \{\{0\}\} \cup \mathcal{P}(\llbracket 1, n \rrbracket).$$

First, we consider the coupling time τ_B of the Glauber dynamics of the bounding chains without oracle sampling, both in the case of the Gibbs sampler and of the Dyer-Greenhill sampler.

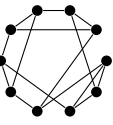
Since at most one vertex is removed from D at each iteration, and the algorithm finishes when $D = \emptyset$, this coupling time is lower bounded by the hitting time of

$$\{\langle B, \{0\} \rangle\} \cup \{\langle B, D \rangle \mid 0 \notin D\}.$$

Furthermore, since no vertex can be added to B so long as D contains both 0 and an element in $\llbracket 1, n \rrbracket$, $B = \emptyset$ until one of those states is reached.

In the case of the Gibbs sampler, if $\lambda > 1$, the expected hitting time of $\langle \emptyset, \{0\} \rangle$ is $O(\lambda^n)$. Furthermore, before reaching this state, the probability of removing 0 from D is exactly $\frac{1}{(\lambda+1)(n+1)}$ at each time step. For n large enough, this gives that

$$\mathbf{E}[\tau_B] \geq (n+1)(\lambda+1).$$



4. INDEPENDENT SETS

For the Dyer-Greenhill sampler, the coupling time τ_B^{DG} is greatly reduced: the hitting time H of $\{\langle B, D \rangle \mid 0 \notin D\}$ has expectation

$$\mathbf{E}[H] = \frac{\lambda + 1}{\lambda} \frac{n + 1}{n} \frac{1}{p_s}.$$

This is due to the fact that the first attempt to swap a vertex other than 0 will immediately remove 0 from D , since it is the only neighbor of the selected vertex.

On the other hand, for the bounding chain to couple, every vertex must be selected at least once for addition or removal. As at each step, the modified vertex is chosen uniformly and independently at random, a coupon collector argument gives that

$$\mathbf{E}[\tau_B^{DG}] \geq n \ln n + O(1).$$

Now, let us consider the (forward) coupling time τ_0 of the coupling chain with oracle sampling. The coupling time is at most the hitting time of $\langle B, \emptyset \rangle$. We have two main steps to consider:

1. The hitting time of $\{\langle B, D \rangle \mid 0 \notin D\}$;
2. From there, the hitting time of $\{\langle B, \emptyset \rangle\}$.

Let us first focus on the hitting time of $\{\langle B, D \rangle \mid 0 \notin D\}$. Construct a birth-and-death process on $\llbracket 0, n \rrbracket$, where state i represents the cardinal of C , supposing 0 has not yet been added to C . In state i , the active events are those that move vertices between D and C . As a consequence, the probabilities $p_{i,i+1}$ and $p_{i+1,i}$ to go respectively from state i to state $i + 1$ and from $i + 1$ to i are given by

$$p_{i,i+1} = \frac{n-i}{n-i+i\lambda} \text{ and } p_{i+1,i} = \frac{(i+1)\lambda}{n-i-1+(i+1)\lambda}. \quad (4.7)$$

Computations show that the stationary distribution π of this birth-and-death process satisfies, for all $i \in \llbracket 0, n \rrbracket$,

$$\pi(i) = \left(\binom{n-1}{i-1} \lambda^{-(i-1)} + \binom{n-1}{i} \lambda^{-i} \right) \pi(0).$$

This yields that $\pi(0) \geq \frac{1}{2} \left(1 + \frac{1}{\lambda}\right)^{-n}$.

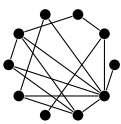
The bounding chain can be bounded by the following process: when in state 0, vertex 0 can be removed with probability $\frac{1}{n+1}$ (all events are active for removal, none for addition). Keeping in mind that $\lambda \geq n$, the expected time τ_1 for reaching a state $\langle B, D \rangle$ where $0 \notin D$ is therefore

$$\mathbf{E}[\tau_1] = \frac{(n+1)}{\pi(0)} \leq 2e(n+1).$$

For the second step, consider the birth-and-death process on $\llbracket 0, n \rrbracket$ where state i represents the sets $\langle B, D \rangle$ for which $|B| = n - i$ and $0 \notin D$. For $i > 0$, i vertices are active for addition and at least $n - i$ are active for removal. The transitions probabilities are exactly the probabilities $p_{i,j}$ defined in Eq. (4.7).

Simple computations show that the hitting time τ_2 of state 0 from state n satisfies

$$\mathbf{E}[\tau_2] = \left(\frac{1+\lambda}{\lambda} \right)^n + n \leq n + e^{\frac{n}{\lambda}} = n + O(1).$$



Finally, note that in state n , vertex 0 is active for addition, and in case this event is generated (which happens with probability $\frac{1}{n+1}$), we have to take into account the return time from the first step ($0 \in D$) to the second step ($0 \notin D$). By the Markov inequality, the probability that state n is visited again before state 0 is at most $\frac{\pi(n)}{\pi(0)} = \lambda^{-n}$.

As a consequence, the expected coupling time satisfies

$$\mathbf{E}[\tau_0] \leq \mathbf{E}[\tau_1] + \mathbf{E}[\tau_2] + O(1) \leq (2e + 1)n + O(1).$$

Notice that the coupling time does not depend on λ and is linear in n . It therefore does better than the other samplers presented above.

4.3 Numerical experiments

We now do an experimental comparison of the three samplers described in Section 4.1 for two models: the star graph, that has been precisely analysed in Paragraph 4.2, and the Barabási-Albert model [1].

Star graph We performed experiments for a star graph with 100 vertices and for different values of λ . For each value of λ and each sampler, 1000 experiments have been performed, and the average number of transitions computed is depicted in Figures 4.4.

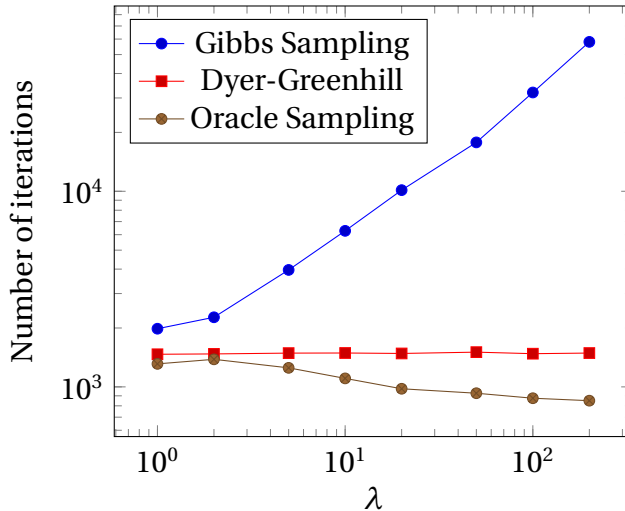
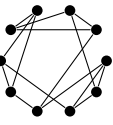


Figure 4.4 – Number of events generated by CFTP algorithms for the star graph with 100 vertices for different values of λ .

The first remark is that both Dyer-Greenhill and oracle sampling samplers outperform the Gibbs sampler. Second, the Dyer-Greenhill sampler seems insensitive to the value of λ , which conforms to the bound $n \ln n + O(1)$ given in Section 4.2. Finally, the oracle sampling scheme is always the most efficient algorithm. It is noticeable that the number of event generated decreases with λ . This can be explained the following way: large independent sets are favored when λ grows. Then, after reaching the independent set $\llbracket 1, n \rrbracket$, whose probability grows with λ , the probability that the next event is active is less than $\frac{1}{1+\lambda}$. As a consequence, many events are skipped.



4. INDEPENDENT SETS

The difference in behavior between the Dyer-Greenhill and oracle sampling samplers is more obvious with the star graph with 1000 vertices, as depicted in Figure 4.5 (100 experiments are run for each value of λ).

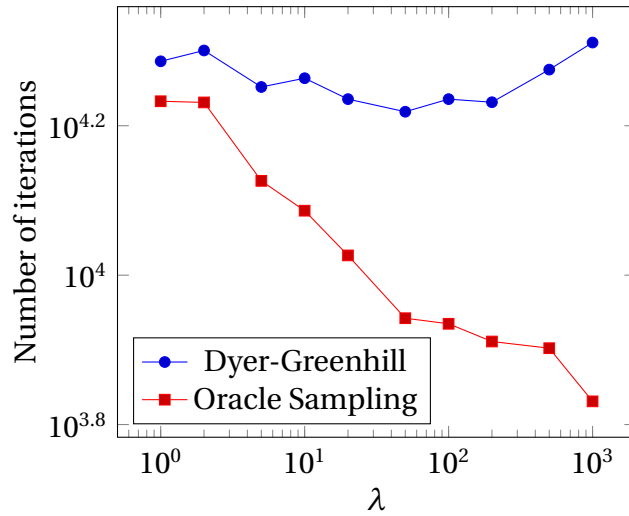


Figure 4.5 – Number of events generated by CFTP algorithms for the star graph with 1000 vertices for different values of λ .

Barabási-Albert model We now generate a random graph with preferential attachment. Start from a clique with 5 vertices and at each step add one new vertex v and two edges (v, w_1) and (v, w_2) , where w_1 and w_2 are chosen at random with probability proportional to their degree. Figure 4.6 compares the average number of events generated for 100 experiments with the three samplers, for graphs with 100 vertices.

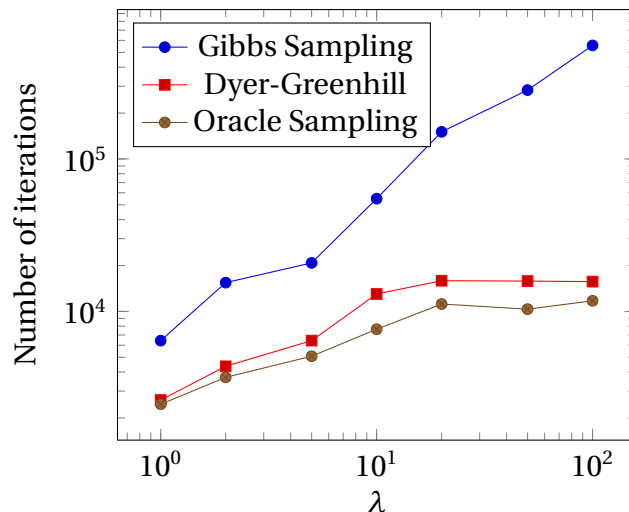
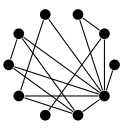


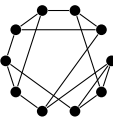
Figure 4.6 – Number of events generated by CFTP algorithms for the Barabási-Albert model with 100 vertices for different values of λ .

Similarly to the star graph, Dyer-Greenhill and oracle sampling samplers outperform the Gibbs sampler, and the oracle sampling sample is sensitively



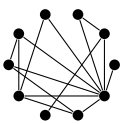
CHAPTER 4. ORACLE SAMPLING

better than the Dyer-Greenhill one. For large values, those two samplers are not sensitive to λ (or slightly improve when λ grows).

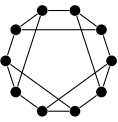


Bibliography

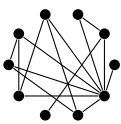
- [1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Modern Phys.*, 74(1):47–97, 2002.
- [2] David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at <http://www.stat.berkeley.edu/~aldous/RWG/book.html>.
- [3] Zeyuan Allen Zhu, Aditya Bhaskara, Silvio Lattanzi, Vahab S. Mirrokni, and Lorenzo Orecchia. Expanders via local edge flips. *CoRR*, abs/1510.07768, 2015.
- [4] S. Asmussen and P.W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Applications of Mathematics. Springer Science, 2007.
- [5] James Aspnes and Gauri Shah. Skip graphs. *ACM Trans. Algorithms*, 3(4), November 2007.
- [6] N. Berestycki. Mixing times of markov chains: Techniques and examples. *Alea-Latin American Journal of Probability and Mathematical Statistics*, 2016.
- [7] Shankar Bhamidi, Guy Bresler, and Allan Sly. Mixing time of exponential random graphs. *The Annals of Applied Probability*, 21(6):2146–2170, 12 2011.
- [8] Bela Bollobas. *Modern Graph Theory*. Springer, 1998.
- [9] J. Borcea, P. Brändén, and T.M. Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009.
- [10] Colin Cooper, Martin Dyer, and Andrew J. Handley. The flip markov chain and a randomising p2p protocol. In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing*, PODC '09, pages 141–150, New York, NY, USA, 2009. ACM.
- [11] Colin Cooper, Ralf Klasing, and Tomasz Radzik. A randomized algorithm for the joining protocol in dynamic distributed networks. Technical Report RR-5376, INRIA, November 2004.
- [12] Victor Debray. Temps de mélange de processus de markov distribués sur des graphes réguliers. Master's thesis, École Polytechnique, 2014.



- [13] D Dubhashi and D. Ranjan. Balls and bins: a study in negative dependence. *Basic Research in Computer Science*, RS-96-25, 1996.
- [14] Martin Dyer and Catherine Greenhill. On Markov chains for independent sets. *J. Algorithms*, 35(1):17–49, 2000.
- [15] Abbas El Gamal and Young-Han Kim. *Network information theory*. Cambridge University Press, Cambridge, 2011.
- [16] P. Erdős and A Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.
- [17] Tomas Feder, Adam Guetz, Milena Mihail, and Amin Saberi. A local switch markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS '06*, pages 69–76, Washington, DC, USA, 2006. IEEE Computer Society.
- [18] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue of random regular graphs. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing, STOC '89*, pages 587–598, New York, NY, USA, 1989. ACM.
- [19] David Galvin and Prasad Tetali. Slow mixing of glauber dynamics for the hard-core model on regular bipartite graphs. *Random Struct. Algorithms*, 28(4):427–443, July 2006.
- [20] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, November 1984.
- [21] George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Symposium on Theoretical Aspects of Computer Science (STACS2011)*, volume 9, pages 57–68, Dortmund, Germany, March 2011.
- [22] George Giakkoupis. Tight bounds for rumor spreading with vertex expansion. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14*, pages 801–815, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.
- [23] W.K. Grassmann. Transient solutions in markovian queueing systems. *Computers & Operations Research*, 4(1):47 – 53, 1977.
- [24] Olle Häggström. *Finite Markov chains and algorithmic applications*, volume 52 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2002.
- [25] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [26] Paul W. Holland and Samuel Leinhardt. Transitivity in structural models of small groups. *Comparative Group Studies*, 2(2):107–124, 1971.



- [27] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561 (electronic), 2006.
- [28] Mark Huber. Perfect sampling using bounding chains. *Ann. Appl. Probab.*, 14(2):734–753, 2004.
- [29] Riko Jacob, Andrea Richa, Christian Scheideler, Stefan Schmid, and Hanjo Täubig. A distributed polylogarithmic time algorithm for self-stabilizing skip graphs. In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing*, PODC '09, pages 131–140, New York, NY, USA, 2009. ACM.
- [30] M. Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, dec 1989.
- [31] Bo Jiang, Nidhi Hegde, Laurent Massoulié, and Don Towsley. How to Optimally Allocate Your Budget of Attention in Social Networks. In *IEEE INFOCOM 2013 - IEEE Conference on Computer Communications*, Torino, Italy, April 2013.
- [32] Libin Jiang, Mathieu Leconte, Jian Ni, R. Srikant, and Jean C. Walrand. Fast mixing of parallel glauber dynamics and low-delay CSMA scheduling. *IEEE Transactions on Information Theory*, 58(10):6541–6555, 2012.
- [33] Johan Jonasson. Mixing times for the interchange process. *Alea-Latin American Journal of Probability and Mathematical Statistics*, 9(2):667–683, 2012.
- [34] David G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *Ann. Math. Statist.*, 24(3):338–354, 09 1953.
- [35] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
- [36] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. Providence, R.I. American Mathematical Society, 2009. With a chapter on coupling from the past by James G. Propp and David B. Wilson.
- [37] T.M. Liggett. Negative correlations and particle systems. *Markov Processes and Related Fields*, (8):547–564, 2002.
- [38] Peter Mahlmann and Christian Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *SPAA*, 2005.
- [39] Peter Mahlmann and Christian Schindelhauer. Distributed random digraph transformations for peer-to-peer networks. In Phillip B. Gibbons and Uzi Vishkin, editors, *SPAA 2006: Proceedings of the 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, Cambridge, Massachusetts, USA, July 30 - August 2, 2006, pages 308–317. ACM, 2006.



- [40] Andrey Andreyevich Markov. Extension of the law of large numbers to dependent quantities. *Izv. Fiz.-Matem. Obsch. Kazan Univ. (2nd Ser)*, 15:135–156, 1906.
- [41] L. Massoulié, A.-M. Kermarrec, and A.J. Ganesh. Network awareness and failure resilience in self-organizing overlay networks. *22nd International Symposium on Reliable Distributed Systems*, 2003.
- [42] Laurent Massoulié, Erwan Le Merrer, Anne-Marie Kermarrec, and Ayalvadi Ganesh. Peer counting and sampling in overlay networks: Random walk methods. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 123–132, New York, NY, USA, 2006. ACM.
- [43] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087, 1953.
- [44] Furcy Pin, Ana Bušić, and Bruno Gaujal. Acceleration of perfect sampling by skipping events. In *Proceedings of the 5th International Conference on Performance Evaluation Methodologies and Tools (Valuetools)*, 2011.
- [45] James Gary Propp and David Bruce Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9(1-2):223–252, 1996.
- [46] Thomas Sauerwald and He Sun. Tight bounds for randomized load balancing on arbitrary network topologies. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 341–350, Washington, DC, USA, 2012. IEEE Computer Society.
- [47] Jason Schweinsberg. An $o(n^2)$ bound for the relaxation time of a markov chain on cladograms. *Random Struct. Algorithms*, 20(1):59–70, jan 2002.
- [48] Devavrat Shah and Jinwoo Shin. Randomized scheduling algorithm for queueing networks. *Ann. Appl. Probab.*, 22(1):128–171, 2012.
- [49] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393(6684):440–442, June 1998.

Abstract

This thesis focuses on the rapid mixing of graph-related Markov chains.

The main contribution concerns graphs with local edge dynamics, in which the topology of a graph evolves as edges slide along one another. We propose a classification of existing models of dynamic graphs, and illustrate how evolving along a changing structure improves the convergence rate. This is complemented by a proof of the rapid mixing time for one such dynamic. As part of this proof, we introduce the partial expansion of a graph. This notion allows us to track the progression of the dynamic, from a state with poor expansion to good expansion at equilibrium.

The end of the thesis proposes an improvement of the Propp and Wilson perfect sampling technique. We introduce oracle sampling, a method inspired by importance sampling that reduces the overall complexity of the Propp and Wilson algorithm. We provide a proof of correctness, and study the performance of this method when sampling independent sets from certain graphs.

Keywords

Social networks
Dynamic graphs
Markov chains

Résumé

Cette thèse porte sur la rapidité du temps de mélange de chaînes de Markov sur des graphes.

La contribution principale concerne les graphes avec des dynamiques locales sur les arêtes, la topologie du graphe évoluant au fur et à mesure que les arêtes glissent les unes le long des autres. Nous proposons une classification des différents modèles existants de graphes dynamiques, tout en illustrant l'importance des transitions le long d'une structure mouvante pour améliorer la vitesse de convergence. Cette étude est complétée par la preuve, pour l'une de ces dynamiques, d'un temps de mélange rapide. Nous définissons notamment l'expansion partielle d'un graphe. Celle-ci permet de suivre l'avancement de la dynamique, partant d'un état de faible expansion, jusqu'à obtention d'une bonne expansion à l'équilibre.

La fin de cette thèse porte sur une amélioration de l'algorithme de simulation parfaite de Propp et Wilson. Nous introduisant un oracle pour les transitions, inspiré de l'échantillonnage préférentiel, qui permet de réduire la complexité de l'algorithme. Nous fournissons une preuve de correction, ainsi qu'une étude de l'impact de cette méthode sur la vitesse d'échantillonnage d'ensembles indépendants pour certains graphes.

Mots Clefs

Réseaux sociaux
Graphes dynamiques
Chaînes de Markov