UNIVERSITY OF LEEDS

This is a repository copy of *The flip Markov chain for connected regular graphs*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/133705/

Version: Accepted Version

Article:
Cooper, C, Dyer, M, Greenhill, C et al. (1 more author) (2019) The flip Markov chain for connected regular graphs. Discrete Applied Mathematics, 254. pp. 56-79. ISSN 0166-218X

https://doi.org/10.1016/j.dam.2018.06.019

(c) 2018 Elsevier B.V. All rights reserved. Licensed under the Creative Commons Attribution-Non Commercial No Derivatives 4.0 International License (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Reuse
This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

The flip Markov chain for connected regular graphs *†

Colin Cooper[‡]
Department of Computer Science
King's College London
London WC2R 2LS, U.K.
colin.cooper@kcl.ac.uk

Martin Dyer[‡]
School of Computing
University of Leeds
Leeds LS2 9JT, U.K.
m.e.dyer@leeds.ac.uk

Catherine Greenhill[§]
School of Mathematics and Statistics
UNSW Australia
Sydney, NSW 2052, Australia
c.greenhill@unsw.edu.au

Andrew Handley[¶]
School of Computing
University of Leeds
Leeds LS2 9JT, U.K.
jobriath@gmail.com

13 June 2018

Abstract

Mahlmann and Schindelhauer (2005) defined a Markov chain which they called k-Flipper, and showed that it is irreducible on the set of all connected regular graphs of a given degree (at least 3). We study the 1-Flipper chain, which we call the flip chain, and prove that the flip chain converges rapidly to the uniform distribution over connected $2r$-regular graphs with n vertices, where $n \geq 8$ and $r = r(n) \geq 2$. Formally, we prove that the distribution of the flip chain will be within ε of uniform in total variation distance after $\text{poly}(n, r, \log(\varepsilon^{-1}))$ steps. This polynomial upper bound on the mixing time is given explicitly, and improves markedly on a previous bound given by Feder et al. (2006). We achieve this improvement by using a direct two-stage canonical path construction, which we define in a general setting.

This work has applications to decentralised networks based on random regular connected graphs of even degree, as a self-stabilising protocol in which nodes spontaneously perform random flips in order to repair the network.

Keywords: Markov chain; graph; connected graph; regular graph

---

*An earlier version of this paper appeared as an extended abstract in PODC 2009 [6].

†This research was partly performed while the first three authors were visiting the Simons Institute for the Theory of Computing.

‡Supported by EPSRC Research Grant EP/M004953/1.

§Supported by the Australian Research Council Discovery Project DP140101519.

¶Supported by EPSRC Research Grant EP/D00232X/1.

1 Introduction

Markov chains which walk on the set of random regular graphs have been well studied. The switch chain is a very natural Markov chain for sampling random regular graphs. A transition of the switch chain is called a *switch*, in which two edges are deleted and replaced with two other edges, without changing the degree of any vertex. The switch chain was studied by Kannan et al. [14] for bipartite graphs, and in [4] for regular graphs. Although a switch changes only a constant number of edges per transition, it is not a local operation since these edges may be anywhere in the graph.

Random regular graphs have several properties which make them good candidates for communications networks: with high probability, a random Δ-regular graph has logarithmic diameter and high connectivity, if $\Delta \geq 3$. (See, for example, [20].) Bourassa and Holt [3] proposed a protocol for a decentralised communications network based on random regular graphs of even degree. Since then, Markov chains which sample regular graphs using local operations have been applied to give *self-stabilising* or *healing* protocols for decentralised communications networks which have a regular topology. To help the network to recover from degradation in performance due to arrivals and departures, clients in a decentralised network can spontaneously perform local transformations to re-randomize the network, thereby recovering desirable properties.

Mahlmann and Schindelhauer [16] observed that the switch operation is unsuitable for this purpose, since a badly-chosen switch may disconnect a connected network. Furthermore, since the switch move is non-local there are implementation issues involved in choosing two random edges in a decentralised network. (If the network is an expander then this problem can be solved by performing two short random walks in the network and taking the final edge of each for the next operation [3].)

To overcome these problems, Mahlmann and Schindelhauer proposed an alternative operation which they called a k-Flipper. When $k = 1$, the 1-Flipper (which we call the *flip*) is simply a restricted switch operation, in which the two edges to be switched must be at distance one apart in the graph. Figure 1 illustrates a switch and a flip.

Figure 1: Left: a switch operation, denoted by $\Rightarrow_S$. Right: a flip, denoted by $\Rightarrow_F$.

By design, the flip operation preserves the degree of all vertices and preserves connectivity. Indeed, it is the smallest such randomising operation: no edge exchange on fewer than four vertices preserves the degree distribution, and the only shallower tree than the 3-path is the 3-star, which has no degree-preserving edge exchange. In order to use flips as a self-stabilising protocol, each peer waits for a random time period using a Poisson clock, after which it instigates a new flip. This involves communication only with peers at distance at most two away in the network.

We will show that flips have asymptotic behaviour comparable to switches, in that both operations randomise a network in polynomial time. Simulations also suggest that they operate equally fast, up to a constant factor (see [11, Section 2.3.6]). A discussion of a distributed implementation of the flip operation is given in [6].

Our approach will be to relate the mixing time of the flip chain $\mathcal{M}_{\mathrm{F}}$ to an intermediate chain $\mathcal{M}_{\mathrm{SC}}$, which is the switch chain restricted to connected graphs. In turn, the mixing time of the connected switch chain $\mathcal{M}_{\mathrm{SC}}$ will be related to the mixing time of the (standard) switch chain $\mathcal{M}_{\mathrm{S}}$, using the analysis from [4]. Each of these two steps will be performed using a technique which we call "two-stage direct canonical path construction", described in Section 2. Specifically, we prove the following result. This is a corrected version of the bound given in [6], which failed on certain rare graphs.

Theorem 1.1. *For each $n \geq 8$ let $\Delta = \Delta(n) \geq 4$ be a positive even integer such that $n \geq \Delta + 1$. The mixing time of the flip Markov chain $\mathcal{M}_{\mathrm{F}}$ on the set of Δ-regular graphs with n vertices is at most*

$$480\,\Delta^{35}\,n^{15}\,\left(\Delta n \log(\Delta n) + \log(\varepsilon^{-1})\right),$$

of which the two-stage direct construction is responsible for a factor of $480\,\Delta^{12} n^7$.

Section 3 relates the flip chain and the connected switch chain, while Section 4 relates the connected switch chain and the switch chain. The proof of Theorem 1.1 can be found at the end of Section 4. The rest of this section contains some background material, some helpful graph theory results, precise definitions of the flip and switch Markov chains, and some definitions regarding canonical paths.

We remark that a "one-stage direct analysis" of the flip chain seems difficult. Furthermore, the multicommodity flow defined for the switch chain in [4] does not apply directly to the connected switch chain, since the flow may pass through disconnected graphs, even when both end-states are connected. It may be possible to circumvent this problem by using an alternative multicommodity flow argument, but we do not explore that option here.

1.1 Notation, terminology and history

We will use the notation $(ab, cd \Rightarrow ac, bd)$ to denote the switch operation as illustrated in Figure 1.

The switch chain $\mathcal{M}_{\mathrm{S}}$ applies a random switch at each step, and the flip chain $\mathcal{M}_{\mathrm{F}}$ applies a random flip at each step. (We define both chains formally in Section 1.4.) The state space Ω_S for the switch chain is the set of all Δ-regular graphs on the vertex set $[n] = \{1, 2, \ldots, n\}$, while the state space Ω_F for the flip chain is the set of all connected graphs in Ω_S.

The switch chain is known to be ergodic, and has uniform stationary distribution on Ω_S. Mahlmann and Schindelhauer [16] showed that the flip chain is ergodic on Ω_F. They also proved that the transitions of the flip chain are symmetric, so that the stationary distribution of the flip chain is uniform on Ω_F. (We use a slightly different transition procedure, described in Section 1.4, which is also symmetric.)

The mixing time of a Markov chain is a measure of how many steps are required before the Markov chain has a distribution which is close to stationary. Kannan et al. [14] considered the mixing time of the switch Markov chain on "near regular" bipartite graphs. Cooper et al. [4] gave a polynomial bound on the mixing time $\tau_{\mathrm{CDG}}(\varepsilon)$ of the switch chain for all regular simple graphs. The result of [4] allows the degree $\Delta = \Delta(n)$ to grow arbitrarily with n: it is not restricted to regular graphs of constant degree.

In Section 3 we extend the result of [4] from switches to flips, using a method which we call *two-stage direct canonical path construction.* We restrict our attention to even degree, since this allows the network to have any number of vertices (even or odd). So for the rest of the paper we assume that $\Delta = 2r$ for some $r \geq 2$, unless otherwise stated. It happens that even-degree graphs have some other desirable properties, as will we see in Section 1.2, which will simplify parts of our argument. However, we see no reason why the flip chain would fail to be rapidly mixing for odd degrees.

Mahlmann and Schindelhauer [16] do not provide a bound for the mixing time of $\mathcal{M}_{\mathrm{F}}$, though they do show [16, Lemma 9] that a similar chain with the flip edges at a distance of $\Theta(\Delta^2 n^2 \log \varepsilon^{-1})$ apart will give an ε-approximate expander in time $\mathrm{O}(\Delta n)$. However, these moves are highly non-local and therefore are unsuitable for the application to self-stabilisation of decentralised networks.

An upper bound for the mixing time of $\mathcal{M}_{\mathrm{F}}$ was given by Feder et al. [9] using a comparison argument with the switch Markov chain, $\mathcal{M}_{\mathrm{S}}$. Applying a comparison argument to relate these chains is a difficult task, since a switch may disconnect a connected graph and a flip cannot. Feder et al. [9] solve this difficulty by embedding $\mathcal{M}_{\mathrm{S}}$ into a restricted switch chain which only walks on connected graphs. They prove that this can be done in such a way that a path in $\mathcal{M}_{\mathrm{S}}$ is only polynomially lengthened. Their result relaxes the bound in [4] by $\mathrm{O}(\Delta^{41} n^{45})$. By employing a two-stage direct construction, we give a much tighter result, relaxing the bound on $\tau_{\mathrm{CDG}}(\varepsilon)$ by a factor of $\mathrm{O}(r^{12} n^7)$ in the case of $2r$-regular graphs with $r \geq 2$.

Recently, Allen-Zhu et al. [1] proved a result related to the work of Mahlmann and Schindelhauer [16, Lemma 9]. They proved that for $\Delta = \Omega(\log n)$, after at most $O(n^2 \Delta^2 \sqrt{\log n})$ steps of the flip chain, the current graph will be an algebraic expander in the sense that the eigenvalue gap of the adjacency matrix is $\Omega(\Delta)$. The proof is based on tracking the change of the graph Laplacian using a potential function rather than analysing the flip chain directly, and does not address the distribution of the output graph, or the case $\Delta = o(\log n)$, which may be the most convenient in practical applications.

Indeed, it may be possible to reduce the mixing time bound by combining our analysis with the results of [1]. If $\Delta = o(\log n)$ then the bound from Theorem 1.1, is $O(n^{16} \operatorname{poly}(\log n))$. Otherwise, when $\Delta = \Omega(\log n)$, the first $O(n^2 \Delta^2 \sqrt{\log n})$ steps of the flip chain may be viewed as a preprocessing phase. As shown by [1], at the end of this preprocessing phase the current graph is an expander with high probability, and hence has logarithmic diameter. The diameter appears in our analysis, for example in the proof of Lemma 3.1 below, and in the analysis of the switch chain [4]. Therefore, it is possible that this preprocessing phase could result in an improvement in the mixing time bound of at least one factor of the order $\frac{\log n}{n}$. To achieve this improvement, it appears necessary to show that a canonical path between two expanders only visits graphs with logarithmic diameter. We do not pursue this approach here.

1.2 Graph-theoretical preliminaries

For further details see, for example, Diestel [8]. Unless otherwise stated, G denotes a simple graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$. The set of neighbours in G of a vertex v is denoted by $N_G(v)$.

A vertex set $S \subseteq V$ has *edge boundary* $\partial_E S = \{uv \,|\, u \in S, v \notin S\}$, which contains the edges crossing from inside to outside of S.

A graph G is 2-edge-connected if it is connected and every edge of G lies in a cycle. More generally, an edge-cut in G is a set $F \subseteq E$ of edges of G such that the graph $G - F$ (which is G with the edges in F deleted) is disconnected. If $|F| = k$ then F is a k-edge-cut. If vertices v, w belong to distinct components of $G - F$ then we say that F *separates* v and w in G.

A *cut vertex* in a graph G is a vertex v such that deleting v disconnects the graph.

If p is a path from v to w then we say that p is a (v, w)-path. Further, if $p = v \cdots a \cdots b \cdots w$ then we write $p[a : b]$ to denote the (inclusive) subpath of p with endvertices a and b.

We make use of the following structural properties of $2r$-regular graphs with $r \geq 2$.

Lemma 1.2. *Let $r \geq 2$ be an integer and let $G = (V, E)$ be a $2r$-regular graph on $n \geq \max\{8,\, 2r + 1\}$ vertices. Then G satisfies the following properties:*

(i) *For any $S \subseteq V$, the edge boundary $\partial_E S$ of S is even.*

(ii) *Every edge of G lies in a cycle; that is, every connected component of G is 2-edge-connected.*

(iii) *G has fewer than $n/(2r)$ connected components.*

(iv) *Let u, v be distinct vertices in the same connected component of G. The number of 2-edge-cuts in G which separate u and v is at most $n^2/(15r^2)$.*

Proof. For (i), we argue by contradiction. Assume that there is an odd number $k = 2k' + 1$ of edges incident upon some set S. Let $G[S]$ be the subgraph induced by S. Let $|S| = s$ and suppose that $G[S]$ has m edges. Denote the sum of degrees of the vertices in S by $d_G(S)$. Clearly $d_G(S) = 2rs$. Exactly k edges incident with S have precisely one endvertex in S, so

$$2m = d_{G[S]}(S) = 2rs - k = 2rs - 2k' - 1 = 2(rs - k') - 1,$$

but no integer m satisfies this equation. Hence our assumption is false, proving (i).

For (ii), it suffices to observe that every component of G is Eulerian, as all vertices have even degree. Next, (iii) follows since the smallest $2r$-regular graph is K_{2r+1}, and G can have at most $n/(2r + 1) < n/(2r)$ components this small. The proof of (iv) is deferred to the Appendix, as it is somewhat lengthy. □

1.3 Canonical paths

We now introduce some Markov chain terminology and describe the canonical path method. For further details, see for example Jerrum [12] or Sinclair [17].

Let $\mathcal{M}$ be a Markov chain with finite state space Ω and transition matrix P. We suppose that $\mathcal{M}$ is ergodic and has unique stationary distribution π. The graph underlying the Markov chain $\mathcal{M}$ is given by $\mathcal{G} = (\Omega, E(\mathcal{M}))$ where the edge set of $\mathcal{G}$ corresponds to (non-loop) transitions of $\mathcal{M}$; specifically,

$$E(\mathcal{M}) = \{xy \mid x, y \in \Omega,\ x \neq y,\ P(x,y) > 0\}$$

We emphasise that $\mathcal{G}$ has no loops.

Let σ, ρ be two probablity distributions defined on a finite set Ω. The *total variation distance* between σ and ρ, denoted $\mathrm{d_{TV}}(\sigma, \rho)$, is defined by

$$\mathrm{d_{TV}}(\sigma, \rho) = \tfrac{1}{2} \sum_{x \in \Omega} |\sigma(x) - \rho(x)|.$$

The *mixing time* $\tau(\varepsilon)$ of a Markov chain is given by

$$\tau(\varepsilon) = \max_{x \in X} \min \{T \geq 0 \mid \mathrm{d_{TV}}(P_x^t, \pi) \leq \varepsilon \text{ for all } t \geq T\},$$

where P_x^t is the distribution of the random state X_t of the Markov chain after t steps with initial state x. Let the eigenvalues of P be

$$1 = \lambda_0 > \lambda_1 \geq \lambda_2 \cdots \geq \lambda_{N-1} \geq -1.$$

Then the mixing time satisfies

$$\tau(\varepsilon) \leq (1 - \lambda_*)^{-1} \left(\ln(1/\pi^*) + \ln(\varepsilon^{-1})\right) \tag{1.1}$$

where $\lambda_* = \max\{\lambda_1, |\lambda_{N-1}|\}$ and $\pi^* = \min_{x \in \Omega} \pi(x)$ is the smallest stationary probability. (See [17, Proposition 1].)

The *canonical path method*, introduced by Jerrum and Sinclair [13], gives a method for bounding λ_1. Given any two states $x, y \in \Omega$, a "canonical" directed path γ_{xy} is defined from x to y in the graph $\mathcal{G}$ underlying the Markov chain, so that each step in the path corresponds to a transition of the chain. The *congestion* $\bar{\rho}(\Gamma)$ of the set $\Gamma = \{\gamma_{xy} \mid x, y \in \Omega\}$ of canonical paths is given by

$$\bar{\rho}(\Gamma) = \max_{e \in E(\mathcal{M})} Q(e)^{-1} \sum_{\substack{x,y \in \Omega \\ e \in \gamma_{xy}}} \pi(x)\pi(y)|\gamma_{xy}|$$

where $Q(e) = \pi(u)\, P(u,v)$ when $e = uv \in E(\mathcal{M})$. If the congestion along each transition is low then the chain should mix rapidly. This can be made precise as follows: if $\lambda^* = \lambda_1$ then

$$(1 - \lambda_1)^{-1} \leq\ \bar{\rho}(\Gamma). \tag{1.2}$$

(See [17, Theorem 5] or [12] for more details.) In many applications the chain $\mathcal{M}$ is made *lazy* by replacing its transition matrix P by $(I + P)/2$: this ensures that the chain has no negative eigenvalues and hence that $\lambda_* = \lambda_1$. However, in many cases it is easy to see that $(1+\lambda_{N-1})^{-1}$ is smaller than the best-known upper bound on $(1-\lambda_1)^{-1}$, such as that provided by the canonical path method (1.2). In this situation, combining (1.1) and (1.2) gives an upper bound on the mixing time without the need to make the chain lazy.

The *multicommodity flow* method is a generalisation of the canonical path method. Rather than send $\pi(x)\pi(y)$ units down a single "canonical" path from x to y, a set $\mathcal{P}_{xy}$ of paths is defined, and the flow from x to y is divided among them. The congestion can be defined similarly, and low congestion leads to rapid mixing: see Sinclair [17].

Cooper et al. [4] gave a multicommodity flow argument to bound the mixing time of the switch chain $\mathcal{M}_{\mathrm{S}}$. For every pair of distinct states x, y, a set of paths from x to y was defined, one for each vector $(\phi_v)_{v\in[n]}$, where ϕ_v is a "pairing" of the edges of the symmetric difference of x and y incident with the vertex v. Then the flow between x and y was shared equally among all these paths. In our analysis below, we assume that a pairing has been *fixed* for each distinct pair of states x, y, leading to one canonical path γ_{xy} between them. We will work with the set of these canonical paths for the switch chain, $\Gamma_S = \{\gamma_{xy}\}$. Since our congestion bounds will be independent on the choice of pairing, the same bound would hold if the original multicommodity flow from [4] was used.

1.4 The switch and flip Markov chains

Given $\Delta \geq 3$ and $n \geq \Delta + 1$, the *switch chain* $\mathcal{M}_{\mathrm{S}}$ has state space Ω_S given by the set of all d-regular graphs on n vertices. The flip chain $\mathcal{M}_{\mathrm{F}}$ has state space Ω_F given by the set of all connected Δ-regular graphs on n vertices. We restrict our attention to regular graphs of even degree, setting $\Delta = 2r$ for some integer $r \geq 2$.

Let P_S (respectively, P_F) denote the transition matrix of the switch (respectively, flip) Markov chain. The stationary distribution of the switch (respectively, flip) chain is denoted by π_S (respectively, π_F). The graph underlying the Markov chain will be denoted by $\mathcal{G}_S = (\Omega_S, E(\mathcal{M}_{\mathrm{S}}))$ and $\mathcal{G}_F = (\Omega_F, E(\mathcal{M}_{\mathrm{F}}))$, respectively.

The transition procedure for the switch Markov chains is given in Figure 2.

```
From G ∈ Ω_S do
    choose non-adjacent distinct edges ab, cd, u.a.r.,
    choose a perfect matching M of {a, b, c, d} u.a.r.,
    if M ∩ E = ∅ then
        delete the edges ab, cd and add the edges of M,
    otherwise
        do nothing
    end if;
end;
```

Figure 2: The switch Markov chain $\mathcal{M}_{\mathrm{S}}$.

The switch chain is irreducible [15, 18]. Furthermore, as $P_S(x, x) \geq 1/3$ for all $x \in \Omega_S$, the switch chain is aperiodic, and hence ergodic. Using an approach of Diaconis and Saloff-Coste [7, p. 702], this also implies that the smallest eigenvalue λ_{N-1} of $\mathcal{M}_{\mathrm{S}}$ satisfies

$$(1 + \lambda_{N-1})^{-1} \leq \tfrac{1}{2} \max_{x\in\Omega_S} P(x, x)^{-1} \leq 3/2.$$

Since the upper bound on $(1-\lambda_1)^{-1}$ proved in [4] is (much) larger than constant, this proves that we can apply the canonical path method to the switch chain without making the switch chain lazy.

If x, y are distinct elements of Ω_S which differ by a switch then $P_S(x,y) = 1/(3a_{n,2r})$ where (see [4])

$$a_{n,2r} = \binom{rn}{2} - n\binom{2r}{2}, \tag{1.3}$$

Hence the transition matrix P_S is symmetric and the stationary distribution of the switch chain is uniform over Ω_S.

Figure 3 gives the transition procedure of the flip chain. Here we start from a randomly chosen vertex a and perform a random walk from a of length 3, allowing backtracking. There are $(2r)^3 n$ choices for the resulting walk (a, b, c, d), and we choose one of these uniformly at random.

```
From G ∈ Ω_F do
    choose a vertex a ∈ [n] u.a.r.,
    choose a walk (a, b, c, d) of length 3 from a, u.a.r.,
    if |{a, b, c, d}| = 4 and E ∩ {ac, bd} = ∅ then
        delete the edges ab, cd and add the edges ac, bd,
    otherwise
        do nothing
    end if;
end;
```

Figure 3: The flip Markov chain $\mathcal{M}_\mathrm{F}$.

Mahlmann and Schindelhauer proved that the flip chain is irreducible on Ω_F [16]. For all states $x \in \Omega_F$ we have $P(x,x) \geq 1/(2r)$, since in particular the proposed transition is rejected when $c = a$. Therefore the flip chain is aperiodic, and hence ergodic. Again, by [7, p. 702], this implies that the smallest eigenvalue λ_{N-1} of $\mathcal{M}_\mathrm{F}$ satisfies $(1+\lambda_{N-1})^{-1} \leq r$. This in turn allows us to apply the canonical path method without making the chain lazy, since the upper bound we obtain on $(1-\lambda_1)^{-1}$ will be (much) bigger than r.

Now suppose that x, y are distinct elements of Ω_F which differ by a flip, and that this flip replaces edges ab, cd with ac, bd. Then if exactly one of the edges bc, ad is present in x, say bc, then the transition from x to y takes place if and only if the chosen 3-walk is (a, b, c, d) or (d, c, b, a). Hence in this case

$$P_F(x,y) = \frac{2}{(2r)^3 n} = P_F(y,x).$$

If both of the edges bc and ad are present then either of them could take the role of the "hub edge" of the flip, and

$$P_F(x,y) = \frac{4}{(2r)^3 n} = P_F(y,x).$$

Hence P_F is a symmetric matrix and the stationary distribution of the flip chain is uniform, as claimed. (Mahlmann and Schindelhauer use a slightly different transition procedure, and must argue about the number of triangles which contain a given edge in order to show that their transitions are symmetric: see [16, Lemma 3].)

2 Two-stage direct canonical path construction

Suppose $\mathcal{M}$ is an ergodic Markov chain whose mixing time we would like to study, and $\mathcal{M}'$ is a chain for which a canonical path set Γ' has been defined, with congestion bounded by $\bar{\rho}(\Gamma')$. Let P, π and Ω (respectively, their primed counterparts) be the transition matrix, stationary distribution and state space of $\mathcal{M}$ (respectively, $\mathcal{M}'$). Guruswami [10] proved that under mild conditions on $\mathcal{M}'$, a bound on the mixing time of $\mathcal{M}'$ implies the existence of an appropriate set Γ' with bounded congestion.

We wish to "simulate" transition in $\mathcal{M}'$ using sequences of transitions in $\mathcal{M}$, and showing that not too many transitions in $\mathcal{M}$ can be involved in any one such simulation. Together with $\bar{\rho}(\Gamma')$ this results on a bound of the congestion of Γ, and hence on the mixing time of $\mathcal{M}$. Here we assume that $\Omega \subseteq \Omega'$.

If Ω is a *strict* subset of Ω' then canonical paths in Γ' that begin and end in Ω may travel through states in $\Omega' \setminus \Omega$. (This will be the case in Section 4.) To deal with this, we must define a surjection $h\colon \Omega' \longrightarrow \Omega$ such that $h(u) = u$ for any $u \in \Omega$, and such that the preimage of any $y \in \Omega$ is not too large. This means that each state in Ω "stands in" for only a few states in Ω'. If $\Omega = \Omega'$ then h is the identity map, by definition.

Suppose that $\gamma'_{wz} \in \Gamma'$ is a canonical path from w to z in $\mathcal{G}'$. Write

$$\gamma'_{wz} = (Z_0, Z_1, \dots, Z_j,\ Z_{j+1}, \dots, Z_\ell),$$

with $Z_0 = w, Z_\ell = z$ and $(Z_j, Z_{j+1}) \in E(\mathcal{M}')$ for all $j \in [\ell - 1]$. We now define the *two-stage direct canonical path* $\gamma_{w,z}$ with respect to the fixed surjection $h : \Omega' \to \Omega$.

For each $(Z_j, Z_{j+1}) \in \gamma'_{wz}$, construct a path

$$\sigma_{Z_j Z_{j+1}} = (X_0, X_1, \dots, X_i, X_{i+1}, \dots, X_\kappa),$$

with $X_0 = h(Z_j)$ and $X_\kappa = h(Z_{j+1})$, such that $X_i \in \Omega$ for $i = 0, \dots, \kappa$ and $(X_i, X_{i+1}) \in E(\mathcal{M})$ for $i = 0, \dots, \kappa - 1$. We call $\sigma_{Z_j Z_{j+1}}$ a $(\mathcal{M}, \mathcal{M}')$-*simulation path* (or just a *simulation path*, if no confusion can arise), since this path simulates a single transition in $\mathcal{M}'$ using edges of $\mathcal{M}$.

These simulation paths are concatenated together to form a canonical path γ_{wz} from w to z in $\mathcal{G}$, as follows:

$$\gamma_{wz} = (\sigma_{Z_0 Z_1}, \sigma_{Z_1 Z_2}, \dots, \sigma_{Z_{\ell-1} Z_\ell}).$$

The following algorithmic interpretation of γ_{wz} may be useful as an illustration. Begin by querying γ'_{wz} for the first transition (Z_0, Z_1) to simulate. Beginning from $h(Z_0)$, perform transitions in the corresponding simulation path until state $h(Z_1)$ is reached. Now query γ'_{wz} for the next transition (Z_1, Z_2) and simulate that; and so on, until you have simulated all of γ'_{wz}, completing the simulation path γ_{wz}.

Denote the set of all $(\mathcal{M}, \mathcal{M}')$-simulation paths as

$$\Sigma = \{\sigma_{xy} \mid (x, y) \in E(\mathcal{M}')\}\,.$$

For these simulation paths to give rise to canonical paths for $\mathcal{M}$ with low congestion, no transition in $\mathcal{M}$ can feature in too many simulation paths. For each $t \in E(\mathcal{M})$, let

$$\Sigma(t) = \{\sigma \in \Sigma \mid t \in \sigma\}$$

be the number of simulation paths containing the transition t. The measures of quality of Σ are the maximum number of simulation paths using a given transition,

$$B(\Sigma) = \max_{t \in E(\mathcal{M})} |\Sigma(t)|,$$

and the length of a maximal simulation path,

$$\ell(\Sigma) = \max_{\sigma \in \Sigma} |\sigma|.$$

Note that the simulation paths all depend on the fixed surjection h, and so the same is true of the bounds $B(\Sigma)$ and $\ell(\Sigma)$. In particular, the quality of the surjection h is measured by $\max_{y \in \Omega} |\{z \in \Omega' \mid h(z) = y\}$ and this quantity will be needed when calculating $B(\Sigma)$ in particular (as we will see in Section 4).

Some chains are ill-suited to simulation. For instance, a transition in $\mathcal{M}'$ with large capacity might necessarily be simulated in $\mathcal{M}$ by a simulation path that contained an edge with small capacity. Hence we define two quantities that measure the gap between π and π', and P and P'. We define the *simulation gap* $D(\mathcal{M}, \mathcal{M}')$ between $\mathcal{M}$ and $\mathcal{M}'$ by

$$D(\mathcal{M}, \mathcal{M}') = \max_{\substack{uv \in E(\mathcal{M}) \\ zw \in E(\mathcal{M}')}} \frac{\pi'(z)\, P'(z, w)}{\pi(u)\, P(u, v)}.$$

We combine these ingredients in the following lemma. The key point is that the congestion across transitions in $E(\mathcal{M}')$ (and hence across the simulation paths) is already bounded. If the number of simulation paths making use of any given transition in $E(\mathcal{M})$ is bounded, and the graphs are not too incompatible then the congestion of each two-stage direct canonical path in Γ will also be bounded.

Theorem 2.1. *Let $\mathcal{M}$ and $\mathcal{M}'$ be two ergodic Markov chains on Ω and Ω'. Fix a surjection $h : \Omega' \to \Omega$. Let Γ' be a set of canonical paths in $\mathcal{M}'$ and let Σ be a set of $(\mathcal{M}, \mathcal{M}')$-simulation paths defined with respect to h. Let $D = D(\mathcal{M}, \mathcal{M}')$ be the simulation gap defined above. Then there exists a set Γ of canonical paths in $\mathcal{M}$ whose congestion satisfies*

$$\bar{\rho}(\Gamma) \leq D\, \ell(\Sigma)\, B(\Sigma)\, \bar{\rho}(\Gamma').$$

Proof. We begin with the definition of congestion of the set of paths Γ for $\mathcal{M}$), and work towards rewriting all quantities in terms of the set of canonical paths Γ' for $\mathcal{M}'$, as follows:

$$\begin{aligned}
\bar{\rho}(\Gamma) &= \max_{uv\in E(\mathcal{M})} \frac{1}{\pi(u)P(u,v)} \sum_{\substack{x,y\in\Omega\\ uv\in\gamma_{xy}}} \pi(x)\pi(y)\,|\gamma_{xy}| \\
&= \max_{uv\in E(\mathcal{M})} \frac{1}{\pi(u)P(u,v)} \sum_{\substack{zw\in E(\mathcal{M}')\\ \sigma_{zw}\in\Sigma(uv)}} \sum_{\substack{x,y\in\Omega\\ \sigma_{zw}\in\gamma_{xy}}} \pi(x)\pi(y)\,|\gamma_{xy}| && (2.1)\\
&\le \max_{uv\in E(\mathcal{M})} \frac{1}{\pi(u)P(u,v)} \sum_{\substack{zw\in E(\mathcal{M}')\\ \sigma_{zw}\in\Sigma(uv)}} \sum_{\substack{x,y\in\Omega\\ zw\in\gamma'_{xy}}} \pi(x)\pi(y)\,|\gamma_{xy}|. && (2.2)
\end{aligned}$$

Line (2.1) splits the two-stage canonical paths into simulation paths. By a slight abuse of notation, we write $\sigma_{zw} \in \gamma_{xy}$ to mean that σ_{zw} is one of the simulation paths which was concatenated together to make γ_{xy}. Here we assume that each edge uv occurs at most once in any given simulation path $\sigma \in \Sigma$. Otherwise, the second use of uv in any σ forms a cycle which may be pruned: continue pruning until no edge appears more than once in any simulation path. We also assume that no self-loops appear in any simulation path (these may also be pruned.) Line (2.2) uses the one-to-one correspondence between γ_{xy} and γ'_{xy} for all $x, y \in \Omega$. Note that this correspondence is still one-to-one even when $\Omega \subset \Omega'$, since the endpoints are identical (as h acts as the identity map on Ω) and each canonical path is determined by its endpoints.

Now we observe that for all $uv \in E(\mathcal{M})$ and $zw \in E(\mathcal{M}')$, by definition of the simulation gap D,

$$\frac{1}{\pi(u)P(u,v)} \le D\,\frac{1}{\pi'(z)P'(z,w)}.$$

Similarly, for all $x, y \in \Omega'$,

$$|\gamma_{xy}| \le \ell(\Sigma)\,|\gamma'_{xy}|.$$

Therefore, as each summand of (2.2) is nonnegative and $\Omega \subseteq \Omega'$, we obtain

$$\begin{aligned}
\bar{\rho}(\Gamma) &\le D\,\ell(\Sigma) \max_{uv\in E(\mathcal{M})} \sum_{\substack{zw\in E(\mathcal{M}')\\ \sigma_{zw}\in\Sigma(uv)}} \frac{1}{\pi'(z)P'(z,w)} \sum_{\substack{x,y\in\Omega'\\ zw\in\gamma'_{xy}}} \pi'(x)\pi'(y)\,|\gamma'_{xy}| \\
&\le D\,\ell(\Sigma) \max_{uv\in E(\mathcal{M})} \sum_{\substack{zw\in E(\mathcal{M})\\ \sigma_{zw}\in\Sigma(uv)}} \bar{\rho}(\Gamma') \\
&\le D\,\ell(\Sigma)\,B(\Sigma)\,\bar{\rho}(\Gamma'),
\end{aligned}$$

as required. ☐

We remark that in the above theorem, the surjection h is only needed to construct the set of simulation paths (which are given as input to the theorem).

3 Analysis of the flip chain

Our aim is to define canonical paths in the flip chain that simulate canonical paths defined in the switch chain. There are two main differences between the switch and flip chains which cause difficulties. The state space of the flip chain is restricted to connected graphs, and no flip can disconnect a connected graph. However, a switch can increase or decrease the number of components in a graph by one. Hence, from a connected graph, the set of available flips may be strictly smaller than the set of available switches. Following [9] we overcome these difficulties in two steps, using an intermediate chain to bridge the gap.

Define the *connected switch chain* $\mathcal{M}_{\mathrm{SC}}$ to be the projection of the switch chain $\mathcal{M}_{\mathrm{S}}$ onto state space Ω_{F}. A transition of $\mathcal{M}_{\mathrm{SC}}$ is a *connected switch*, which is a switch on a connected graph that preserves connectedness. To be precise, if P_{SC} denotes the transition matrix of $\mathcal{M}_{\mathrm{SC}}$ and P_{S} denotes the transition matrix of $\mathcal{M}_{\mathrm{S}}$, then for all $x \neq y \in \Omega_{\mathrm{S}}$,

$$P_{\mathrm{SC}}(x,y) = \begin{cases} P_{\mathrm{S}}(x,y) & \text{if } x, y \in \Omega_{\mathrm{F}}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Any transition of $\mathcal{M}_{\mathrm{S}}$ which produces a disconnected graph is replaced by a self-loop in $\mathcal{M}_{\mathrm{SC}}$. Hence, for all $x \in \Omega_{\mathrm{F}}$,

$$P_{\mathrm{SC}}(x,x) = P_{\mathrm{S}}(x,x) + \sum_{y \in \Omega_{\mathrm{S}} \setminus \Omega_{\mathrm{F}}} P_{\mathrm{S}}(x,y).$$

Transitions are symmetric, and so the stationary distribution π_{SC} of $\mathcal{M}_{\mathrm{SC}}$ is the uniform stationary distribution on Ω_F (that is, $\pi_{\mathrm{SC}} = \pi_{\mathrm{F}}$).

We proceed as follows. In Section 3.1 we show that a connected switch can be simulated by sequences of flips, which we will call a *long-flip*. This gives a set of $(\mathcal{M}_{\mathrm{F}}, \mathcal{M}_{\mathrm{SC}})$-simulation paths. Here the surjection h is the identity map, since these chains share the same state space. We apply Theorem 2.1 to this set of simulation paths.

Then in Section 4 we apply the theorem again to relate $\mathcal{M}_{\mathrm{SC}}$ and $\mathcal{M}_{\mathrm{S}}$. We use a construction of Feder et al. [9] to define a surjection $h : \Omega_S \to \Omega_F$. We then define $(\mathcal{M}_{\mathrm{SC}}, \mathcal{M}_{\mathrm{S}})$-simulation paths with respect to this surjection and apply Theorem 2.1. Theorem 1.1 is proved by combining the results of these two sections.

The two-stage direct method has lower cost, in the sense of relaxation of the mixing time, than the comparison method used in [9]. Both approaches have at their heart the construction of paths in $\mathcal{M}_{\mathrm{F}}$ to simulate transitions in $\mathcal{M}_{\mathrm{S}}$, but the two-stage canonical path analysis is more direct and yields significantly lower mixing time bounds than those obtained in [9].

3.1 Simulating a connected switch: the long-flip

We must define a set of $(\mathcal{M}_{\mathrm{F}}, \mathcal{M}_{\mathrm{SC}})$-simulation paths, as defined in Section 2. The two Markov chains have the same state space Ω_F, and so we use the identity map as the surjection.

Let $S = (Z, Z')$ be a connected switch which we wish to simulate using flips. Then Z and Z' are connected graphs which differ by exactly four edges, two in $Z \setminus Z'$ and two in $Z' \setminus Z$. We say that the two edges $ab, cd \in Z \setminus Z'$ have been *switched out* by S, and the two edges

$ac, bd \in Z' \setminus Z$ have been *switched in* by S. Given an arbitrary labelling of one switch edge as ab and the other as cd, these four permutations of the vertex labels

$$(\,), \quad (a\,b)(c\,d), \quad (a\,c)(b\,d), \quad (a\,d)(b\,c) \tag{3.2}$$

preserve the edges switched in and out by $(ab, cd \Rightarrow ac, bd)$.

To simulate the connected switch $S = (Z, Z')$ we will define a simulation path called a *long-flip*, which consists of a sequence of flips

$$\sigma_{ZZ'} = (X_0, X_1, \ldots, X_{i-1}, X_i \ldots, X_\kappa)$$

with $X_0 = Z$ and $X_\kappa = Z'$, such that $(X_{i-1}, X_i) \in E(\mathcal{M}_{\mathrm{F}})$ for $i = 1, \ldots, \kappa$. We define these simulation paths in this section, and analyse their congestion in Section 3.2.

A long-flip uses a *hub path* $p = (b, p_1, p_2, \ldots, p_{\nu-1}, c)$ from b to c. In the simplest case (in which p does not contain a or d and no internal vertex of p is adjacent to either of these vertices) we will simply "reverse" p using flips, as illustrated later in Figure 4. Here by "reversing p" we imagine removing the path p with vertex b at the "left" and c at the "right". Now remove p from the graph and replacing it in the reverse orientation, but with a reattached to the left-most vertex of p, which is now c, and with d reattached to the right-most vertex of p, which is now b. This has the effect of deleting the edges ab, cd and replacing them with ac, bd, as desired. This process is described in more detail in Section 3.1.1 below The choice of hub path is important, and we now explain how the hub path will be (deterministically) selected.

Suppose that $(ab, cd \Rightarrow ac, bd)$ is the connected switch which takes Z to Z', which we wish to simulate using flips. We say that a path p is a *valid hub path* for the switch $(ab, cd \Rightarrow ac, bd)$ in Z if it satisfies the following (a, b, c, d) *path conditions*:

(i) p is a (b, c)-path in Z;

(ii) a and d do not lie on p; and

(iii) the vertices of p are only adjacent through path edges; that is, there are no edges in Z of the form $p_i p_j$ with $|i - j| \neq 1$ for $i, j \in \{0, \ldots, |p|\}$.

We refer to condition (iii) as the *no-shortcut property*. Given a path p which satisfies conditions (i) and (ii), it is easy to find a path with the no-shortcut property, by simply following the shortcuts. A shortest (b, c)-path is a valid (a, b, c, d) path provided it avoids a and d.

Given Z and the switch edges ab, cd, we perform the following procedure:

- Let p be the lexicographically-least shortest (b, c)-path in Z which avoids a and d, if one exists;

- If not, let p be the lexicographically-least shortest (a, d)-path in Z which avoids b and c, if one exists, and apply the vertex relabelling $(a\,b)(c\,d)$. (The result of this relabelling is that now p is the lexicographically-least shortest (b, c)-path in Z which avoids a and d.)

- If a path p was found, return the path p. Otherwise, return FAIL.

If a path p was found by the above procedure then we will say that we are in Case 1. Otherwise the output is "FAIL", and we say that we are in Case 2.

Suppose that if p is a lexicographically-least shortest path between vertices α and β in a graph Z. We treat p as a directed path from α to β, with α on the "left" and β on the "right". Suppose that vertices γ, δ lie on p, with γ to the "left" of δ. Then the lexicographically-least shortest (γ, δ)-path in Z is a subpath of p (or else we contradict the choice of p). This important property, satisfied by hub paths, will be called the *recognisability property.* It will enable us to reconstruct the hub path cheaply, as we will see in Section 3.2.

We now define the simulation path for Case 1 and Case 2 separately. Recall that since flips cannot disconnect a connected graph, the simulation paths always remain within Ω_F.

3.1.1 Case 1: the long-flip, and triangle-breaking

We must define a simulation path $\sigma_{ZZ'}$ for the connected switch from Z to Z' given by $(ab, cd \Rightarrow ac, bd)$. The path p selected above (the lexicographically-least shortest (b, c)-path which avoids a and d) is a valid hub path for this switch, and we refer to it as the hub path. If p has length 1 then this switch is a flip, and so we assume that p has length at least two.

We simulate the connected switch $(ab, cd \Rightarrow ac, bd)$ by reversing the path p, as shown in Figure 4. In (A), the switch $(ab, cd \Rightarrow ac, bd)$ is shown with alternating solid and dashed edges, with solid edges belong to Z and dashed edges belonging to Z'. Write the hub path as $p = (b, p_1, \ldots, p_{\nu-1}, c)$, where $b = p_0$ and $c = p_\nu$.

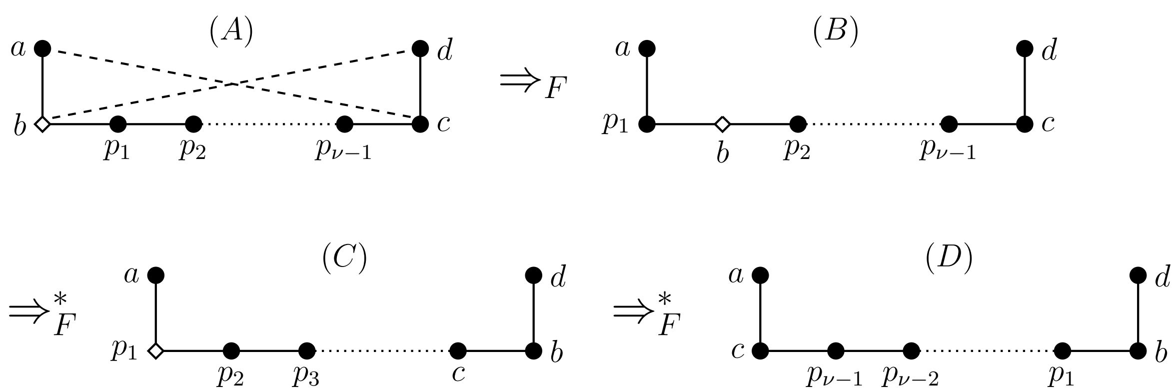


Figure 4: A hub path reversed by flips.

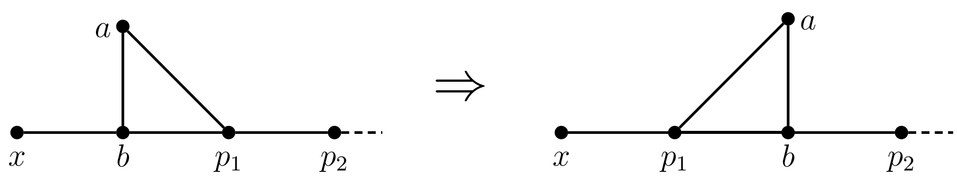
The reversal of p proceeds in *runs*, in which successive vertices are moved along the path one at a time into their final position. The run that moves vertex p_i into place is called the *p_i-run*, during which p_i is referred to as the *bubble* due to the way it percolates along path p. The first stage of the path reversal is the b-run, which moves b adjacent to d. Next is the p_1-run, and then the p_2-run, until the $p_{\nu-2}$-run, which puts c adjacent to a and completes the reversal of p. Specifically, the p_i-run consists of the sequence of flips

$$(a, p_i, p_{i+1}, p_{i+2}), (p_{i+1}, p_i, p_{i+2}, p_{i+3}), (p_{i+2}, p_i, p_{i+3}, p_{i+4}), \ldots, (p_{\nu-1}, p_i, c, p_{i-1}),$$

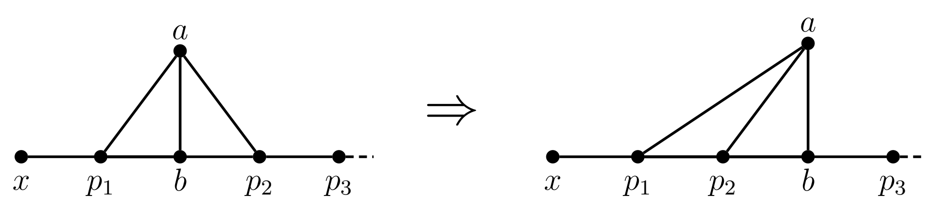
where $p_0 = b$, $p_\nu = c$ (and in the case of the b-run, p_{-1} is understood to denote d). After the p_i run, c has moved one step closer to a on path p. See Figure 4, where the current bubble is shown as a diamond. The result of the first flip of the b-run is shown in (B), and the situation at the end of the b-run is shown in (C), At this stage, b is in its final position, adjacent to d. Finally in (D) we see the graph after all runs are complete, with the path reversed. (Note that $\Rightarrow_F$ denotes a single flip while $\Rightarrow_F^*$ denotes a sequence of zero or more flips.)

Since the distance from c to a decreases by one after each run, the total number of flips used to reverse the path p is $\frac{1}{2}\nu(\nu+1)$. The no-shortcut property of hub paths is required so that no flip in any run is blocked by parallel edges within p.

However, the first flip of the b-run is blocked if p_1 is adjacent to a. Before we can reverse the path p, we must break this *triangle* $[a, b, p_1]$ using the following procedure. Choose the least-labelled vertex x that is adjacent to b but not p_1. The (a, b, c, d) path conditions imply that p_2 is adjacent to p_1, but not to b, so the existence of x follows from $2r$-regularity. Then the triangle $[a, b, p_1]$ can be removed by performing a flip on the path (x, b, p_1, p_2), as shown below.



Now p_1 is still adjacent to a, but it is no longer on the (b, c)-path, and the length of the (b, c)-path has been reduced by 1. However, it is possible that p_2 is also adjacent to a, so we may still have a triangle. If so, we must repeat the process, as indicated below, successively shifting b to the right on the (b, c)-path, and reducing its length. To break the triangle $[a, b, p_2]$ we flip on (p_1, b, p_2, p_3).



More generally, if the edges $ap_1, \ldots, ap_k$ are all present but the edge ap_{k+1} is not then the k'th triangle $[a, b, p_k]$ is broken with the flip $(p_{k-1}, b, p_k, p_{k+1})$. After this flip, b is adjacent to p_k and p_{k+1}. We call these *triangle-breaking flips*.

The no-shortcut property of the (b, c)-path guarantees that all the triangle-breaking flips after the first one are valid. The triangle-breaking process must end when bc becomes an edge, at the very latest. This implies that $k \leq \nu - 1$. Furthermore, since a has degree $2r$ we also have $k \leq 2r - 1$. When the process terminates, the edge ap_{k+1} is absent and bp_{k+1} is the first edge on the (b, c)-path. Now the long-flip can be performed on this new (b, c) hub path. If $k = \nu - 1$ then $p_{k+1} = c$ and this long-flip is a single flip.

After the path is fully reversed we must undo the triangle removals, by reversing each triangle-breaking flip, in reverse order. That is, if k triangle-breaking flips were performed, then performing these flips in order will reinstate these triangles:

$$(p_{k-1}, p_k, b, p_{k+1}),\ (p_{k-2}, p_{k-1}, b, p_k), \ldots (p_1, p_2, b, p_3),\ \ (x, p_1, b, p_2).$$

(When $k = 1$, only the last flip is performed.) We call these *triangle-restoring flips*. These flips are not blocked because the only edges whose presence could block them are guaranteed not to exist by the no-shortcut property of p, and by the fact that these edges were removed by the triangle-breaking flips.

3.1.2 Case 2: The detour flip

In Case 2, we know that there are no paths from b to c in Z which avoid both a and d, and that there are no paths from a to d in Z which avoid b and c. It follows that Z must have the structure illustrated in Figure 5. In this figure, each Z_{ij} (shown as a grey ellipse) is an

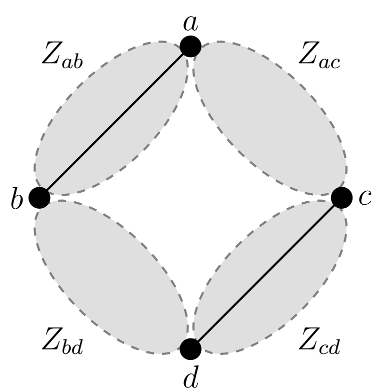


Figure 5: The structure of a graph Z in Case 2.

induced subgraph of Z, with $i, j \in Z_{ij}$. There are no edges from $Z_{ij} \setminus \{i, j\}$ to $Z_{k\ell} \setminus \{k, \ell\}$ whenever $\{i, j\} \neq \{k, \ell\}$, by the assumption that Z is in Case 2. Furthermore, Z_{ab} and Z_{cd} are both connected, since Z is connected and $ab, cd \in E(Z)$.

At most one of Z_{ac} or Z_{bd} may be empty or disconnected, since Z is connected. A deterministic procedure will now be described for constructing a 3-path (u, d, w, z) in Z which will be used for the first flip in the simulation path.

There are two subcases to consider for the choice of w:

- First suppose that, after applying the relabelling $(a\,b)(c\,d)$ if necessary, Z_{ac} is empty or disconnected. Since cd is not a bridge in Z, by Lemma 1.2(ii), there exists a cycle in Z_{cd} which passes through the edge cd. It follows that d has at least one neighbour in $Z_{cd} \setminus c$ which is connected to c by a path in $Z_{cd} \setminus cd$. Let w be the least-labelled such neighbour of d in $Z_{cd} \setminus cd$.

- Secondly, suppose that both Z_{ac} and Z_{bd} are nonempty and connected. (So far, we have not performed any relabelling.) Then at most one of $Z_{ab} \setminus ab$, $Z_{cd} \setminus cd$ may be disconnected, or else the graph Z' will be disconnected, a contradiction. Applying the relabelling $(a\,c)(b\,d)$ if necessary, we can assume that $Z_{cd} \setminus cd$ is connected. Hence there is a path from c to d in $Z_{cd} \setminus cd$, so d has a neighbour in $Z_{cd} \setminus c$ which is joined to c by a path in $Z_{cd} \setminus cd$. Let w be the least-labelled such neighbour of d in $Z_{cd} \setminus c$.

In both subcases, we have established the existence of a suitable vertex w. Next, observe that since (in both the above subcases) Z_{bd} is nonempty and connected, there is a path from b to d in Z_{bd} of length at least two (as $bd \notin E(Z)$). This implies that d has a neighbour in Z_{bd} which is joined to b by a path in $Z_{bd} \setminus d$. Let u be the least-labelled such neighbour of d in $Z_{bd} \setminus b$. Finally, since all neighbours of w belong to Z_{cd} and $u \notin Z_{cd}$, it follows that $\{u, w\} \subseteq N_Z(d) \setminus N_Z(w)$. Since Z is regular, this implies that

$$|N_Z(w) \setminus N_Z(d)| = |N_Z(d) \setminus N_Z(w)| \geq 2.$$

Hence there is some vertex other than d in $N_Z(w) \setminus N_Z(d)$. Let z be the least-labelled such vertex. Then $z \in Z_{cd} \setminus \{c, d\}$, since $c \in N_Z(d)$ and every neighbour of w belongs to Z_{cd}.

Now we can define the simulation path $\sigma_{ZZ'}$ from $X_0 = Z$ to $X_\kappa = Z'$. The first step in the simulation path is the flip on the path (u, d, w, z), giving the graph X_1. We call this the *detour flip*, as it will give us a way to avoid d. See the first line of Figure 6.

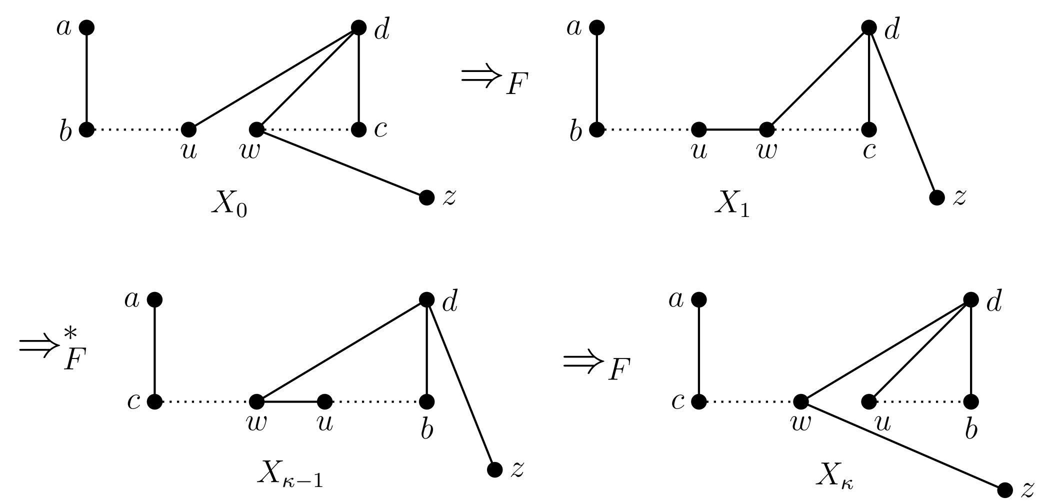


Figure 6: Case 2: the simulation path

After the detour flip, there is a path from b to u in $X_1 \setminus \{a, d\}$, by choice of u, and there is a path from c to w in $X_1 \setminus \{a, d\}$, by choice of w. It follows that there exists a (b, c)-path in X_1 which avoids $\{a, d\}$: let p be the lexicographically-least shortest such path. Note that p necessarily contains the edge uw. (See X_1 in Figure 6: the path p is indicated by the dotted lines together with the edge from u to w.) We proceed as in Case 1, simulating the switch $(ab, cd \Rightarrow ac, bd)$ from X_1 using a long-flip with hub path p. At the end of the long-flip the path p is reversed and we reach the graph $X_{\kappa-1}$ shown at the bottom left of Figure 6.

For the final step of the simulation path, we reverse the detour flip by performing a flip on the path (u, w, d, z). This produces the final graph $X_\kappa = Z'$ shown at the bottom right of Figure 6, completing the description of the simulation path in Case 2.

3.2 Bounding the congestion

The previous section defines a set Σ_{F} of $(\mathcal{M}_{\mathrm{F}}, \mathcal{M}_{\mathrm{SC}})$-simulation paths. First, we prove an upper bound on the maximum length $\ell(\Sigma_{\mathrm{F}})$ of a simulation path in Σ_{F}.

Lemma 3.1. *If $r \geq 2$ then $\ell(\Sigma_{\mathrm{F}}) \leq \frac{1}{2}n^2$.*

Proof. Since hub paths are simple and do not include a or d, every hub path contains at most $n-2$ vertices. Therefore, at most $\frac{1}{2}(n-2)(n-3)$ path-reversing flips will be required. Furthermore, we may have up to $2r-1$ triangle-breaking flips, and up to $2r-1$ triangle-restoring flips. Additionally, if we are in Case 2 then we also have a detour flip and its inverse. Therefore

$$\ell(\Sigma_{\mathrm{F}}) \leq \frac{(n-2)(n-3)}{2} + 4r,$$

and this is bounded above by $\frac{1}{2}n^2$ whenever $r \geq 2$, since $n \geq 2r+1$. □

For the remainder of this subsection we work towards an upper bound on $B(\Sigma_{\mathrm{F}})$, the maximum number of simulation paths containing a given flip. A given flip $t = (X, X')$ could perform one of a number of roles in a simulation path for a connected switch $S = (Z, Z')$:

- a triangle-breaking flip;
- a triangle-restoring flip;
- a flip on the path reversal (that is, part of the long-flip);
- in Case 2, the detour flip (u, w, d, z);
- in Case 2, the inverse detour flip (u, d, w, z).

We bound the number of simulation paths that could use t in each of these roles separately, in Lemmas 3.2–3.4 below. Summing over these bounds over all roles yields an upper bound for $B(\Sigma_{\mathrm{F}})$. In our exposition, we attempt to uniquely determine a given simulation path $\sigma_{ZZ'}$ which contains t, by "guessing" from a number of alternatives. Since the number of possible alternatives is precisely the number of simulation paths $\sigma_{ZZ'}$ that use t, this approach gives the desired bound.

The simulation path is determined once we have identified the switch $(ab, cd \Rightarrow ac, bd)$ and either Z or Z' (since each of these can be obtained from the other using the switch or its inverse). Observe that in order to uniquely identify the switch it is not enough to just identify the two edges e_1, e_2 which are switched out (or to just identify the two edges that are switched in). This follows since there are up to two possible switches on these edges: if $e_1 = ab$ and $e_2 = cd$ then we could switch in either the edges ac, bd or the edges ad, bc. We pay a factor of 2 to distinguish between these two possibilities, and will describe this as *orienting* one of the switch edges.

Finally, suppose that the current flip $t = (X, X')$ switches out the edges $\alpha\beta$, $\gamma\delta$ and replaces them by $\alpha\gamma$, $\beta\delta$. If both $\alpha\delta$ and $\beta\gamma$ are present in X (and hence, both present in X') then either one of these could have been the middle edge of the 3-path chosen for the flip. In this case, if we wish to identify β, say, then there are 4 possibilities. However, if we know a priori that the edge $\alpha\delta$ is not present in X, say, then we can guess β from two possibilities.

Lemma 3.2. *If $t = (X, X')$ is a detour flip then we can guess the connected switch $S = (Z, Z')$ from at most $8r^2n$ possibilities. The same bound holds if t is an inverse detour flip.*

Proof. First suppose that t is a detour flip. This is the first flip in the simulation path, so $Z = X$.

- The flip is on the 3-path (u, w, d, z), and $uz \notin E(X)$ by construction, so we can guess d from 2 possibilities.
- Next, we can guess $c \in N_X(d)$ from at most $2r$ possibilities.
- Finally, guess the edge $ab \in E(X)$ and orient it, from $2rn$ possibilities.

From the switch edges and Z we can obtain Z'. Hence we can guess the switch S from at most $8r^2n$ possibilities overall when t is a detour flip. The same argument holds when t is an inverse detour flip, replacing X and Z by X' and Z' and exchanging the roles of b and c. □

Lemma 3.3. *Suppose that $t = (X, X')$ is a triangle-breaking flip. Then we can guess the connected switch $S = (Z, Z')$ from at most $4(2r)^8n$ possibilities. The same bound holds if t is a triangle-restoring flip.*

Proof. Suppose that t is a triangle-breaking flip.

- One of the vertices involved in the flip t is b, so we can guess it from 4 alternatives. (If t is not the first triangle-breaking flip then by the no-shortcuts property, we can guess b from 2 alternatives. But in the first triangle-breaking flip xp_2 may be an edge of Z, in which case there are 4 possibilities for b.)
- We can guess a by adjacency to b, from $2r$ possibilities.
- We can guess the edge $cd \in E(X)$ and orient it, from $2rn$ possibilities.

We now know the switch vertices, so recovering Z will give us the rest of the path. There may have been some other triangle-breaking flips which took place before t, and if we are in Case 2 then there was also a detour flip at the start of the simulation path. We continue our accounting below.

- First suppose that we are in Case 1, and that t is the kth triangle-breaking flip which has been performed (so far). Then $k \geq 1$. Let $p' = (x, p_1, \ldots, p_k, b)$ denote the path containing vertices which were removed from the hub path by the first k triangle-breaking flips. Now (a, p_1, x) is a 2-path in X, so we can guess x and p_1 from $2r(2r-1) \leq (2r)^2$ alternatives.

- Next, p_k belongs to $N_X(b) \setminus a$, so we can guess p_k from at most $2r$ alternatives. (Here we allow the possibility that $p_1 = p_k$, which covers the case that t is the first triangle-breaking flip.) Then we can reconstruct p' using the fact that $(p_1, \ldots, p_k)$ is the lexicographically-least shortest (p_1, p_k)-path in $X \setminus \{a, d\}$, by the recognisability property. Using p', we can reverse these k triangle-breaking flips to obtain Z from X.
- Finally, we must guess whether we are in Case 1 (1 possibility) or Case 2. If we are in Case 2 then we must also reverse the detour flip before we obtain Z. Since u and w are neighbours of d and z is a neighbour of w, in this step we guess from at most $1 + (2r)^2\,(2r-1) < (2r)^3$ possibilities.

Overall, we can guess the switch S from a total of at most $4(2r)^8 n$ possibilities, when t is a triangle-breaking flip.

The argument is similar when t is a triangle-restoring flip:

- First, suppose that we are in Case 1. Guess b, then guess d from the neighbourhood of b, then guess and orient the edge $ac \in E(X)$. This gives the switch edges, guessed from at most $4\,(2r)^3$ alternatives.
- Recover $p' = (x, p_1, \ldots, p_k, b)$ as in the triangle-breaking case, from at most $(2r)^3$ alternatives. Now we can Z' uniquely by performing the remaining triangle-restoring flips to move b past p_1.
- Again, there at most $(2r)^3$ possibilities for deciding whether or not we are in Case 2 and if so, guessing the inverse detour flip and applying it to produce Z'.

Hence there are at most $4(2r)^8 n$ possibilities for the connected switch S when t is a triangle-restoring flip. □

Lemma 3.4. *Suppose that $t = (X, X')$ is a flip on the path-reversal. We can guess the connected switch $S = (Z, Z')$ from at most $4(2r)^8\,((2r)^2 - 1)\,n^3$ possibilities.*

Proof. Suppose that t is a flip on the path-reversal with respect to the hub path $p = (b, p_1, \ldots, p_{\nu-1}, c)$, where $b = p_0$ and $c = p_\nu$. Then t is part of a p_i run, for some i. The path-reversal may have been preceded by some triangle-breaking flips, and (if we are in Case 2) a detour flip. We need to guess the switch edges and reconstruct Z.

First suppose that we are in Case 1. The typical situation is shown in Figure 7, which shows that graph X' produced by the flip t. The current bubble p_i is shown as a diamond, and the flip t has just been performed on the 3-path $(p_j, p_i, p_{j+1}, p_{j+2})$, moving p_i past p_{j+1}.

- The current bubble p_i is one of the four vertices involved in the flip, so it can be guessed from four possibilities.
- Next we guess a, c and d from n^3 alternatives.
- We recover b as follows: if the current run is the b-run then b is the bubble; otherwise, b is already adjacent to d, and we can guess it from the $2r$ neighbours of d. This gives $2r + 1$ possibilities for b.
- Vertices a' and c' can be guessed from at most $2r$ alternatives each, by their adjacency to a and c.

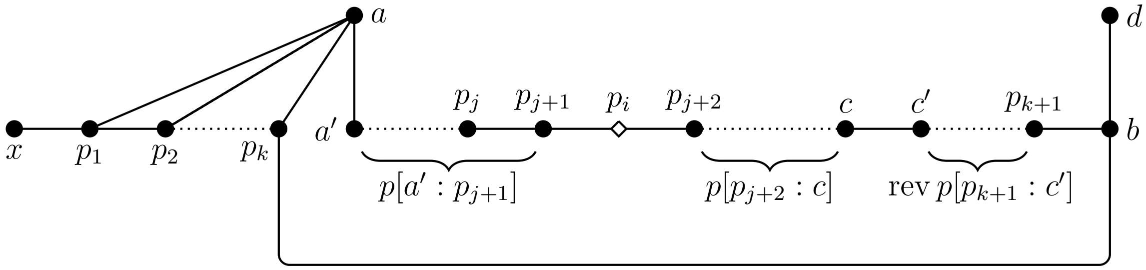


Figure 7: Reconstruction of Z from X', the flip edges and some guessed information.

Now we know the switch $(ab, cd \Rightarrow ac, bd)$, and it remains to recover Z. In order to do that, we must recover the path p, reverse the path-reversal flips and then reverse the triangle-breaking flips, if any.

Observe that having identified p_i, the identities of p_{j+1} and p_{j+2} can be inferred. To see this, note that $p_i p_{j+2}$ is one of the two edges switched in by the flip t, which identifies p_{j+2}, while $p_{j+1}p_{j+2}$ is one of the two edges switched out by the flip t, which identifies p_{j+1}.

Next, observe that $a' = p_{i+1}$, and so the subpath $p_{i+1} \cdots p_{j+1}$ is the lexicographically-least shortest (p_{i+1}, p_{j+1})-path in $X' \setminus \{a, d\}$, by the recognisability property. This holds since the flips made on the path-reversal so far, between Z and X', have not altered this section of the original path. Similarly, the subpath $p_{j+2} \cdots c$ can be recovered as the lexicographically-least shortest (p_{j+2}, c)-path in $X' \setminus \{a, d\}$.

If the current bubble is b then $c' = c$. Otherwise, observe that $c' = p_{i-1}$, since c' is the most recent bubble that successfully completed its run. We continue our accounting below.

- Guess p_{k+1} from $N_{X'}(b) \setminus d$, from at most $2r - 1$ possibilities. Reconstruct the subpath $p_{k+1} \cdots p_{i-1}$ as the lexicographically-least shortest (p_{k+1}, p_{i-1})-path in $X' \setminus \{a, d\}$.
- By now we have recovered all sections of the path between a' and b as shown in Figure 7, and we can reverse all path-reversing flips. Now there may have been no triangle-restoring flips, in which case we have constructed Z, or else we may need to reverse the triangle-breaking flips as explained earlier, after guessing the identities of x, p_1 and p_{k-1} from at most $(2r)^2(2r-1)$ possibilities. This gives a factor of at most $1 + (2r)^2(2r-1) \leq (2r)^3$.
- Finally, multiply by $1 + (2r)^2(2r - 1) \leq (2r)^3$ to account for the fact that we may be in Case 1 or Case 2, as explained in Lemma 3.3.

Overall, we have guessed the switch S from at most

$$4(2r + 1)(2r - 1)(2r)^8\, n^3 = 4(2r)^8 \left((2r)^2 - 1\right) n^3$$

possibilities, when t is a flip on a path-reversal. □

Now we apply the two-stage direct canonical path theorem to our $(\mathcal{M}_{\mathrm{F}}, \mathcal{M}_{\mathrm{SC}})$-simulation paths. Let $\bar{\rho}(\Gamma_{\mathrm{SC}})$ denote the congestion of some set of canonical paths for the connected switch $\mathcal{M}_{\mathrm{SC}}$. We will define such a set of canonical paths in Section 4 below.

Lemma 3.5. *The $(\mathcal{M}_{\rm F}, \mathcal{M}_{\rm SC})$-simulation paths defined in Section 3.1 define a set of canonical paths for $\mathcal{M}_{\rm F}$ with congestion $\bar{\rho}(\Gamma_{\rm F})$ which satisfies*

$$\bar{\rho}(\Gamma_{\rm F}) \leq 8\,(2r)^{11} n^4\, \bar{\rho}(\Gamma_{\rm SC}).$$

Proof. Firstly, we bound the maximum number $B(\Sigma_{\rm F})$ of simulation paths which contain a given flip t, by adding together the number of possibilities from each of the five roles that t may play. Using Lemmas 3.2– 3.4, we obtain

$$\begin{aligned} B(\Sigma_{\rm F}) &\leq 4(2r)^8 \left((2r)^2 - 1\right) n^3 + 8(2r)^8 n + 16r^2 n \\ &\leq 4\left((2r)^{10} - (2r)^8 + 2(2r)^6 + 1\right) n^3 \\ &\leq 4(2r)^{10}\, n^3, \end{aligned}$$

since $n \geq 2r \geq 4$. Next, we calculate the simulation gap $D_{\rm F} = D(\mathcal{M}_{\rm F}, \mathcal{M}_{\rm SC})$. Since $\mathcal{M}_{\rm F}$ and $\mathcal{M}_{\rm SC}$ share the same state space Ω_F, and both have uniform stationary distribution,

$$D_{\rm F} = \max_{\substack{uv \in E(\mathcal{M}_{\rm F}) \\ zw \in E(\mathcal{M}_{\rm SC})}} \frac{|\Omega_{\rm F}|\, P_{\rm SC}(z,w)}{|\Omega_{\rm F}|\, P_{\rm F}(u,v)} = \frac{P_{\rm SC}(z,w)}{P_{\rm F}(u,v)}.$$

Let $uv \in E(\mathcal{M}_{\rm F})$ and $zw \in E(\mathcal{M}_{\rm SC})$. Then $u \neq v$, as self-loop transitions are not included in $E(\mathcal{M}_{\rm F})$ or $E(\mathcal{M}_{\rm SC})$. Hence

$$\frac{1}{P_{\rm F}(u,v)} \leq \frac{(2r)^3 n}{2} \quad \text{and} \quad P_{\rm SC}(z,w) = \frac{1}{3a_{n,2r}} \leq \frac{2}{r^2 n^2}$$

using (1.3), since $n \geq 8$. Therefore $D_{\rm F} \leq 8r/n$.

Substituting these quantities into Theorem 2.1, using Lemma 3.1, we obtain

$$\begin{aligned} \bar{\rho}(\Gamma_{\rm F}) &\leq D_F\, \ell(\Sigma_F)\, B(\Sigma_F)\, \bar{\rho}(\Gamma_{\rm SC}) \\ &\leq 8\,(2r)^{11} n^4\, \bar{\rho}(\Gamma_{\rm SC}), \end{aligned}$$

as claimed. □

4 Disconnected graphs

Next, we must define a set of $(\mathcal{M}_{\rm SC}, \mathcal{M}_{\rm S})$-simulation paths and apply the two-stage direct canonical path method. The process is similar to the approach introduced by Feder et al. [9]: the chief difference is in the analysis.

We begin by defining a surjection $h\colon \Omega_{\rm S} \to \Omega_{\rm F}$ such that $\max_{G \in \Omega_{\rm F}} |h^{-1}(G)|$ is polynomially bounded. Using this we construct short paths in $\mathcal{M}_{\rm SC}$ to correspond to single transitions in $\mathcal{M}_{\rm S}$. In [9] this step is responsible for a factor of $\mathrm{O}(r^{34} n^{36})$ in the overall mixing time bound. In contrast, by using the two-stage direct method we construct canonical paths with a relaxation of only a factor of $120rn^3$.

First we must define the surjection h. Let $G_S \in \Omega_{\mathrm{S}}$ and let $\{H_1, \dots, H_k\}$ be the set of components in G_S. Let v_i the vertex in H_i of highest label, for $i = 1, \dots, k$. By relabelling the components if necessary, we assume that

$$v_1 < v_2 < \cdots < v_k.$$

We call each v_i an *entry vertex*. By construction, v_k is the highest-labelled vertex in G_S.

In each component we will identify a *bridge edge* $e_i = v_i v_i'$, where v_i' is the neighbour of v_i with the highest label. We call edge v_i' an *exit vertex*. The graph $G = h(G_S)$ is formed from G_S by removing the bridge edges $e_1, \dots, e_k$ and replacing them with *chain edges* $e_1', \dots, e_k'$, where $e_i = v_i' v_{i+1}$ for $1 \le i \le k-1$, and $e_k' = v_k' v_1$. We call e_k' the *loopback edge* and we call this procedure *chaining*. (When discussing the chain, arithmetic on indices is performed cyclically; that is, we identify v_{k+1} with v_1 and v_{k+1}' with v_1'.)

By construction we have $v_i' < v_i < v_{i+1}$ for $i = 1, \dots, k-1$. Hence the "head" v_{i+1} of the chain edge e_i' is greater than the "tail" v_i', except for possibly the loopback edge: v_1 may or may not be greater than v_k'. We prove that $G = H(G_S)$ is connected in Lemma 4.1 below.

An example of a chained graph G is shown in Figure 8 in the case $k = 3$. Here $G = H(G_S)$ for a graph G_S with three components H_1, H_2, H_3. The shaded black vertices are the entry vertices v_1, v_2 and v_3, and the white vertices are the exit vertices v_1', v_2' and v_3'. After chaining, H_1 contains two cut edges (shown as dashed edges), because $e_1 = v_1 v_1'$ was part of a cut cycle in H_1. Similarly, H_3 contains one cut edge. Note that these dashed edges are not chain edges. The loopback edge is e_3' and $v_1 < v_2 < v_3$.

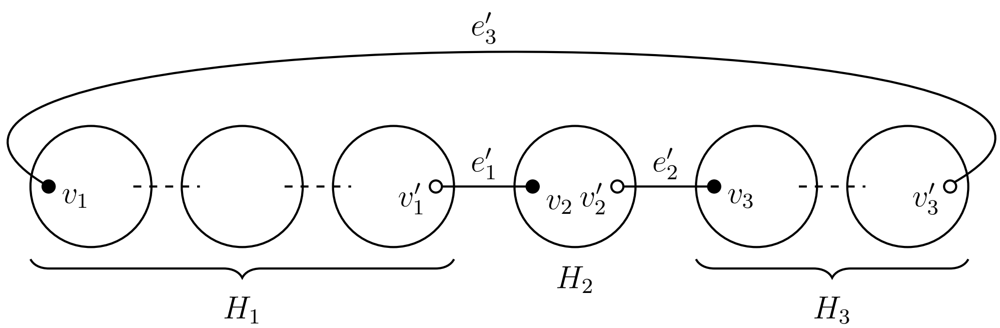


Figure 8: A connected graph $h(G_S) \in \Omega_{\mathrm{F}}$ obtained by chaining a disconnected graph $G_S \in \Omega_{\mathrm{S}}$ with three components H_1, H_2, H_3.

Lemma 4.1. *Let h be the map defined above. Then $h(G_S)$ is connected for all $G_S \in \Omega_{\mathrm{S}}$, and $h\colon \Omega_{\mathrm{S}} \to \Omega_{\mathrm{F}}$ is a surjection.*

Proof. Let $G_S \in \Omega_{\mathrm{S}}$ and suppose $G = h(G_S)$. Lemma 1.2(ii) implies that each connected component H_i remains at least 1-connected after removal of bridge edges. The chain edges then connect these components in a ring, forming a connected graph. Additionally, h preserves degrees since each vertex v_i, v_i' is adjacent to exactly one bridge edge that is removed and exactly one chain edge that is added. Hence, $G \in \Omega_{\mathrm{F}}$. Finally, observe that $h(G) = G$ whenever G is connected, which proves that h is a surjection since $\Omega_{\mathrm{F}} \subset \Omega_{\mathrm{S}}$. $\square$

Lemma 4.2. *Let $h\colon \Omega_{\mathrm{S}} \to \Omega_{\mathrm{F}}$ be defined as above. Then*

$$\max_{G\in\Omega_{\mathrm{F}}} |h^{-1}(G)| \le 2rn.$$

Proof. Let $G_F \in \Omega_F$ be given. First we prove that given an edge $e = \{v, w\}$ of G_F and an orientation (v, w) of e, there is a unique disconnected graph $G_S \in h^{-1}(G_F)$ such that when the chaining procedure is performed with input G_S (producing G_F), the loopback edge e'_k equals e and $v_1 = v$. We recover G_S from G_F and the oriented edge e using the *unchaining algorithm* which we now describe.

To identify e'_1 given e'_k and v_1, begin by searching the graph $G_F \setminus e'_k$, starting from v_1, until a vertex is reached with a label greater than v_1. By construction, this vertex must be v_2, as v_1 has the greatest label of any vertex in H_1. The edge traversed to reach v_2 must be e'_1. Repeat this process until the chain edges $e'_1, \ldots, e'_{k-1}$ have been discovered. Then G_S is recovered by deleting all chain edges (including the loopback edge) from G_F and replacing them with the bridge edges $e_1, \ldots, e_k$.

For future reference, note that this unchaining algorithm will still correctly recover G_S so long as v'_i is any neighbour of v_i in G_S (that is, v'_i does not need to be the neighbour with largest label).

Hence the number of choices for the oriented loopback edge gives an upper bound on $|h^{-1}(G_S)| - 1$. We use the naive estimate of $2r(n-1)$ as an upper bound on this quantity, since the loopback edge could be any oriented edge which does not originate in v_k (the highest-labelled vertex in G_F). Since G_F is the only connected graph in $h^{-1}(G_F)$, we obtain $|h^{-1}(G_S)| \le 2r(n-1) + 1 \le 2rn$. (It should be possible to obtain a tighter bound with a little more work, though we do not attempt that here.) □

4.1 Simulating general switches

Now we must construct a set of $(\mathcal{M}_{\mathrm{SC}}, \mathcal{M}_{\mathrm{S}})$-simulation paths Σ_{SC} with respect to the surjection h defined above. Given a switch $S = (W, W') \in E(\mathcal{M}_{\mathrm{S}})$, with $W, W' \in \Omega_{\mathrm{S}}$, we must simulate it by a path $\sigma_{WW'}$ in Γ_{SC}, consisting only of connected switches. We then give an upper bound on the congestion parameters for the set Σ_{SC} of simulation paths, and apply Theorem 2.1. The main difficulty is that switches in $\mathcal{M}_{\mathrm{S}}$ may create or merge components, whereas connected switches in $\mathcal{M}_{\mathrm{SC}}$ cannot.

In this section we refer to transitions in $E(\mathcal{M}_{\mathrm{S}})$ as *general switches*, to distinguish them from connected switches. We say that a general switch $S = (W, W')$ is

- *neutral* if W and W' have the same number of components;
- *disconnecting* if W' has one more component than W;
- *reconnecting* if W' has one fewer component than W.

The general switch $S = (ab, cd \Rightarrow ac, bd)$ is disconnecting if ab, cd form a 2-edge-cut in the same component of W and ac, bd do not. If S is a reconnecting switch then ab and cd below to distinct components of W. Finally, S is a neutral switch if ab and cd both belong to the same component of W, but do not form a 2-edge-cut, or if ab and cd form a 2-edge-cut in a component of W, and ac and bd form another 2-edge-cut in the same component.

To simulate the general switch $S = (W, W')$, we must define a path

$$\sigma_{W,W'} = (Z_0, Z_1, \ldots, Z_q),$$

in $E(\mathcal{M}_{\text{SC}})$, where $Z_0 = h(W)$ and $Z_q = h(W')$, such that $Z_i \in \Omega_{\text{F}}$ for $i = 0, \ldots, q$ and $(Z_i, Z_{i+1}) \in E(\mathcal{M}_{\text{SC}})$ for $i = 0, \ldots, q-1$. Here are the kinds of connected switch that we will need in our simulation paths.

- Suppose that a bridge edge $v_j v_j'$ of W is one of the edges to be removed by the switch S. This is a problem, since all bridge edges have been deleted and replaced by chain edges during the chaining process used to construct $Z_0 = h(W)$. To deal with this, we first perform a connected switch called a *bridge change* switch. The bridge change switch reinstates the edge $e_j e_j'$ (so that the desired switch S can be performed) and changes the choice of exit vertex (previously e_j') in H_j. Importantly, the resulting chain structure can still be unwound using the algorithm in Lemma 4.2. Specifically, to reinstate the bridge edge $e_j = v_j v_j'$ in component H_j, let $v_j^* \in N_{Z_0}(v_j) \setminus \{v_{j-1}'\}$ be the highest-labelled neighbour of v_j in $Z_0[H_j]$ and perform the connected switch $(v_j v_j^*, v_j' v_{j+1} \Rightarrow v_j v_j', v_j^* v_{j+1})$. (Clearly $v_j^* \neq v_j'$ since the edge $v_j v_j'$ is absent in Z_0, and $v_{j+1} v_j^*$ is not an edge of Z_0 due to the component structure of W, so this connected switch is valid.) The new chain edge is $v_j^* v_{j+1}$ and the new bridge edge is $v_j v_j^*$. We denote the bridge change switch by BΔ. See Figure 9.

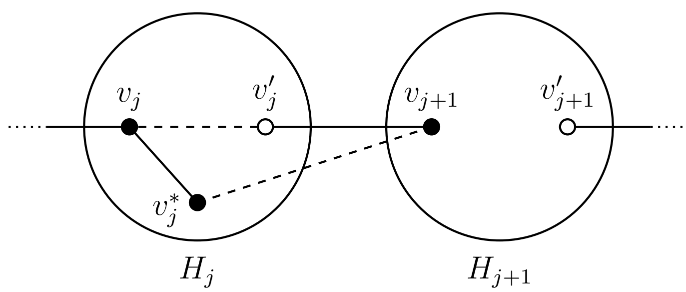


Figure 9: A bridge change in component H_j.

- We will need a connected switch to simulate the general switch S. We denote this switch as NS, DS or RS if S is neutral, disconnecting or reconnecting, respectively, and call it the *neutral* (respectively, *disconnecting* or *reconnecting*) *simulation flip*. It removes the same two edges that S removes, and inserts the same two edges that S inserts.

- Let Z be the current graph after the DS, RS or NS has been performed. We say that component H_j is *healthy* in Z if there is no neighbour of the entry vertex v_j in Z with an label higher than the exit vertex v_j'. We must ensure that all components of Z are healthy before performing the unchaining algorithm, or else we lose the guarantee that $Z = h(W')$.

  A component H_j can only become unhealthy if some switch along the simulation path involves the entry vertex v_j and introduces a new neighbour of v_j, say v_j^*, which has a higher label than the (current) exit vertex v'_j. In this case, v^*_j also has the highest label among all current neighbours of v_j. (In fact, the switch that makes H_j unhealthy will be the switch which simulates S, namely, the DS, RS or NS.) Note, if there has been a bridge change in component H_j at the start of the simulation path, then v'_j might not be the exit vertex originally produced in H_j by the chaining procedure. But this will not cause any problems, as we will see.

We can make H_j healthy by performing the bridge change switch $(v_j v^*_j, v'_j v_{j+1} \Rightarrow v_j v'_j, v^*_j v_{j+1})$, precisely as described above. (Now the main purpose of this switch is not to reinstate an edge, but to ensure the correct vertex of H_j becomes the exit vertex before unchaining.) When this bridge change is performed at the end of the simulation path we will refer to it as a *bridge rectification* switch, denoted BR.

- If S is a disconnecting switch then performing this switch will produce a disconnected graph (and destroy the chain structure) by splitting H_j into two components, one of which is not chained. In order to avoid this, we perform a *disconnected housekeeping switch*, denoted DHK, before the disconnecting simulation flip (DS). The DHK is defined in detail in Section 4.1.2 below.
- Similarly, if S is a reconnecting switch then performing this switch destroys the chain structure, by merging H_i and H_j together, forming one component which is incident with four chain edges instead of two. To fix the chain structure we perform a *reconnecting housekeeping switch*, denoted RHK, after the reconnecting simulation flip (RS). More detail is given in Section 4.1.3 below.

We now define simulation paths for the three types of general switch $S = (W, W')$. The paths are summarised in Table 1, with optional steps denoted by square brackets. Here "optional" means that these steps may or may not appear in the simulation path for a general switch of that type. However, given a particular switch S, the definition of the simulation path complete determines whether or not these "optional" switches are required.

Neutral switch	$\longrightarrow$	[BΔ] NS [BR]
Disconnecting switch	$\longrightarrow$	[BΔ] DHK DS [BR] [BR]
Reconnecting switch	$\longrightarrow$	[BΔ] [BΔ] RS RHK [BR]

Table 1: Simulation path for a general switch $S \in \Omega_{\mathrm{S}}$ using connected switches.

4.1.1 The simulation path for a neutral switch

Suppose that the general switch S is a *neutral switch*. The simulation path for S is formed using the following procedure.

- Let H_i be the component of W that contains all vertices involved in S.
- If the bridge edge $v_i v'_i$ is switched away by S then perform a bridge change.

- After the bridge change, if any, perform a connected switch NS which switches in and out the the edges specified by S.
- If a bridge edge is unhealthy after this switch then perform a bridge rectification.

4.1.2 The simulation path for a disconnecting switch

Suppose that the general switch S is a *disconnecting switch*. The simulation path for S is formed using the following procedure.

- Let H_j contain the 2-edge-cut ab, cd and let B_1, B_2 be the subgraphs of H_j separated by this cut. Without loss of generality, suppose that B_1 contains v_j.
- If a bridge change is required in H_j then perform the bridge change. In that case, $v_j v_j'$ is one of the edges forming the 2-edge-cut, so all other neighbours of v_j must lie in B_1. Therefore the new exit vertex v_j^* also belongs to B_1.
- After the bridge change, if any, notice that switching in and out the edges specified by S would split H_j into two components B_1 and B_2. Of these, B_1 is already in the correct place in the chain, but B_2 is not connected to the chain at all. This is not allowed, since we are restricted to connected switches. Therefore we must first perform the *disconnected housekeeping switch* (DHK) to place B_2 into the correct position in the chain. This switch is described in more detail below.
- After the DHK has been performed, we perform the disconnecting switch specified by S. (This is now safe, as the DHK ensures that the resulting graph is not disconnected.)
- Up to two bridge edges may now be unhealthy. If so, then perform up to two bridge rectification switches. For definiteness, if two BR switches are required, perform the one with the lower component index first.

It remains to specify the disconnected housekeeping switch DHK. See Figure 10. Let v_+ be the vertex in B_2 with the highest label and let v_+' be its greatest neighbour. Then $e_+ = v_+ v_+'$ must become the bridge edge of B_2. Find i such that $v_i < v_+ < v_{i+1}$, so B_2 belongs in the chain between H_i and H_{i+1} in the chain. (Note that v_k is the highest-labelled vertex in the graph, so $v_+ \leq v_k$, but it is possible that $v_+ < v_1$, in which case B_2 must become the first component in the chain and $v_k' v_+$ must become the new loopback edge.) Then DHK is the switch $(v_+ v_+', v_i' v_{i+1} \Rightarrow v_+ v_i', v_+' v_{i+1})$, which inserts B_2 into the correct position in the chain as required.

4.1.3 The simulation path for a reconnecting switch

Suppose that the general switch S is a reconnecting switch. The simulation path for S is defined using the following procedure.

- Suppose that $ab \in H_i$ and $cd \in H_j$ where, without loss of generality $v_i < v_j$. Up to two bridge changes may be necessary, as this switch involves two components. If both are necessary, perform the bridge change in H_i first.

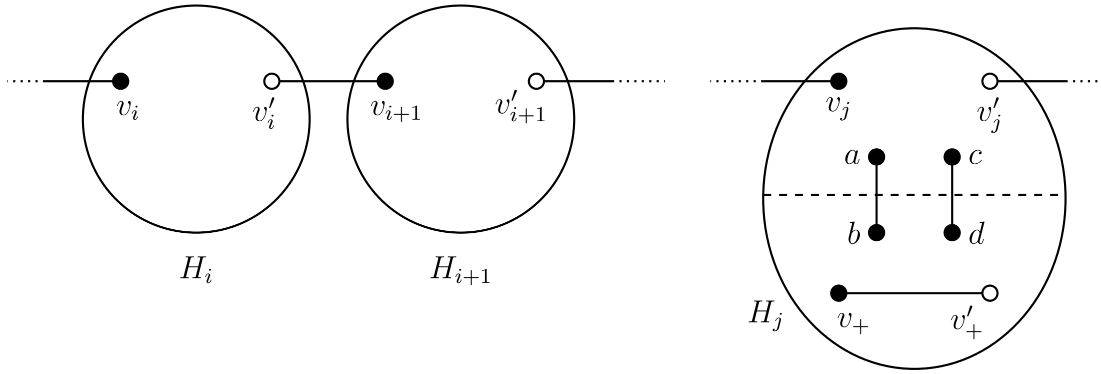


Figure 10: The situation just before a disconnecting housekeeping switch (DHK).

- Then perform the *reconnecting switch* RS that switches in and out the edges specified by S. At this point, the component formed by linking H_i and H_j is incident with four chain edges, so the chain structure is not correct. In effect, this new component is occuping both position i and position j in the chain. Since $v_i < v_j$, vertex v_j must be the entry vertex of the new component, so the correct position for this component is between H_{j-1} and H_{j+1}.
- The *reconnecting housekeeping* switch (RHK) $(v_i v'_{i-1}, v'_i v_{i+1} \Rightarrow v_i v'_i, v'_{i-1} v_{i+1})$ restores the chain by placing the new component in position between H_{j-1} and H_{j+1}, linking H_{i-1} directly to H_{i+1} (which is now the next component in the chain) and reinstating the bridge edge $v_i v'_i$. Here v'_i and v'_{i-1} refer to endvertices of the *current* chain edges, *after* the at most two bridge changes have been performed. (These may differ from the endvertices of the original chain edges, before the bridge rectification steps.)
- Finally, at most one bridge rectification may be required to ensure that the single bridge edge $v_j v'_j$ is healthy.

In the special case when the reconnecting switch $S = (W, W')$ results in a connected graph W', we must have $i = 1$ and $j = 2$: here the reconnecting housekeeping switch reinstates *both* bridge edges, and no bridge rectification is required.

4.2 Bounding the congestion

Let us calculate upper bounds for $B(\Sigma_{\mathrm{SC}})$ and $\ell(\Sigma_{\mathrm{SC}})$. From Table 1, the longest possible simulation path consists of five connected switches, so

$$\ell(\Sigma_{\mathrm{SC}}) = 5. \tag{4.1}$$

We spend the rest of this subsection bounding $B(\Sigma_{\mathrm{SC}})$.

Each connected switch $t = (Z, Z')$ can play a number of roles in a simulation path. We consider each kind of general switch separately (neutral, disconnecting, reconnecting). As in Section 3.2, the number of simulation paths (of a given kind) containing t is equal to the amount of extra information required to identify a general switch S of this kind uniquely, given t (and assuming that t belongs to the simulation path for S): we speak of "guessing" this information. Again, it is enough to identify one of W or W', together with the edges switched in/out by S, and then the other graph (W' or W) can be inferred.

4.2.1 Neutral switches

A connected switch $t = (Z, Z')$ in $\mathcal{M}_{\mathrm{SC}}$ could take on three possible roles when simulating a neutral switch $S = (W, W')$. We consider each separately below.

(**B**Δ) First, suppose that the connected switch t is a bridge change in a simulation path for a neutral (general) switch S.

- Since t is a bridge change, one of the edges which is switched in by t is also an edge which is switched out by S. Guess which one, from 2 alternatives.
- The other edge which is switched out by S is an edge of Z', so this edge and its orientation can be guessed from fewer than $2rn$ alternatives.
- Also guess whether W is connected and if not, guess the oriented loopback edge from at most $2rn$ alternatives (as in the proof of Lemma 4.2). If W is connected then $W = Z$. Otherwise, we must apply the unchaining algorithm from Lemma 4.2 to the graph Z with respect to the chosen oriented loopback edge. This produces the graph W, and then using the switch edges we can determine W'.

In all we guessed S from $2 \cdot (2rn)^2 = 8r^2n^2$ alternatives.

(**NS**) Next, suppose that the connected switch t is a neutral simulating switch for a neutral (general) switch S.

- The edges switched in/out by t are precisely the edges switched in/out by S. This gives us the vertices $\{a, b, c, d\}$ involved in the switch. Guess the (current) oriented loopback edge in Z from at most $2rn$ alternatives, and perform the unchaining algorithm to reveal the chain structure of Z. In particular, we now know the entry vertex and (current) exit vertex of every component.
- If t was not preceded by a bridge change then $Z = Z_0$. Otherwise, a bridge change has been performed before t, which implies that $Z = Z_1$. In this case, one of the edges switched out by t was originally a bridge edge in W, and hence is incident with v_i for some component H_i of W. Since S is a neutral switch, all vertices involved in the switch belong to the same component H_i, and hence v_i has the largest label among a, b, c, d. Without loss of generality, if $a = v_i$ then $b = v_i'$, while our knowledge of the chain structure of Z gives us the current exit vertex v_i^*. Hence we can uniquely reverse the bridge change switch to determine Z_0. Therefore we can find Z_0 by guessing between just two alternatives, namely, whether or not a bridge change was performed.

  From Z_0 we obtain W by deleting the chain edges and reinstating the bridge edges (taking care to adjust the exit vertex of H_i if a bridge change has been performed). Since we know W and the switch edges, we can determine W'.

We have guessed S from $4rn$ alternatives, when t is a neutral simulating switch. (The above analysis holds even if the switch S replaces one 2-edge-cut in W by another.)

(**BR**) Finally, suppose that t is a bridge rectification switch in a simulation path for a neutral (general) switch S. Then t is the final switch in the simulation path.

- Guess the oriented loopback edge in Z', from at most $2rn$ alternatives, and perform the unchaining algorithm on Z' to produce W'. (Again, this includes the possibility that W' is connected, in which case $W' = Z'$.)
- Since t is a bridge rectification, the switch S switches in an edge which is incident with the entry vertex v_i of H_i, giving it a new neighbour with a higher label than the current exit vertex in H_i. This edge is one of the two edges switched out by t: guess which one, from 2 alternatives.
- The other edge which is switched in by S, together with its orientation, can be guessed from the (oriented) edges of Z: at most $2rn$ possibilities. Now W can be inferred from W' and the switch edges.

We have guessed S from at most $2rn \cdot 2 \cdot 2rn = 8r^2n^2$ alternatives.

Adding the contributions from these three roles, a connected switch t may be part of at most

$$8r^2n^2 + 4rn + 8r^2n^2 \leq (8r^2 + 1)n^2 \tag{4.2}$$

simulation paths for neutral general switches.

Before we move on to consider disconnecting and reconnecting switches, we make a couple of comments. The analysis of the neutral simulation switch (NS) shows that once the edges of the general switch S have been guessed, we have enough information to decide whether a bridge rectification switch is required and if so, its specification. Since these bridge rectification switches do not affect the number of choices for S in each subcase below, we only mention them when t itself is playing the role of a bridge rectification, and otherwise we omit them from our accounting.

In every situation we must guess the oriented loopback edge, so that we can perform the unchaining algorithm. In some cases we must also be careful to specify which graph is given as input to the unchaining algorithm. In particular, we must not perform the unchaining algorithm immediately after a disconnecting housekeeping switch or immediately before a reconnecting housekeeping switch, since in those graphs the chaining structure has been temporarily compromised.

4.2.2 Disconnecting switches

A connected switch $t = (Z, Z')$ in $\mathcal{M}_{\mathrm{SC}}$ could take on five possible roles when simulating a disconnecting switch $S = (W, W')$. We consider each separately below.

(**BΔ**) If t is a bridge change then we can guess S from at most $8r^2n^2$ alternatives, as explained in Section 4.2.1.

(**DHK**) Next, suppose that t is a disconnecting housekeeping switch.

- Guess the (current) oriented loopback edge as an edge of Z from at most $2rn$ possibilities. Perform the unchaining algorithm on Z to reveal the chain structure of Z. Again, this tells us the entry and (current) exit vertex of every component of W.
- Decide whether or not a bridge change was performed before t, from 2 possibilities. As described in Section 4.2.1, we can uniquely determine the bridge change from the chain structure of Z, so in both cases we can find Z_0 (note that $Z = Z_0$ if there was no bridge change, and $Z = Z_1$ if a bridge change was performed before t). By reversing the bridge change if necessary we obtain Z_0, and then from our knowledge of the chain structure we can determine W. (If a bridge change was performed then we must use the original exit vertex for that component: this vertex is revealed by reversing the bridge change.)

  Since t is a disconnected housekeeping switch, it switches out a chain edge $v'_j v_{j+1}$, for some j, and an edge $v_+ v'_+$ which is entirely contained within a component H_i, for some $i \neq j$. Hence we can distinguish between these two edges of t and identify which is the chain edge and which is $v_+ v'_+$, also determining i and j. Furthermore, we can identify v_+ uniquely as it has a higher label than v'_+.
- Let v_i be the entry vertex in H_i. A disconnected housekeeping switch is only performed when the disconnected simulation flip (DS) representing S would switch out the two edges of a 2-edge-cut separating v_+ and v_i, say. By Lemma 1.2(iv), we can guess which 2-edge-cut is switched out by S from at most $n^2/(15r^2)$ possibilities. The orientation of the switch S is uniquely determined, since there is only one possibility which would disconnect H_i. Now we know W and the switch edges, we can determine W'.

We have guessed S from at most $2rn \cdot 2 \cdot n^2/(15r^2) = 4n^3/(15r)$ possibilities.

(**DS**) Now suppose that t is a disconnecting simulation switch. The edges switched in/out by t are precisely the edges switched in/out by S, so we do not need to guess these. Note that Z does not have a valid chain structure, but Z' does.

- We know that W' is not connected, so we guess the oriented loopback edge from the (oriented) edges of Z': at most $2rn$ possibilities. Perform the unchaining algorithm on Z', revealing the chain structure of Z'. Suppose that H_i and H_j are the two components of Z' which contain an edge which was switched in by t, and suppose that $i < j$. Then the disconnecting housekeeping switch (DHK) is uniquely determined: it is the switch $(v_i v'_i, v'_{i-1} v_{i+1} \Rightarrow v_i v'_{i-1}, v'_i v_{i+1})$.
- Guess whether or not a bridge change was performed before the disconnecting housekeeping switch: if so, it is uniquely specified (as explained earlier), so we guess from 2 possibilities. Reverse this bridge change to obtain Z_0 and hence find W. From W and the switch edges we can determine W'.

We have guessed the general switch S from at most $4rn$ possibilities.

(**The first BR**) Now suppose that t is the first bridge rectification switch in the simulation path for a disconnecting (general) switch S. There may or may not be a second bridge rectification switch after t. The calculations are similar to the bridge rectification case in Section 4.2.1.

- Guess the oriented loopback edge in Z', from at most $2rn$ alternatives, and perform the unchaining algorithm to reveal the chain structure of Z'. (Note, we know that W' is disconnected when S is a disconnecting switch.)
- Since t is a bridge rectification switch, one of the edges switched in by S is $v_i v_i^*$, where v_i is the entry vertex of some component H_i and v_i^* has a higher label than the current exit vertex v_i' in that component. This edge is then switched out by t. (It is possible that this condition is satisfied by both edges of S, but if so, we know that the first bridge rectification will act on the component H_i with the lower index first.) Guess which edge of t is $v_i v_i^*$, for a factor of 2.
- Guess the other edge switched out by S, and the orientation of the switch, from at most $2rn$ possibilities. (The other edge switched out by S must lie in a distinct component from H_i, but for an upper bound we ignore this.) Whether or not a second bridge change will be needed after t is completely determined by the switch edges and the chain structure, as described in Section 4.2.1. If a second bridge rectification is needed, perform it to produce Z_q. From Z_q we may determine W' using our knowledge of the chain structure of Z' (and adjusting the exit vertex of the component in which the second bridge rectification was performed, if necessary). From W' and the switch edges we may deduce W.

We have guessed S from at most $2rn \cdot 2 \cdot 2rn = 8r^2n^2$ alternatives.

(**The second BR**) Here we assume that one bridge rectification has been already performed to produce Z. The connected switch t is the last one in the simulation path.

- Again, we guess the oriented loopback edge for Z', from at most $2rn$ possibilities. Apply the unchaining algorithm to produce W'.
- Choose which edge switched out by t is an edge switched in by S, out of 2 possibilities. The endvertex of this edge with the higher label is v_i, for some i.
- Since this is the second bridge rectification, the other edge switched in by S was $v_j v_j^*$ for some other component H_j, and this edge has been made into the bridge edge in H_j, using the first bridge rectification. Therefore we can identify and orient this edge once we guess H_j, from at most n/r possibilities (using Lemma 1.2(iii)). This determines the edges switched in by S. From W' and the switch edges, we can determine W.

We have guessed S from at most $2rn \cdot 2 \cdot n/r = 4n^2$ possibilities.

Summing over these five roles, a connected switch t can be included in at most

$$8r^2n^2 + 4n^3/(15r) + 4rn + 8r^2n^2 + 4n^2 \le 9rn^3 \tag{4.3}$$

simulation paths for disconnecting general switches. (The upper bound follows since $2 \le r \le n/2$.)

4.2.3 Reconnecting switches

A connected switch $t = (Z, Z')$ in $\mathcal{M}_{\text{SC}}$ could take on give possible roles when simulation a reconnecting (general) switch $S = (W, W')$. We consider each separately below.

(**The first BΔ**) When t is the first bridge change we can guess S from at most $8r^2n^2$ alternatives, using arguments very similar to those given in Section 4.2.1.

(**The second BΔ**) Now suppose that t is the second of two bridge changes in the simulation path for some reconnecting (general) switch S.

- Guess the oriented loopback edge in Z, from at most $2rn$ possibilities. Perform the unchaining algorithm to reveal the chain structure of Z.
- Since t is a bridge change, one of the edges switched in by t is an edge which will be switched out by S. Choose which, from 2 possibilities. This edge is incident with the entry vertex v_j of some component H_j.
- Since t is the second bridge change, the other edge which will be switched out by S must be incident with the entry vertex of some other component H_i, with $i < j$. (It has been put back into Z by the first bridge change.) By Lemma 1.2(iii) there are at most $n/(2r)$ components, so we choose one to determine v_i.
- Choose a non-chain edge incident with v_i and orient it, from $2(2r-1)$ possibilities. This determines the edges switched in and out by S. From these edges and W, we may determine W'.

We have guessed S from at most $2rn \cdot 2 \cdot n/(2r) \cdot 2(2r-1) < 8rn^2$ possibilities.

(**RS**) Suppose that t is a reconnecting simulation switch. The edges switched in/out by t are precisely the edges switched in/out by S.

- Guess the oriented loopback edge for Z from at most $2rn$ possibilities. Perform the unchaining algorithm on Z to reveal the chain structure of Z.
- Up to two bridge changes may have been performed before t, corresponding to the up to two edges switched out by S which are incident with entry vertices in some component of Z. If two bridge changes have been performed then their order is uniquely determined. As argued in Section 4.2.1, we must just decide whether or not a bridge change has been performed in each component, and then the rest is specified, so there are $2^2 = 4$ possiblities for the bridge changes before t, including the possibility that there were none. Once the bridge change switches are known, they can be reversed, which reveals W. Together with the switch edges, this determines W' and hence S.

We have guessed S from at most $8rn$ possibilities.

(**RHK**) Suppose that t is a reconnecting housekeeping switch. Then Z' has a valid chain structure, though Z does not.

- Guess the oriented loopback edge for Z' from at most $2rn$ possibilities. Perform the unchaining algorithm on Z' to reveal the chain structure of Z'.

- The switch t involves entry vertices v_i and v_j, where $i < j$. Since t is a reconnecting housekeeping switch, the edges switched in by S form a 2-edge-cut in Z' which separate v_i and v_j. By Lemma 1.2(iv) we can guess this 2-edge-cut from at most $n^2/(15r^2)$ possibilities. This specifies the edges switched out by S. The orientation of the switch is determined by the fact that one edge switched out by S is contained in H_i and the other is contained in H_j.
- Finally, we must decide whether or not a bridge rectification is needed to produce W', giving 2 possibilities. If a bridge rectification is needed then it is uniquely determined: perform it to obtain W', and then W can be obtained using the switch edges.

We have guessed S from a total of at most $2rn \cdot n^2/(15r^2) \cdot 2 = 4n^3/(15r)$ possibilities.

(**BR**) Suppose that t is a bridge rectification for a reconnecting (general) switch S. Then t is the final switch in the simulation path. In particular, W' is disconnected, since otherwise the reconnecting housekeeping switch produces W', and no BR or unchaining is necessary.

- Guess the oriented loopback edge and perform the unchaining algorithm in Z' to produce W', from at most $2rn$ alternatives.
- One of the edges switched in by S is an edge which was switched out by t: choose one, from 2 possibilities, and call it e.
- Let $\widehat{e}$ be the other edge which is switched in by S. Recall that the edges removed by the RHK are $e_{i-1} = v_i v'_{i-1}$ and $e_i = v'_i v_{i+1}$, where e'_i, e'_{i-1} are the chain edges in the graph Z_q obtained *after* the (at most two) bridge changes were performed. (See Section 4.1.3.) Let Z_{q+1} be the result of performing the RS simulating S, starting from Z_q. Since the edges switched in by the RS must be disjoint from the edges of Z_q, we conclude that $\widehat{e} \notin \{e'_{i-1}, e'_i\}$: the three edges $\widehat{e}$, e'_{i-1}, e'_i are distinct and all belong to Z_{q+1}. Next, the RHK (Z_{q+1}, Z_{q+2}) removes precisely two edges, namely e'_{i-1}, e'_i, and hence $\widehat{e}$ is still present in $Z = Z_{q+2}$. Therefore we can guess and orient $\widehat{e}$ from among the $2rn$ oriented edges of Z. (In fact, the two edges switched in by S form a 2-edge-cut in Z, but we do not use that fact here.) Now we know W' and the switch edges, we can determine W.

We have guessed S from at most $2rn \cdot 2 \cdot 2rn = 8r^2n^2$ possibilities.

Summing the contribution from these five roles, a connected switch t can be included in at most

$$8r^2n^2 + 8rn^2 + 8rn + 4n^3/(15r) + 8r^2n^2 \leq 11rn^3 \tag{4.4}$$

simulation paths for reconnecting general switches. (The upper bound follows since $2 \leq r \leq n/2$.)

4.3 Completing the analysis

Now we apply the two-stage direct canonical path construction to our set of $(\mathcal{M}_{\mathrm{SC}}, \mathcal{M}_{\mathrm{S}})$-simulation paths. Here $\bar{\rho}(\Gamma_{\mathrm{S}})$ denotes the congestion of any set of canonical paths, or multi-commodity flow, for the switch chain. We will use the bound given in [4].

Lemma 4.3. *The* $(\mathcal{M}_{\mathrm{SC}}, \mathcal{M}_{\mathrm{S}})$*-simulation paths defined in Section 4.1 define a set of canonical paths for* $\mathcal{M}_{\mathrm{SC}}$ *with congestion* $\bar{\rho}(\Gamma_{\mathrm{SC}})$ *which satisfies*

$$\bar{\rho}(\Gamma_{\mathrm{SC}}) \leq 120\, rn^3\, \bar{\rho}(\Gamma_{\mathrm{S}}).$$

Proof. First, by adding together (4.2) – (4.4), the maximum number $B(\Sigma_{\mathrm{SC}})$ of simulation paths containing a given connected switch satisfies

$$B(\Sigma_{\mathrm{SC}}) \leq (8r^2+1)n^2 + 9rn^3 + 11rn^3 \leq 24rn^3,$$

using the fact that $r \geq 2$ and $n \geq 2r+1 \geq 5$. Next we calculate the simulation gap $D_{\mathrm{SC}} = D(\mathcal{M}_{\mathrm{SC}}, \mathcal{M}_{\mathrm{S}})$. Recall that for all $uv \in E(\mathcal{M}_{\mathrm{SC}})$ and $zw \in E(\mathcal{M}_{\mathrm{S}})$ we have $u \neq v$ and $z \neq w$, and hence

$$P_{\mathrm{SC}}(u,v) = P_{\mathrm{S}}(z,w) = \frac{1}{3a_{n,r}},$$

by definition of both chains. The state space of $\mathcal{M}_{\mathrm{S}}$ is Ω_{S} and the state space of $\mathcal{M}_{\mathrm{SC}}$ is Ω_{F}, with $\Omega_{\mathrm{F}} \subseteq \Omega_{\mathrm{F}}$. Since both $\mathcal{M}_{\mathrm{S}}$ and $\mathcal{M}_{\mathrm{SC}}$ have uniform stationary distribution,

$$D_{\mathrm{SC}} = \max_{\substack{uv \in E(\mathcal{M}_{\mathrm{SC}}) \\ zw \in E(\mathcal{M}_{\mathrm{S}})}} \frac{|\Omega_{\mathrm{F}}|\, P_{\mathrm{S}}(z,w)}{|\Omega_{\mathrm{S}}|\, P_{\mathrm{SC}}(u,v)} \leq \max_{\substack{uv \in E(\mathcal{M}_{\mathrm{SC}}) \\ zw \in E(\mathcal{M}_{\mathrm{S}})}} \frac{P_{\mathrm{S}}(z,w)}{P_{\mathrm{SC}}(u,v)} = 1.$$

Substituting these values and (4.1) into Theorem 2.1 gives

$$\begin{aligned} \bar{\rho}(\Gamma_{\mathrm{SC}}) &\leq D_{SC}\, \ell(\Sigma_{SC})\, B(\Sigma_{SC})\, \bar{\rho}(\Gamma_{\mathrm{S}}) \\ &\leq 120\, rn^3\, \bar{\rho}(\Gamma_{\mathrm{S}}), \end{aligned}$$

as required. □

Finally, we may prove Theorem 1.1.

Proof of Theorem 1.1. Combining the bounds of Lemma 3.5 and Lemma 4.3 gives

$$\frac{\bar{\rho}(\Gamma_{\mathrm{F}})}{\bar{\rho}(\Gamma_{\mathrm{S}})} \leq 480\,(2r)^{12} n^7. \tag{4.5}$$

The upper bound on the mixing time of the switch chain given in [4, 5] has (an upper bound on) $\bar{\rho}(\Gamma_{\mathrm{S}})$ as a factor. Therefore, multiplying this bound by the right hand side of (4.5), we conclude that the mixing time of the flip chain is at most

$$480\,(2r)^{35}\, n^{15}\, \left(2rn\log(2rn) + \log(\varepsilon^{-1})\right),$$

completing the proof of Theorem 1.1. (This uses the fact that $|\Omega_S| \leq (rn)^{rn}$, which follows from known asymptotic enumeration results [2].) □

It is unlikely that the bound of Theorem 1.1 is tight. Experimental evidence was presented in an earlier version of this work [6] which provides support for the conjecture that the true mixing time for the flip chain is $\mathrm{O}(n\log n)$, when r is constant. Perhaps $O(rn\log n)$ is a reasonable conjecture when r grows with n.

A Maximising 2-edge-cuts separating two vertices

We now present the deferred proof of Lemma 1.2(iv). Our aim here is to present a simple upper bound on $\Lambda(u,v)$ which holds for all relevant values of n and r, though our proof establishes tighter bounds than the one given in the statement of Lemma 1.2(iv).

Proof of Lemma 1.2(iv). Let $\Lambda(u,v)$ denote the number of 2-edge-cuts in G which separate u and v. By considering the connected component of G which contains u, if necessary, we may assume that G is connected. Consider the binary relation $\sim$ on V defined by $w_1 \sim w_2$ if and only if there are at least three edge-disjoint paths between w_1 and w_2 in G. Then $\sim$ is an equivalence relation [19] which partitions the vertex set V into equivalence classes $U_0,\ldots,U_k$. For $j=0,\ldots,k$ let $H_j = G[U_j]$ be the subgraph of G induced by U_j. Each H_j is either a maximal 3-edge-connected induced subgraph of G, or a single vertex. Without loss of generality, suppose that $u \in H_0$ and $v \in H_k$.

Note that the number of edges from H_i to H_j is either 0, 1 or 2 for all $i \neq j$, by construction. Define the *node-link* multigraph $\widetilde{G}$ of G by replacing each H_j by a single vertex h_j, which we call a *node*, and replacing each edge from a vertex of H_i to a vertex of H_j by an edge from h_i to h_j, which we call a *link*. In particular, if there is a 2-edge-cut from H_i to H_j in G then the link h_ih_j has multiplicity 2 in $\widetilde{G}$. Each node has even degree in $\widetilde{G}$, by Lemma 1.2(i). Furthermore, every link in $\widetilde{G}$ belongs to a cycle (possibly a 2-cycle, which is a double link), by Lemma 1.2(ii), and these cycles in $\widetilde{G}$ must be edge-disjoint, or the corresponding edge of G cannot be part of a 2-edge-cut. Therefore $\widetilde{G}$ is a planar multigraph (with edge multiplicity at most two) which has a tree-like structure, as illustrated in Figure 11: the black squares represent the nodes of $\widetilde{G}$. A node of $\widetilde{G}$ which belongs to more than one cycle is a *join node*.

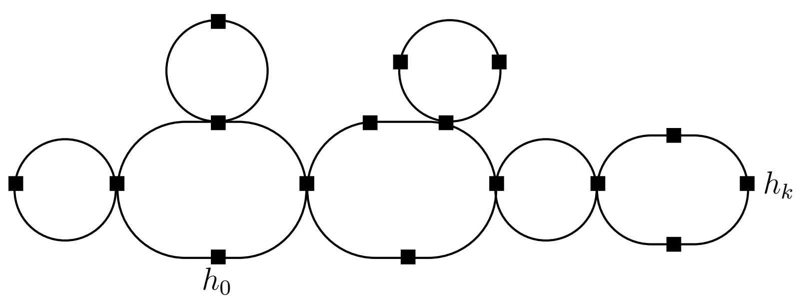


Figure 11: The node-link graph $\widetilde{G}$

The join nodes each have degree 4, and all other nodes of $\widetilde{G}$ have degree two.

Since we seek an upper bound on $\Lambda(u,v)$ for a given number of vertices n, we may assume that all nodes of $\widetilde{G}$ lie on some path from h_0 to h_k in $\widetilde{G}$. (This corresponds to trimming all "branches" of $\widetilde{G}$ which do not lie on the unique "path" from h_0 to h_k in $\widetilde{G}$, viewed as a tree.) This only removes 2-edge-cuts of G which do not separate u and v. Hence for an upper bound, by relabelling the nodes if necessary, we can assume that $\widetilde{G}$ consists of ℓ cycles $C_1,\ldots,C_\ell$, where h_0 belongs only to C_1 and h_k belongs only to C_ℓ, such that C_j and C_{j+1} intersect at a single join node h_j, for $j = 1, \ldots, \ell - 1$, while all other cycles are disjoint. This situation is illustrated in Figure 12. (In particular, h_0 and h_k are not join nodes.)

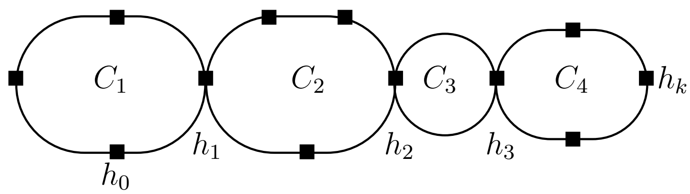


Figure 12: The node-link graph after trimming: an example with $\ell = 4$ cycles.

Each cycle C_j can be considered as the disjoint union of two paths between h_{j-1} and h_j, or between $h_{\ell-1}$ and h_k if $j = \ell$. Denote by a_j, a'_j the lengths of these two paths around C_j. Any 2-edge-cut which separates u from v in G corresponds to a pair of links in $\widetilde{G}$ which both belong to some cycle C_j, with one link from each of the two paths which comprise C_j. Hence $\Lambda(u,v) = \sum_{j=1}^{\ell} a_j a'_j$. If the length $a_j + a'_j$ of each cycle is fixed then the quantity $a_j a'_j$ is maximised when $|a_j - a'_j| \leq 1$. Therefore $\Lambda(u,v) \leq \sum_{i=1}^{\ell} s_i^2$, where $s_j = |C_j|/2 \geq 1$. (This upper bound holds even when some cycles have odd length.) Now, the total number of nodes in $\widetilde{G}$ is $N = 2\left(\sum_{j=1}^{\ell} s_j\right) - (\ell - 1)$, by inclusion-exclusion. Thus, for a given number of nodes N and a given value of ℓ, the bound S is maximised by setting $s_j = 1$ for $j = 2, \ldots, \ell$, and $s_1 = (N - \ell + 1)/2$. Therefore

$$\Lambda(u,v) \leq T = \frac{(N - \ell + 1)^2}{4} + \ell - 1. \tag{A.1}$$

First suppose that $r \geq 3$. Then every subgraph H_j must contain a vertex which is only adjacent to other vertices of H_j, since every node in $\widetilde{G}$ has degree at most 4. Therefore each H_j must contain at least $2r + 1$ vertices, so there are $N \leq n/(2r+1)$ nodes in $\widetilde{G}$. Note that $\mathrm{d}T/\mathrm{d}\ell = -\ell + 1$ which equals zero when $\ell = 1$ and is negative when $\ell > 1$. So T is largest when $\ell = 1$, giving $\Lambda(u,v) \leq T \leq N^2/4 \leq n^2/(4r+2)^2$, as claimed. An extremal graph G is shown in Figure 13: each grey block is K_{2r+1} minus an edge. This construction gives an extremal example for any positive $n = 0 \bmod (4r+2)$, so this upper bound is asymptotically tight. Certainly the weaker bound $\Lambda(u,v) \leq n^2/(15r^2)$ also holds when $r \geq 3$.

Now suppose that $r = 2$ and consider the induced subgraphs $H_0, \ldots, H_k$. Again we may assume that $s_2 = \cdots = s_\ell = 1$, giving (A.1). If h_j is a join node of $\widetilde{G}$ then H_j may be as small as a single vertex, since each join node of $\widetilde{G}$ has degree 4. We call such a join node h_j a *singleton*. Otherwise, if H_j contains a vertex whose neighbourhood is contained within H_j then H_j contains at least $2r + 1 = 5$ vertices. In particular, this must hold for h_k and for all nodes in C_1 except h_1. A final option (only available when $r = 2$) is that a join node h_j corresponds to to a subgraph H_j which is isomorphic to K_4, and hence has precisely four vertices: two of these vertices are joined to a vertex in H_{j-1} and two are joined to H_{j+1} (or H_k, if $j = \ell - 1$). Here we use the fact that any part U_j of the partition of V with more than one vertex must have at least 4 vertices, as $G[U_j]$ must be 3-edge-connected. Since G has no

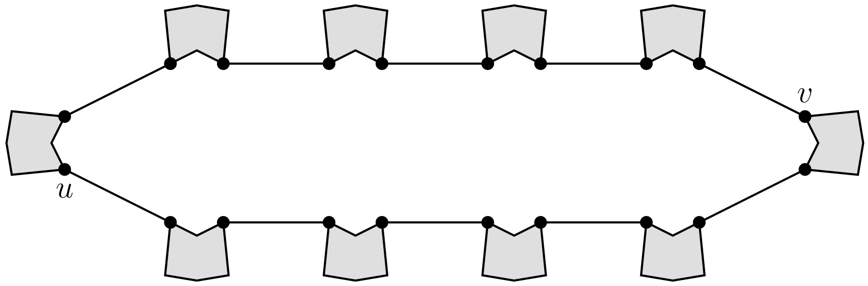


Figure 13: Extremal graph when $r \geq 3$.

repeated vertices, at most every second join node can be a singleton, with the remaining join nodes contributing at least 4 vertices to G.

It follows that $\Lambda(u,v) \leq T = s_1^2 + \ell - 1$ and

$$n \geq 10s_1 + \left\lceil \frac{\ell-1}{2} \right\rceil + 4\left\lceil \frac{\ell-1}{2} \right\rceil = \begin{cases} 10s_1 + 5(\ell-1)/2 & \text{if } \ell \text{ is odd,} \\ 10s_1 + 5\ell/2 - r & \text{if } \ell \text{ is even.} \end{cases}$$

For an upper bound on $\Lambda(u,v)$ we take equality the above expression for n and assume that ℓ is even, leading to $\Lambda(u,v) \leq T = s_1^2 - 4s_1 + 2(n+4)/5$. (If ℓ is odd then only the term which is independent of s_1 changes.) Hence $\mathrm{d}T/\mathrm{d}s_1$ is negative when $s_1 = 1$, zero when $s_1 = 2$ and positive for $s_1 \geq 3$. It follows that for a given value of n, the maximum possible value of T occurs when $s_1 = 1$ or when s_1 is as large as possible.

When $s_1 = 1$ we have $\Lambda(u,v) \leq \ell \leq 2(n-6)/5$, at least when ℓ is even. A graph G on n vertices which meets this bound can be constructed whenever $n \geq 11$ and $n = 1 \bmod 5$. The example shown in Figure 14 has $n = 21$ vertices and $\ell = 6$ 2-edge-cuts between u and v. Since $2(n-6)/5 \leq n^2/60$ for all values of n, this establishes the required bound.

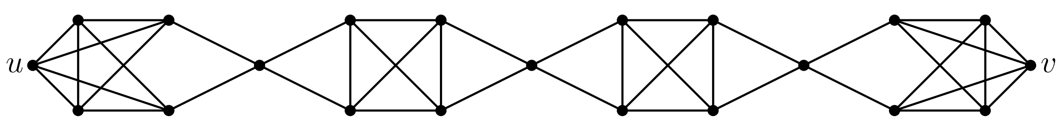


Figure 14: Maximum number of separating 2-edge-cuts for $n = 21$

Next, consider the subcase when s_1 is as large as possible, for a fixed value of n. This is achieved by setting $\ell = 1$, so there are no join nodes. Then every H_j has at least 5 vertices, so for an upper bound we take $s_1 = n/10$ and $\Lambda(u,v) \leq s_1^2 = n^2/100 < n^2/60$. This completes the proof. □

References

[1] Z. Allen-Zhu, A. Bhaskara, S. Lattanzi, V. Mirrokni and L. Orecchia, Expanders via local edge flips, in *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, New York–Philadelpha, 2016, pp. 259–269.

[2] E. Bender and E.R. Canfield, The asymptotic number of labelled graphs with given degree sequences, *Journal of Combinatorial Theory (Series A)* **24** (1978), 296–307.

[3] V. Bourassa and F. Holt. Swan: Small-world wide area networks. In *Proceedings of International Conference on Advances in Infrastructure (SSGRR 2003w)*, L'Aquila, Italy, 2003. Paper #64.

[4] C. Cooper, M. Dyer, and C. Greenhill. Sampling regular graphs and a peer-to-peer network. *Comb. Probab. Comput.*, 16(4):557–593, 2007.

[5] C. Cooper, M.Dyer and C. Greenhill, Corrigendum: Sampling regular graphs and a peer-to-peer network, 2012. arXiv:1203.6111

[6] C. Cooper, M. Dyer, and A. Handley. The flip Markov chain and a randomising P2P protocol. In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing*, ACM Press, New York, 2009.

[7] P. Diaconis and L. Saloff-Coste. Comparison theorems for reversible Markov chains. *Annals of Applied Probability*, 3:696–730, 1993.

[8] R. Diestel, *Graph Theory (4th edn.)*, Springer, Berlin, 2010.

[9] T. Feder, A. Guetz, M. Mihail, and A. Saberi. A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks. *FOCS*, 2006.

[10] V. Guruswami. Rapidly mixing Markov chains: A comparison of techniques (a survey). Unpublished manuscript, 2000. arXiv:1603.01512

[11] A.J. Handley, Self-organising, self-randomising peer-to-peer networks, Ph.D. thesis, University of Leeds, 2013.

[12] M. Jerrum. *Counting, Sampling and Integrating: Algorithms and Complexity.* Birkhäuser, 2003.

[13] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

[14] R. Kannan, P. Tetali, and S. Vempala. Simple markov-chain algorithms for generating bipartite graphs and tournaments. *Random Struct. Algorithms*, 14(4):293–308, 1999.

[15] J. Petersen, Die Theorie der regulären graphs, *Acta Mathematica* **15** (1891), 193–220.

[16] P. Mahlmann and C. Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *SPAA '05: Proceedings of the 17th Annual ACM Symposium on Parallelism in algorithms and architectures*, pages 155–164, New York, NY, USA, 2005. ACM.

[17] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1:351–370, 1992.

[18] R. Taylor, Constrained switching in graphs, in *Combinatorial Mathematics VIII*, Springer Lecture Notes in Mathematics vol. 884, 1981, pp.314–336.

[19] W.T. Tutte, *Connectivity in Graphs*, University of Toronto Press, Toronto, 1966.

[20] N. C. Wormald. Models of random regular graphs. In J.D. Lamb and D.A. Preece, editors, *Survey in Combinatorics*, London Mathematical Society Lecture Notes **276**. Cambridge University Press, Cambridge, 1999.