Networks of random cycles

Colin Cooper

Dept. of Computer Science
Kings College
London WC2R 2LS, UK
colin.cooper@kcl.ac.uk

Martin Dyer

School of Computing
University of Leeds
Leeds LS2 9JT, UK
dyer@comp.leeds.ac.uk

Andrew J. Handley

School of Computing
University of Leeds
Leeds LS2 9JT, UK
jobriath@comp.leeds.ac.uk

July 7, 2010

Abstract

We present a family of peer-to-peer network protocols that yield regular graph topologies having known Hamilton cycles. These topologies are equivalent, in a well-defined sense, to the *random regular graph*. As a consequence, we have connectivity deterministically, and logarithmic diameter and expansion properties with high probability.

We study the efficacy of certain simple topology-altering operations, designed to introduce randomness. These operations enable the network to self-stabilise when damaged. They resemble the operations used by Cooper, Dyer and Greenhill (2007) for a similar purpose in the case of random regular graphs.

There is a link between our protocols and certain combinatorial structures which have been studied previously, in particular *discordant permutations* and *Latin rectangles*. We give the first rigorous polynomial mixing-time bounds for natural Markov chains that sample these objects at random. We do so by developing a novel extension to the canonical path technique for bounding mixing times: *routing via a random destination*. This resembles a technique used by Valiant (1982) for low-congestion routing in hypercubes.

1 Introduction

Peer-to-peer networks are distributed, decentralised networks of peers with homogeneous responsibilities. They provide a natural and elegant means of solving large computational and storage problems when there is no clear organisational authority. Their decentralisation gives various benefits—absence of a single point of failure, greater potential for privacy, excellent scalability—but presents interesting challenges regarding organisation and maintenance. Much work [2, 9, 17, 18, 20, 24] has gone into engineering self-organising peer-to-peer protocols that yield topologies with favourable properties (low diameter, resilience to attack, low maintenance cost, high connectivity, and so on).

This paper's contribution is fourfold. We propose a family of simple network topologies, based on unions of random cycles, that are *contiguous* to random regular graphs (see section 2), and hence display the attractive properties above. We define two small, natural topology-randomising operations which act to keep the network randomised and hence can correct damage. We draw a link between these peer-to-peer networks and *discordant permutations*, and so derive the first rigorous polynomial-time mixing results on these objects. Finally, in bounding the mixing time, we introduce a novel extension to the *canonical path* technique, which resembles Valiant's low-congestion routing in hypercubes [26].

This paper is organised as follows. The remaining sections of this introduction offer discussion of self-stabilising peer-to-peer networks and discordant permutations with application to simple networks. Section 2 develops notation and techniques used throughout the rest of the paper. Section 3 specifies the family of protocols and introduces the randomising operations with some evidence of their efficacy. Section 4 bounds their running time mathematically. Finally, section 5 restricts those operations to preserve simplicity. It also contains the main theoretical contributions of this paper.

1.1 Self-stabilising peer-to-peer networks

This paper focusses on *Cycle networks* composed of the union of an outer Hamilton cycle and $r-1$ inner 2-regular *layers*. There are two kinds of Cycle network. The layers of a *Hamiltonian network* are random Hamilton cycles. The layers of a *Factor network* are random 2-factors of the complete graph, which in general are sums of disjoint cycles. As shown in Theorem 1 below, the probability spaces of Hamiltonian and Factor networks are contiguous to those of random regular graphs and random regular multigraphs (respectively). Hence, both networks have the desirable properties of random regular graphs, such as low diameter and resilience to damage.

An interesting benefit of a randomised architecture is that self-stabilisation is not a question of maintaining intricate, potentially delicate structure. Rather, too much structure is a handicap. An approach that has proven effective [3, 20] is to perform small randomising operations spontaneously upon the network in an effort to "re-randomise" and combat damage. We say that an operation *randomises* a graph in a set P when repeated application of the operation makes the graph an almost-uniform random sample from P.

To this end we employ two such randomising operations, the *switch* and the *transposition*. (See Figure 1). Both affect only a constant number of connections in the network, and both will be shown to randomise it rapidly.

1.2 Simplicity and discordance

If no effort is made to ensure that a Cycle network G is simple it is likely to have self-loops and/or parallel edges. We treat such networks below, but they clearly have properties undesirable for a peer-to-peer network. We will now consider how simplicity may be imposed on G.

As each layer of G is a set of disjoint cycles there is a natural correspondence between a layer and a permutation. Details are given in section 3. If G is viewed as a block of permutations of $[n] = \{0, 1, \ldots, n-1\}$ then parallel edges occur when a given index maps onto a given value in more than one permutation and self-loops occur precisely when index i maps onto i. Viewed

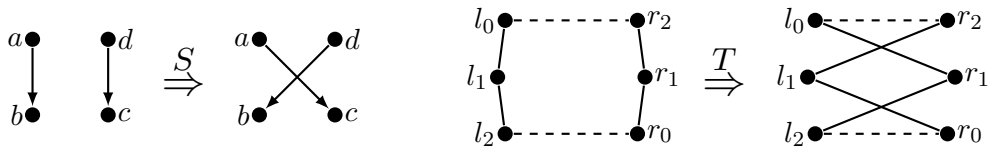

Figure 1: The randomising operations studied in this paper. **Left:** a switch. The edges ab, dc are replaced with ac, db. The edges are directed to guarantee a certain kind of switch. This is a logical artefact: the underlying communication link is bidirectional. **Right:** a transposition, which operates on a Hamilton cycle. It has the effect of swapping two vertices within that cycle.

in this manner, G is closely related to a *Latin rectangle*, which is a $q \times n$ matrix in which the elements of $[n]$ appear exactly once in each row and at most once in each column.

Permutations π and σ are *mutually discordant* when $\forall i \in [n]\ \pi(i) \neq \sigma(i)$, which is a symmetric relation. Represent a $q \times n$ Latin rectangle as $\Pi = (\pi_0, \pi_1, \ldots, \pi_{q_1})$, where the π_i's are mutually discordant. We say that each π_i is Π-*discordant*. This is by analogy to previous work in this area [22, 27] which define π as q-*discordant* if $\pi(i) \notin \{i, i+1, \ldots, i+q-1\}$ for every $i \in [n]$, where arithmetic is $\bmod n$. The two formulations are closely related. If $s_j(i) = i+j \bmod n$ then π is q-discordant precisely when Latin rectangle $\Pi = (s_0, s_1, \ldots, s_{q-1}, \pi)$. Well-known instances of discordant permutation problems include *derangements* ($\pi(i) \notin \{i\}$) and the *problème des ménages* ($\pi(i) \notin \{i, i+1\}$ [16, 25]), which are 1-discordant and 2-discordant, respectively.

These structures are not easy to reason about. They do not form permutation groups, so refined group-theoretic methods [5] are inapplicable. Exact enumeration has been accomplished for q-discordance when $q \leq 5$ [22]. Asymptotic results are known for the number of Latin rectangles with $q \in \mathrm{o}\big(n^{6/7}\big)$ [10]. Our algorithm for sampling discordant permutations is indicated only in distributed settings. Otherwise, McKay and Wormald [19] give an algorithm based on "switchings" that generates a random Latin rectangle in time $\mathrm{O}\big(nq^3\big)$ when $q \in \mathrm{o}(\sqrt[3]{n})$. It is not clear how to extend their accept/reject algorithm to a decentralised setting. Jacobson and Matthews [12] define a Markov chain that almost uniformly samples random *Latin squares* (where $r = n$). They offer only intuition as to why their chain might be rapidly mixing.

2 Preliminaries

Graph $G = (V, E)$ has vertex set $V = V(G)$ and edge set $E = E(G) \subseteq \binom{V}{2}$. For convenience we assume $V = [n] = \{0, \ldots, n-1\}$, where $n = |V|$.

An important concept in the study of random graphs is *contiguity*. Two families of random graphs are contiguous if their distributions are similar enough that properties that hold for one hold automatically for the other. Following are interesting random $2r$-regular graph families of size n: $\mathbb{G}^*(n, 2r)$ denotes a natural family of multigraphs, $\mathbb{G}(n, 2r)$ denotes uniform simple graphs, and $\mathbb{H}(n)$ denotes uniform random Hamilton cycles. For a thorough treatment see Janson [13]. We paraphrase the relevant result from there now.

Theorem 1 (From Theorem 9.32, Janson et al. [13], pp258)**.** *Let $\mathbb{G} = \mathbb{G}_1 + \cdots + \mathbb{G}_r$, where $r \geq 1$ and $\mathbb{G}_1, \cdots, \mathbb{G}_r$ are independent random regular (multi-)graphs such that $\mathbb{G}_i$ is a copy of either $\mathbb{G}^*(n, 2)$ or $\mathbb{H}(n)$. Thus $\mathbb{G}$ is a $2r$-regular multigraph. Let $\mathbb{G}'$ be another random graph family formed in this way. Then properties true of $\mathbb{G}$ a.a.s., such as expansion properties and $2r$-connectedness, are true also of $\mathbb{G}'$. A similar result is obtained for simple graphs with $\mathbb{G}^*$ replaced by $\mathbb{G}$ and $+$ replaced by $\oplus$, the graph sum conditioned on the result being simple.*

Let M be an ergodic Markov chain over Ω with transition matrix P and unique stationary distribution π. If we start in state $x \in \Omega$, the distribution of the random state at time t is denoted P_x^t. We measure the difference between two such distributions π and σ using the *total variation distance*, defined as $\mathrm{d_{TV}}(\mu, \sigma) = (1/2) \sum_{x \in \Omega} |\mu_x - \sigma_x|$. The mixing time of M is

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

The mixing time measures how long a Markov chain must be allowed to run before it is close to its stationary distribution. If it can be bounded above by a polynomial in the size of a state and $\log \varepsilon^{-1}$ then we say that it is *rapidly mixing*.

To bound the mixing time we may set up a multi-commodity flow over the state space Ω that routes $\pi(x)\pi(y)$ units of distinguishable fluid between all pairs $x, y \in \Omega$. If it is possible to do this in such a way that no transition becomes too congested then a polynomial bound results [4, 15, 14, 23]. One method of doing this is to define a set of directed *canonical paths* $\Gamma = \{\gamma_{xy}\}_{x,y\in\Omega}$ and route all the fluid from x to y along γ_{xy}. As in Jerrum [14], define the congestion of set Γ as

$$\rho(\Gamma) = \max_{t=(u,v)} \left\{ \frac{1}{\pi(u)P(u,v)} \sum_{\mathrm{cp}(t)} \pi(x)\pi(y)|\gamma_{xy}| \right\}, \tag{1}$$

where $\mathrm{cp}(t)$ is the set of canonical paths using transition t. Then the mixing time of M can be bounded above by

$$\tau(\varepsilon) \leq 2\rho\,(2\ln \varepsilon^{-1} + \ln \pi_0^{-1}), \tag{2}$$

where $\pi_0 = \min_{x\in\Omega} \pi(x)$.

The problem becomes one of bounding the number of canonical paths using a given edge. Suppose we have an injective *encoding function* $\eta_t : \mathrm{cp}(t) \to \Omega$. Intuitively, this bounds the number of canonical paths using t away from the worst case of $\binom{|\Omega|}{2}$ to just $|\Omega|$. For specifics, see [14].

Fact 1. *One acquires a polynomial bound on the mixing time of a Markov chain if, given the current transition t, an encoding state $\eta_t(x, y)$, and at most a polynomial amount of additional information, one can recover the entire x, y canonical path.*

3 Network definitions

There are two kinds of Cycle network.

Definition 1. *A* Hamiltonian network *is a $2r$-regular graph on n vertices comprising an "outer" Hamilton on consecutive vertices $[n]$ and $r-1$ distinct "inner" random Hamilton cycles, referred to as layers.*

Definition 2. *A* Factor Network *is a $2r$-regular multigraph on n vertices. It is similar to the Hamiltonian Network, but instead of inner Hamilton cycles, it contains $r-1$ distinct inner random 2-factors of the complete graph, which resemble unions of disjoint cycles.*

Self-loops and parallel edges prevent optimal connectivity in a network topology as they represent missed opportunities for a node to communicate with other nodes. We additionally define simple versions of the above that avoid these deficiencies by conditioning on the resulting topology's being simple.

Cycle networks can easily grow or shrink as nodes enter or leave. A node that wishes to join the network must know the identity of at least one node already in that network. Such an intermediary is called the *gateway node*. All nodes may serve as a gateway, perhaps by publishing their IP address and listening on known ports. Joining is accomplished by the *clothespin* operation. The new node samples r edges, one in each layer, by some means (e.g., a random walk). It then interposes itself between the endpoints of these edges. Parting is the complementary operation: the leaving node instructs its neighbours on each layer to connect to one another, ensuring the topology remains consistent when it leaves. Joining causes a parallel edge when any of the pinned edges share a node. This is easy to check for and the joining node need only sample edges until they are independent. Parting causes a parallel edge in a Factor graph when the parting node is on a 3-cycle in some layer. The two remaining nodes on that cycle find themselves with a parallel edge and must rejoin that layer through a partial clothespin.

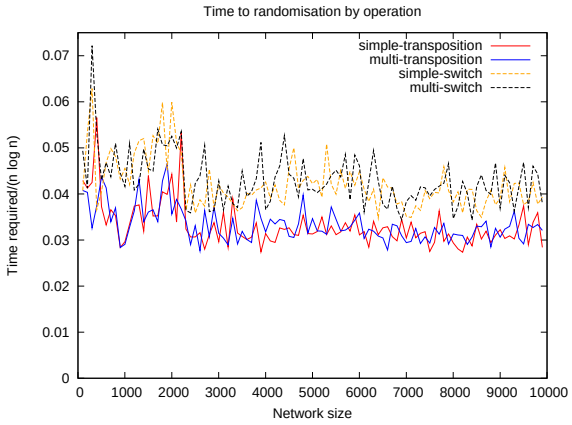

Figure 2: Evidence for a $O(n \log n)$ mixing time. Each trial measures the time required by the various randomising operations to reduce the diameter of $r = 2$ ring lattices of increasing size to within 1.5 times the expected diameter of a random regular graph. This time is divided by $n \log n$, which appears to yield a constant.

3.1 Self-stabilisation

Randomness is core to Cycle networks and it is the constant self-stabilising effect of the randomising operations that allows them to recover from damage accumulated during the natural evolution of a real-life network. The two randomising operations of interest to us, the switch and the transposition, are pictured in Figure 1. Transpositions serve to swap two elements within or between cycles. They preserve Hamilton cycles and so are natural candidates for the Hamiltonian network. Let us call simplicity-preserving transpositions *simple-transpositions* and non-simplicity-preserving, *multi-transpositions*. Switches are adapted from their general graph definitions [2] to preserve the identities of the layers. They tend to destroy and join cycles within a layer, so are best suited to Factor networks. Define *simple-* and *multi-switches* in the obvious fashion.

At a rate proportional to $\log n$, nodes spontaneously initiate the appropriate randomising operation. (For distributed methods of estimating n, see [17].) In case of collisions (with joins, parts, or other randomising operations) we cancel the operation. We do not expect many switch-switch or transposition-transposition collisions due to their infrequency and the fact that they are so small compared with even a modest network.

Sections 4 and 5 give asymptotic results. Here we present empirical evidence that the switch and transposition randomise a graph in around $O(n \log n)$ applications. The number of these applications, divided among the n nodes, gives the above rate. In the case of non-simplicity-preserving operations this agrees with their asymptotic bounds; for simple switches, it is encouraging evidence that the bound acquired is weaker than the true running time.

Figure 2 shows a graph of the observed behaviour for each of the randomising operations on graphs of various sizes. All simulations start with the highly structured ring lattice (each vertex is connected to its r closest neighbours on each side of the outer Hamilton cycle), which has diameter $\Omega(n)$. The flip [20, 3], an operation related to the switch, requires $n^3 \log n$ steps until randomisation, suggesting that the ring lattice is not a trivial starting state. We use diameter as a proxy for "randomness", for which we cannot test directly. This is reasonable as low diameter is one of the properties for which we chose random regular graphs. The simulation halts when the diameter is within a factor of 1.5 of the expected value in $\mathbb{G}(n, 2r)$ [28]. Near to the expected value the measured diameter acts like an unbiased random walk; the multiplicative factor serves to cut down the noise that this would cause.

From the graph a few things are clear. Transpositions seem marginally faster than switches, by a constant. Simple and multi forms of switches and transpositions seem roughly equivalent. Further evidence for this conclusion is the fact that in none of the simple trials did the number of blocked moves rise above ten. This confirms the intuition that, in large networks, the

overwhelming majority of potential moves are not blocked by the simplicity requirement.

3.2 Representation as discordant permutations

The rest of this paper rigorously analyses the randomising operations on Cycle network G. Multigraph Cycle networks may be represented as independent permutations, but the correspondence between simple G and Latin rectangle Π is not so immediate, so we explain it now.

Suppose G is a Factor network. We may ban self-loops by setting $s_0 \in \Pi$. Give the cycles of 2-regular layer G_i an arbitrary orientation. It now corresponds to exactly one permutation—say, $P(G_i)$. If $P(G_i) \in \Pi$ bans directed edge uv then we also wish to ban vu, so a layer additionally puts a second permutation $P'(G_i) = [P(G_i)]^{-1}$ in Π. Hence, simple G corresponds to a $(2r+1) \times n$ Latin rectangle.

A given switch will alter only one layer, called its *active* layer $P(G_i) = \pi$, say. From its point of view the other layers (equivalently, $2r-1$ permutations) are fixed. π remains Π-discordant if we keep it discordant with respect to the $2r-1$ fixed permutations and ban 2-cycles within π, which is equivalent to ensuring discordance with $P'(G_i)$.

In this paper we focus on the case that $r = 2$. This fixes the sole inner later as active. From its point of view Π-discordance is equivalent to ensuring $\pi \notin \{i-1, i, i+1\}$, which is a symmetric form of 3-discordance.

4 Randomising operations on multigraphs

If we are not concerned about avoiding self-loops or parallel edges then it is relatively easy to show that both switches and transpositions require only $\mathrm{O}(n \log n)$ time to randomise a layer. The ease of acquiring mixing time bounds for multigraph operations seems to be due to the existence of a known Hamilton cycle, which gives vertex identifiers topological relevance. It contrasts sharply with the effort required to bound the mixing time of the switch on general random regular graphs, the tightest standing bound on which is $\mathrm{O}\big(n^9 \log n\big)$ [2]. By having an operation choose at random the inner layer upon which it operates this implies the Cycle network is randomised in $\mathrm{O}(rn \log n)$ for constant r.

4.1 Multi-transpositions

A transposition swaps two vertices on or between cycles and hence maintains a Hamilton cycle. It is easily implemented in a peer-to-peer setting, requiring a random walk (of length $\log n$ on an already-randomised network) and then a constant number of edge modifications. As such, it is an attractive choice of randomising operation. The following result due to Diaconis and Shahshahani [5], paraphrased into Markov chain terminology, shows that it mixes in time $n \log n$.

Theorem 2 (Diaconis and Shashahani [5])**.** *Let $\mathcal{M}_T$ be a Hamiltonian-graph-valued Markov chain over random transpositions, with transition probabilities*

$$P(x,y) = \begin{cases} 1/n & \text{if } x = y, \\ 2/n^2 & \text{if } x \text{ and } y \text{ differ by exactly one transposition}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

then $\mathcal{M}_T$ has uniform stationary distribution and mixing time $\tau_{\mathcal{M}_\mathrm{T}}(\varepsilon) \in \mathrm{O}(n \log n)$.

4.2 Multi-switches

A switch breaks edges ab, cd, say, and "switches in" one of ac, bd or ad, bc. One of these breaks cycles (factor switch) and the other reverses a portion of a cycle (Hamiltonian switch). To distinguish between the two we can give each cycle's edges a logical direction. A Hamiltonian switch can randomise a Hamiltonian in $\mathrm{O}(n \log n)$ time [6] but may leave $\Omega(n)$ edge directions inconsistent. These are expensive to update in a peer-to-peer setting, so we focus on the factor switch (hereafter simply "switch"), which does not incur this penalty.

1. Choose $r \in \{0, 1\}$ u.a.r. If $r = 1$ set $X_1 \leftarrow X_0$ and stop.

2. Choose distinct $i, j \in [n]$ u.a.r. If $X_0 \circ (i, j)$ contains a discordance collision, set $X_1 \leftarrow X_0$; else, set $X_1 \leftarrow X_0 \circ (i, j)$.

Figure 3: Trial defining one step of $\mathcal{M}_S$ from state X_0.

Recall that each layer has corresponding permutation $P(G_i) = \pi$ with arbitrary cycle orientation (it is natural to use the logical orientation). A switch on edges ab, cd forms π' by setting $\pi'(a) = \pi(b)$, $\pi'(b) = \pi(a)$, and $\pi'(i) = \pi(i)$ $(i \in [n] \setminus \{a, b\})$. Expressed in this form it is clear that the bound in [5] applies and we obtain that switches require $\mathrm{O}(n \log n)$ time.

5 Randomising operations on simple graphs

This section deals with discordant analogues of the switch and transposition operations, which preserve network simplicity for little extra work at the protocol level. Fix a layer G_i with Latin rectangle Π and let $\pi = P(G_i)$ be a permutation representing the active layer. The other layers remain static, and it is convenient to represent the positions that they block with a *collision function* $I(i) = \{\sigma(i) \mid \sigma \in \Pi \setminus \{P(G_i), P'(G_i)\}\}$.

Many of the proofs in this section are applicable to general Latin rectangles and the application-specific requirement that π avoids 2-cycles would muddy them. We postpone discussion of that additional restriction until Section 5.4. Until then, define $\mathfrak{D}_n = \mathfrak{D}_{n,\Pi}$, the set of all permutations discordant with respect to Π, and let $q = |\Pi| - 1$ be the number of permutations with respect to which π is discordant.

5.1 Discordant switches on permutations

Define the discordant switch Markov chain, $\mathcal{M}_S$, over the state space $\mathfrak{D}_n$ and with discordant switches as transitions, according to Figure 3. A switch $(i, j) \in [n]^2$ is *discordant* in $X \in \mathfrak{D}_n$ when $X(i) \notin I(j)$ and $X(j) \notin I(i)$. If switch (i, j) is discordant then $X \circ (i, j) = X' \in \mathfrak{D}_n$.

Lemma 1. *If $n \geq 5q + 3$ then, for all $X, Y \in \mathfrak{D}_n$, it is possible to transform X into Y with a sequence of no more than $3n$ discordant switches.*

The idea of the lemma is that we avoid blocks on the path from X to Y by moving the elements that we want to switch to some neutral index before switching. If n is large enough then we can always find a suitable index.

So $\mathcal{M}_S$ is connected, and the probability $\frac{1}{2}$ self-loop on each state ensures aperiodicity, so $\mathcal{M}_S$ is ergodic with unique stationary distribution π. Since discordant switches are symmetric, in the sense that $X_1 = X_0 \circ (i, j) \in \mathfrak{D}_n$ iff $X_0 = X_1 \circ (i, j) \in \mathfrak{D}_n$, π is uniform, and so $\mathcal{M}_S$ samples uniformly at random from $\mathfrak{D}_n$.

5.2 A first attempt using canonical paths

When attempting to bound the running time of $\mathcal{M}_S$ it is not clear how to apply light-weight techniques such as coupling when $q > 1$. The blocked moves produce too many bad cases so we proceed with the more flexible canonical path approach. The minority of "bad" paths with many blocked transitions prevents the standard technique from yielding a meaningful bound but in the next section we describe a novel method to avoid these bad paths.

Recall Fact 1 and suppose we are building the X–Z canonical path. We gain a polynomial bound on the mixing time $\tau_{\mathcal{M}_S}(\varepsilon)$ if we are able to construct the entire path knowing only the following: the switch just performed, (X_{R-1}, X_R), from which we recover the indices switched, (i^-, j^-); and an *encoding* state E that tells us how to get closer to Z. We may also guess extra bits in a way that is made precise in Lemma 3, for a relaxation of the bound that is exponential in the number of bits. Our aim now is to present an algorithm, CP, that, given the above information, can specify the next switch.

	$\cdots$	i	$\cdots$	j	$\cdots$	k	$\cdots$
Y	$\cdots$	$Y(i)$	$\cdots$	$Y(j)$	$\cdots$	$Y(k)$	$\cdots$
X_R	$\cdots$	$X_R(i)$	$\cdots$	$Y(i)$	$\cdots$	$X_R(k)$	$\cdots$
X_{R+1}	$\cdots$	$X_R(i)$	$\cdots$	$X_R(k)$	$\cdots$	$Y(i)$	$\cdots$
X_{R+2}	$\cdots$	$Y(i)$	$\cdots$	$X_R(k)$	$\cdots$	$X_R(i)$	$\cdots$

Figure 4: The encoding has specified a switch (i, j) which is blocked on permutation X_R. Index k is chosen to permit the following: a *sideways* switch (j, k), followed by an *unblocked* switch (i, k). Note that k may also be less than j (but not less than i).

First let us define an encoding E for the canonical path from X to Z. Let S be a sequence of *sequential* switches that transform X into Z. The mth sequential switch is of the form (m, j_m) $(0 \leq m < n)$, where m is the *sequent* and the corresponding $j_m = S(m)$ is the *consequent*. Assume for now that all sequential switches in S are discordant. If the mth switch need do nothing then we set $S(m) = m$, a *null* switch. Clearly, S is completely specified by the ordered list of its consequents, which we call its consequent-list. It is not hard to see the following lemma.

Lemma 2. *There exists a bijection between consequent-lists and permutations.*

For our purposes any such bijection h will do, but in the interests of concreteness we set down an intuitive mapping h from $\pi \in \mathfrak{S}_n$ to some consequent-list S. Let $h(\pi(m)) = S(m) = \pi(m) + r_\pi(m)$, where $r_\pi(m) = |\{m^* \in [m] \mid \pi(m^*) > m\}|$ counts the number of times π chose a value greater than m in previous (lower) values of $\pi(m)$. This brings the different but commensurate choices of π and S into accord.

It is likely that S specifies a non-discordant switch. Let $S(i) = (i, j)$ be such a *blocked* switch. Since we cannot perform $S(i)$ directly without leaving $\mathfrak{D}_n$, we will perform one *sideways* switch to break the block, followed by an *unblocked* switch that correctly sets the contents of index i. To be precise, the sideways switch chooses $k \in G_k$, where

$$G_k = \{k \in [n] \mid k > i,\ k \neq j,\ X_R(i), X_R(j) \notin I(k),\ X_R(k) \notin I(j)\}\,.$$

Note that when $i > n - 3q - 1$ there is no guarantee that $G_k \neq \emptyset$. However, when choosing the encoding, and hence the canonical path, we may condition on the final $c_n = 3q - 1$ transitions not being blocked. This bans at most $k^{c_n} \in \mathrm{o}(n)$ paths, but the number of total paths grows with n, so conditioning in this way has no effect asymptotically.

Having selected an appropriate k we perform the switches illustrated in Figure 4. These two switches alter the contents of index k, but if k is chosen deterministically (e.g., the lowest-indexed candidate) then its effects can be allowed for when choosing the encoding. By now we have enough information to choose a consequent-list S, and can set $E = h^{-1}(S)$. It is worth noting that in this proof the encoding is fixed for a given canonical path, which is not necessary, or even generally the case for canonical path proofs.

We now state the canonical path algorithm CP. In order to recognise each sideways switch (j, k) when we see it we remember each j for $\lg n$ bits. Call the set of j's remembered this way C. The previous switch might have been either forwards, sideways, or unblocked. We must recover its sequent. The method for forwards and unblocked switches (case A) differs from that for sideways switches (case B), so we must first determine which of these two cases we are in. If $i^- \notin C$ then the previous switch must have been case A; otherwise, it could be either a sideways step or a normal switch that by chance shares the same index. We guess which is true, for 1 bit of additional information per entry of C. One step of the algorithm is complete when we have determined the next switch to perform. The two cases are as follows.

Case A *The previous switch was forwards or unblocked:* In either case, i^- is the previous sequent, and we correctly obtain the current sequent by setting $i = i^- + 1$. If $i = n$ then we have obtained Z and we halt. Otherwise, from the encoding we recover $j = h(E)(i)$. If $i = j$ then this is a null switch and we repeat case A with $(i^-, j^-) \leftarrow (i, j)$. If $X_R(i) \notin I(j)$ then the next switch is (i, j); otherwise this switch is blocked. In this case we select the lowest index $k \in G_k$ as specified above and perform the sideways switch (j, k). (Recall that we can assume that $G_k \neq \emptyset$ at each point during the construction of the path.)

Case B *The previous switch was sideways:* In this case, (i^-, j^-) correspond to the previous switch's (j, k). We guess which is which for 1 bit. Now, a sideways switch was necessary, meaning that $X_{R-1}(i) \in I(j)$. So there are only $|I(j) \setminus \{j\}| = q - 1$ possible values of $X_R(i)$, and $X_R(i) = X_{R-1}(i)$, so we can guess the identity of i from $q-1$ choices. Now we perform (i, k). By definition of k this cannot be blocked, so we are guaranteed to make progress on this switch.

To analyse the congestion of CP we require two lemmas.

Lemma 3. *Let t be a sequential switch of sequent i; $\mathrm{cp}(t)$, the set of canonical paths using t; and b, be the number of bits guessed in the course of the algorithm. Then $|\mathrm{cp}(t)| \leq \frac{n!}{n-i}\, 2^b$.*

The encoding serves to bound the number of paths running through t; and each bit guessed causes the number of paths through the guessing state to at most double.

Lemma 4. *For constant q, $\mathfrak{D}_n$ tends in size towards $e^{-q}\, n!$.*

As observed before (e.g., [10]), we can obtain a lower bound from Van der Waerden's bound [7, 8] on the permanent of a doubly-stochastic matrix and an upper bound from Minc's inequality [21, 1] on 0,1-matrices with given row sums. The acquired bound is sufficient for us, although improved bounds do exist (see, e.g., the improvement on Minc in [11]).

Recalling the definition of congestion in (1),

$$\rho(\Gamma) \leq \max_t \left\{ |\mathfrak{D}_n|(n-i) \cdot \frac{n + l_\mathrm{o}}{|\mathfrak{D}_n|^2} \cdot |\mathrm{cp}(t)| \right\} \tag{4}$$

$$\leq \max_t \left\{ \frac{2n(n-i)}{|\mathfrak{D}_n|} \cdot \frac{n!}{n-i} \cdot 2^{l_\mathrm{o}(\lg n + \lg(q-1)+2)} \right\} \tag{5}$$

$$\leq \frac{2^{2l_\mathrm{o}+1}(q-1)^{l_\mathrm{o}}}{Q_n} \cdot n^{l_\mathrm{o}+1} \tag{6}$$

where $|\gamma_{xy}|$ is the length of the canonical path between x and y and $Q_n = |\mathfrak{D}_n|/|\mathfrak{S}_n| < 1$. Line 4 uses the following facts: π is uniformly $|\mathfrak{D}_n|^{-1}$; the u, v transition probability in a switch of sequent i is $1/(n-i)$; and the maximum canonical path length is $n + l_\mathrm{o}$. Line 5 uses $l_\mathrm{o} < n$, Lemma 3, and the number of bits guessed in the algorithm.

So the congestion of CP depends on l_o and Q_n. Lemma 4 shows that Q_n tends towards a constant. As for l_o, we currently have no bound other than the trivial one that $l_\mathrm{o} \leq n$, which relaxes the above bound to $\Omega(n^n)$. Because congestion is a multiplicative factor in the mixing time 2, we need a method of bounding l_o, which is the focus of the next section.

5.3 Canonical paths via random states

The problem in our application is that it is not clear how to design a canonical path schema such that there are constant blocked transitions in every $\gamma_{X,Z}$. However, there are expected constant blocked switches on every path, which suggests the following trick that is to our knowledge novel. For each $X, Z \in \mathfrak{D}_n$:

1. Let $A_l(X) \subseteq \mathfrak{D}_n$ be the set of discordant permutations that are *reachable* from discordant permutation X by algorithm CP with at most l sideways switches. Lemma 5 shows that one can find constant l_o, dependent only on q, such that $|A_{l_\mathrm{o}}(X)| \geq (\frac{1}{2} + \delta)|\mathfrak{D}_n|$ for $\delta > 0$.

2. Let $B(X,Z) = A_{l_o}(X) \cap A_{l_o}(Z)$ be the set of states that are reachable from both X and Z with at most l_o sideways switches. By the preceding point, $|B(X,Z)| \geq 2\delta|\mathfrak{D}_n|$. Though the actual set may differ for each X, Z pair, this would only reduce congestion.

3. Choose $Y \in B(X,Z)$ in such a way as to minimise the number of times any given Y is chosen in Γ. The flow from X to Z can be routed along the canonical paths $\gamma_{X,Y}$ and $\gamma_{Y,Z}$, effectively joining them at Y. Call these new canonical paths Γ'. In the worst case the new congestion is given by $\rho(\Gamma') \leq 2\rho(\Gamma) \cdot 1/2\delta$, which is equivalent to $\rho(\Gamma)$ up to a constant.

Lemma 5. *Let l_o be any integer that satisfies $(eq/l_o)^{l_o} < \frac{1}{2}$. If Then the set of reachable discordant permutations $A(X)$ has size $(\frac{1}{2} + \delta)|\mathfrak{D}_n|$ $(\delta > 0)$. In particular, if $r = 2$, $l_o = 8$ and $\delta \geq 0.882$.*

(This can be shown by a proof similar to that of Chernoff bounds.)

Inequality (2) converts congestion into a bound on mixing time and yields the following theorem.

Theorem 3. *The Markov chain $\mathcal{M}_S$ over 3-discordant permutations using discordant switches mixes in time $\tau_{\mathcal{M}_S}(\varepsilon) \in \mathrm{O}\big(n^{10} \ln n\, \varepsilon^{-1}\big)$.*

5.4 Simple-switches

We now discuss how to extend the above argument to *strongly-discordant* permutations, which additionally ban 2-cycles. Let $\mathfrak{D}_n^*$ be the set of strongly-discordant permutations. A *strongly discordant switch* (i,j) in X is a discordant switch subject to the *2-cycle restriction*, that is, that $X(X(i)) \neq j \wedge X(X(j)) \neq i$. This additional restriction prevents i and j from being at distance 2 on a cycle in X, which is the only way of creating a 2-cycle with a switch when $X \in \mathfrak{D}_n^*$.

Remark 1. *Generalised c-cycle-restrictions suggest a means of banning cycles of any constant length.*

Suppose the encoding instructs us to move $X_R(j)$ $(= m)$ into i. Switch (i,j) is now subject to the *2-cycle-restriction* that $X_R(X_R(i)) \neq j$ and $X_R(X_R(j)) \neq i$. It happens that a modified sideways switch will deal with the former inequality. For the latter we define a *cyclophobic* switch that finds some neutral index $l \in G_l^*$ and switches it with $X_R(j)$, thus removing the $m\, X_R(m)$ edge that was the problem. For (i,j) on X_R with $X_R(X_R(j)) = i$ we define

$$G_l^* = \{l \in [n] \mid l > i, l \neq j,\, X_R(l) \notin I(m),\, i \notin I(l),\, X_R(X_R(m)) \neq l,\, X_R(X_R(l)) \neq m\}\,.$$

There is the concern that altering the contents of position m might constitute a step backwards. However, if m were in its final state then this would imply that Y contains a 2-cycle, which we disallow in Lemma 7.

We now modify the sideways switch to avoid 2-cycles. Choose k from

$$G_k^* = \{k \in G_k \mid X_R(X_R(j)) \neq k, X_R(X_R(k)) \neq j,\, X_R(k) \neq k, X_R(X_R(i)) \neq k, X_R(X_R(j)) \neq i\}\,,$$

which is simply G_k with added 2-cycle-restrictions on (j,k) in X_R and (i,k) in X_{R+1}. The final three inequalities derive from 2-cycle-restrictions on the latter. Note that the final inequality does not depend on k. If $X_R(X_R(j)) = i$ then there is no way to perform a sideways switch, so we must perform cyclophobic switches before sideways switches.

The changes to algorithm CP are as follows. First, in addition to the list of sideways switches C, we have a list of cyclophobic switches D. As before, if $i^- \notin C \cup D$ then it is direct or unblocked (case A); if in C we must guess between case A and case B; and if in D we must guess between case A and case C. These latter guesses cost 1 bit per entry of C and D. Let us consider the cases.

Modified Case A *The previous switch was a normal or unblocked switch:* Recover i and j as before. If $X_R(X_R(j)) = i$ then we must perform a cyclophobic switch (m, l) where $l \in G_l^*$ as defined above. Otherwise, test for $X_R(X_R(i)) = j$ and the usual I-restriction test. If either fails, a sideways switch (j, k) $(k \in G_k^*)$ will avoid the block. If no test fails then we know that the switch (i, j) is strongly discordant and are free to perform it.

Modified Case B *The previous switch was sideways:* This is largely unchanged but we choose i from q choices, as there is the additional possibility that the sideways switch was caused because $X_{R-1}(X_{R-1}(i)) = j$.

New Case C *The previous switch was cyclophobic:* (i^-, j^-) correspond to (m, l), and we guess which is which for 1 bit. Since the previous switch was cyclophobic we know that $i = X_{R-1}(l)$ and $j = X_R^{-1}(m)$ There is still the possibility that $X_R(X_R(i)) = j$ or that the I-restriction fails: This is handled exactly as in case A.

Let us analyse the congestion of these new canonical paths, Γ^*, by contrast with Γ. Lemma 6 bounds the size of $\mathfrak{D}_n^*$ and Lemma 7 gives a new, weaker bound for l_o. However, we cannot distinguish between blocks caused by 2-cycle restrictions and blocks caused by I restrictions, so must assume that both occur. By analogy to (5),

$$\rho(\Gamma^*) \leq \max_t \left\{ \frac{2 \cdot 2n(n-i)}{|\mathfrak{D}_n|} \cdot \frac{n!}{n-i} \cdot 2^{l_o(\lg n + \lg q + 2) + l_o(\lg n + 2)} \right\}.$$

Lemma 6. *For constant q, $\mathfrak{D}_n^*$ tends in size towards $\frac{1}{2}e^{-q}\, n!$.*

One may choose a 2-cycle then apply Minc to count the discordant configurations given this. It is routine to show that the effect of fixing a 2-cycle is $o(n)$.

Lemma 7. *Let l_o be any integer that satisfies $(e(q+1)/l_o)^{l_o} < \frac{3}{4}$. Then the set of reachable discordant permutations $A(X)$ has size $(\frac{1}{2} + \delta)|\mathfrak{D}_n|$ $(\delta > 0)$. In particular, if $r = 2$, $l_o = 12$ and $\delta \geq 0.152$*

This follows because there is one additional position that we must avoid switching with for any given sequent, and $\mathfrak{D}_n^*$ is smaller than $\mathfrak{D}_n$ by a factor of 2.

All of the above serves to establish the following and final theorem.

Theorem 4. *Let $\mathcal{M}_S^*$ be a Markov chain over Factor networks for $r = 2$ using strongly-discordant switches as defined above. Then $\tau_{\mathcal{M}_S^*}(\varepsilon) \in \mathrm{O}\big(n^{25} \log n\, \varepsilon^{-1}\big)$.*

6 Concluding remarks

We have described a family of networks and examined randomising operations on them. In the course of analysing simple-switches we have developed a robust mixing time analysis technique that can handle blocked moves when the expected number of them on a random path is constant. Simple-transpositions have not been considered here but there are signs that they are easier to analyse than simple-switches—for instance, the absence of self-loops or parallel edges within a layer. The canonical path via random state technique should extend naturally to these.

By randomly choosing the inner layer on which they operate, it is easy to see that the multi-switch and multi-transposition randomise Cycle networks with any constant r. We have shown that this holds for simple-switches when $r = 2$. It seems a natural conjecture that the same is true for constant r.

Acknowledgements

This work was funded by the EPSRC AMORPH project `#EP/D00232X/1`.

References

[1] L. M. Brègman. Some peroperties of nonnegative matrices and their permanents. *Soviet Math. Dokl.*, 14(4):945–949, 1973.

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

[3] C. Cooper, M. Dyer, and A. J. Handley. The flip markov chain and a peer-to-peer network. To appear.

[4] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Annals of Applied Probability*, 1:36–61, 1991.

[5] Persi Diaconis and Mehrdad Shahshahani. Generating a random permutation with random transpositions. *Probability Theory and Related Fields*, 57(2):159–179, June 1981.

[6] Rick Durrett. Shuffling chromosomes. *Journal of Theoretical Probability*, 16(3):725–750, July 2003.

[7] G. Egorychev. The solution of van der waerden's problem for permanents. *Advances in Mathematics*, 42(3):299–305, December 1981.

[8] D. I. Falikman. Proof of the van der waerden conjecture on the permanent of a doubly stochastic matrix. *Mat. Zametki*, 29(6):931–938, 957, 1981.

[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] C. Godsil and B. McKay. Asymptotic enumeration of latin rectangles. *Journal of Combinatorial Theory, Series B*, 48(1):19–44, February 1990.

[11] Mark Huber and Jenny Law. Fast approximation of the permanent for very dense problems. In *SODA '08*, pages 681–689, Philadelphia, PA, USA, 2008.

[12] Mark T. Jacobson and Peter Matthews. Generating uniformly distributed random latin squares. *Journal of Combinatorial Designs*, 4(6):405–437, 1996.

[13] S. Janson, T. Łuczak, and A. Ruciński. *Random Graphs*. Wiley, 2000.

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

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

[16] I. Kaplansky and J. Riordan. The problème des ménages. *Scripta Math*, 12:113–124, 1946.

[17] C. Law and K. Y. Siu. Distributed construction of random expander networks. *INFOCOM 2003*, 3, 30 March-3 April 2003.

[18] Xiaozhou Li, Jayadev Misra, and C. Greg Plaxton. Maintaining the ranch topology. *Journal of Parallel and Distributed Computing*, in press, 2010.

[19] B. D. MacKay and N. C. Wormald. Uniform generation of random Latin rectangles. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 9:179–186, 1991.

[20] P. Mahlmann and C. Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *SPAA '05*, pages 155–164, New York, NY, USA, 2005. ACM.

[21] H. Minc. Upper bounds for permanents of $(0,1)$-matrices. *Bulletins of the American Mathematical Society*, 69:789–791, 1963.

[22] Joseph Santmyer. Five discordant permutations. *Graphs and Combinatorics*, 9(2):279–292, June 1993.

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

[24] I. Stoica, R. Morris, D. Karger, F. M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01*, volume 31, pages 149–160, New York, NY, USA, October 2001. ACM Press.

[25] J. Touchard. Sur un problème de permutations. *Comptes Rendus de l'Academie des Sciences Paris*, 198:631–633, 1934.

[26] L. G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11:350–361, 1982.

[27] E. G. Whitehead. Four discordant permutations. *Journal of the Australian Mathematical Society (Series A)*, 28:369–377, 1979.

[28] N. Wormald and V. Sanwalani. The diameter of random regular graphs. Incomplete paper.

A Omitted proofs

Proof of Lemma 4. As frequently observed before (e.g., [10]), we can obtain a lower bound from Van der Waerden's bound on the permanent of a doubly-stochastic matrix; and an upper bound follows from Minc's inequality on 0,1-matrices with given row sums.

Let A be an $n \times n$ matrix whose rows correspond to indices of a permutation π and columns correspond to where π maps a given index. We build A by placing a 0 in places forbidden by the discordance property, and a 1 in all other positions. All row and column sums are all equal to $n - q$, as there are q forbidden places in each row and column. In this schema, the number of π's that do not violate the discordance property is exactly the permanent of A, $\operatorname{per} A$.

Let $A^{\circ} = A/(n-q)$. The matrix A° is double-stochastic, so Van der Waerden's conjecture (which was proven by independently by Egorychev and by Falikman [7, 8]) yields

$$\begin{aligned} \operatorname{per} A &= \operatorname{per} A^{\circ}(n-q)^n && (7)\\ &\geq n!\, n^{-n}(n-q)^n && (8)\\ &= n!\, \left(1 - \frac{q}{n}\right)^n \\ &\sim n!\, e^{-q} \end{aligned}$$

for fixed q, where (7) is due to the linearity of permanents and (8) is Van der Waerden's conjecture.

An upper bound follows from a conjecture of Minc's [21] which was proven by Brègman [1].

	$\cdots$	i	$\cdots$	j	$\cdots$	k	$\cdots$
X	$\cdots$	$X(i)$	$\cdots$	$X(j)$	$\cdots$	$X(k)$	$\cdots$
Y	$\cdots$	$Y(i)$	$\cdots$	$X(i)$	$\cdots$	$Y(k)$	$\cdots$
X'	$\cdots$	$X(k)$	$\cdots$	$X(j)$	$\cdots$	$X(i)$	$\cdots$
Y'	$\cdots$	$Y(k)$	$\cdots$	$X(i)$	$\cdots$	$Y(i)$	$\cdots$
X''	$\cdots$	$X(k)$	$\cdots$	$X(i)$	$\cdots$	$X(j)$	$\cdots$

Figure 5: Three switches to bring X and Y closer together. With suitably neutral k, two (i,k) switches in X and Y avoid potential blocks, then a final switch reduces the distance between the two permutations.

It gives us

$$\begin{aligned}
\operatorname{per} A &\le \prod_{i=1}^{n} (r_i!)^{1/r_i} \qquad (9)\\
&\le ((n-q)!)^{\frac{n}{n-q}}\\
&\sim \left[\sqrt{2\pi n}\, e^{-(n-q)} (n-q)^{n-q}\right]^{\frac{n}{n-q}}\\
&\sim \sqrt{2\pi n}\, e^{n} (n-q)^{n}\\
&\sim \sqrt{2\pi n}\; e^{n} n^{n} \left(1 - \frac{q}{n}\right)^{n}\\
&\sim n!\, e^{-q}
\end{aligned}$$

where (9) is Minc's conjecture.

□

Proof. Proof of Lemma 1

Suppose $\mathbb{H}(X,Y) > 0$, and let $i,j \in S(X,Y)$ be such that $Y(j) = X(i)$, as pictured in Figure 5. Let

$$G \;=\; \{k : k \neq i,j,\; X(k), Y(k) \notin I(i),\; X(i), Y(i), X(j) \notin I(k)\}\,.$$

Suppose $G \neq \emptyset$, and choose any $k \in G$. Choice of k implies that $X(i)$ can be swapped with $X(k)$ and $Y(i)$ can be switched with $Y(k)$, so (i,k) is discordant in both X and Y. Perform these switches together, giving X' and Y' in Figure 5. Clearly, $\mathbb{H}(X',Y') = \mathbb{H}(X,Y)$. Now perform the switch (j,k) in X'. This is discordant because $X'(j) = X(j) \notin I(k)$. Note that we must have $X'(k) \notin I(j)$, since $X'(k) = Y'(j) = X(i)$. Then we have the final line in Figure 5.

Now $\mathbb{H}(X'',Y') = \mathbb{H}(X',Y') - 1 = \mathbb{H}(X,Y) - 1$. We have performed three discordant switches, two in X and one in Y. Thus, since $\mathbb{H}(X,Y) \le n$, we can achieve $\mathbb{H}(X^{(2n)}, Y^{(n)}) = 0$, so $\mathrm{d}(x^{(2n)}, y^{(n)}) = 0$. Hence $\mathrm{d}(X,Y) \le 3n$, unless we ever encounter the condition $G = \emptyset$.

So let us estimate the size of G. Clearly $i,j \notin G$ removes 2 possible positions from G. Each of the five conditions $X(k), Y(k) \notin I(i)$ and $X(i), Y(i), X(j) \notin I(k)$ can remove at most $\max_i |I(i)| \le q = 2r-1$ possible positions from G. Thus $|G| \ge n-(2+5q)$. Hence, if $n \ge 5q+3$, we always have $G \neq \emptyset$.

□

Proof. Proof of Lemma 2

The consequent of sequential switch m can range from m to $n-1$, so there are $n-m$ ways to choose it. This choice of consequent is independent for each of the n switches in a consequent-list, so the number of consequent-lists is $n! = n(n-1)\cdots 1$.

□

Proof. Proof of Lemma 3

For the first part of the proof, assume $b = 0$. Fix some switch t. The sequent provides a natural ordering on the switches in canonical paths in $\mathrm{cp}(t)$, and we say that switches having lower sequent than t are *before* t, and those having higher sequent are *after* t. Let A_t be the set of states from which all paths in $\mathrm{cp}(t)$ start, and B_t, the set of states at which they end. Since there is exactly one canonical path between two states, $|\mathrm{cp}(t)| \leq |A||B|$.

First we build B_t. Let t_i be a switch with sequent i. We can choose the next switch t_{i+1} in $n - i - 1$ ways, as we only need to choose its consequent. Iterating this approach tells us that $|B_t| \leq (n - i - 1)!$. A similar argument yields $|A_t| \leq n!/(n - i)!$. Multiplying these together we find that, when $b = 0$, $|\mathrm{cp}(t)| \leq n!/(n - i)$.

Now to generalise to any b. Let S_l be the bound when $b = l$. We have already determined S_0. Assume that we have a bound S_{l-1} and wish to find some S_l. Suppose we are in switch t when we guess an additional bit. That bit may change the set of paths reached from t depending on its value so that we will choose to go to B_t if the bit is zero and B'_t if it is one. In the worst case these sets are distinct and equally large, so we must double our previous bound to cover the additional possible paths. Hence, $S_l = 2S_{l-1}$. This yields the stated bound.

□

Proof. Proof of Lemma 5

Consider an arbitrary set of switches from $X \in \mathfrak{D}_n$. Let $X_i \in \{0, 1\}$ be the indicator random variable for the event that the ith switch is blocked, so $N = \sum_{i=1}^{n} X_i$ is the number of blocked switches. Let $z \geq 0$. Then we have

$$(1 + z)^{X_i} \;=\; 1 + X_i z,$$

so

$$(1 + z)^N \;=\; \prod_{i=1}^{n} (1 + z)^{X_i} \;=\; \prod_{i=1}^{n} (1 + X_i z) \;=\; \sum_{j=0}^{n} p_j(\mathbf{X}) z^j,$$

where

$$p_j(\mathbf{X}) \;=\; \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq n} X_{i_1} X_{i_2} \cdots X_{i_j}.$$

Now, in the random permutation model,

$$\mathbb{E}[p_j(\mathbf{X})] \;\leq\; \binom{n}{j} \frac{q^j}{n(n-1)\cdots(n-j+1)} \;=\; \frac{q^j}{j!},$$

so

$$\mathbb{E}\left[(1 + z)^N\right] \;\leq\; \sum_{j=0}^{n} \frac{q^j}{j!} z^j \;<\; \sum_{j=0}^{\infty} \frac{(qz)^j}{j!} \;=\; e^{qz}.$$

Now, using Markov's inequality,

$$\Pr(N \geq l_\mathrm{o}) \;=\; \Pr\left((1 + z)^N \geq (1 + z)^{l_\mathrm{o}}\right) \;\leq\; \frac{\mathbb{E}[(1 + z)^N]}{(1 + z)^{l_\mathrm{o}}} \;\leq\; \frac{e^{qz}}{(1 + z)^{l_\mathrm{o}}} \;=\; f(z),$$

say, which is minimised when $z = (l_\mathrm{o} - q)/q = z_0$, say. Thus

$$\Pr(N \geq l_\mathrm{o}) \;\leq\; f(z_0) \;=\; \frac{e^{l_\mathrm{o} - q}}{(l_\mathrm{o}/q)^{l_\mathrm{o}}} \;=\; e^{-q}(eq/l_\mathrm{o})^{l_\mathrm{o}}.$$

We wish to find l_o such that, in the random permutation model,

$$\Pr(N \geq l_\mathrm{o}) \;<\; e^{-q}/2,$$

since we must scale up the probability by $Q_n^{-1} = e^q$ for the discordant permutation model, and we require the probability to be less than $\frac{1}{2}$ in that model.

Let $s(q, l_o) = (eq/l_o)^{l_o}$, so we need $s(q, l_o) < \frac{1}{2}$, i.e. $q < l_o 2^{-1/l_o}/e$. For each q, we will have at most l_o discordances with probability at least $t(q, l_o) = 1 - s(q, l_o + 1) > \frac{1}{2}$ if

q	l_o	$t(q, l_o)$	q	l_o	$t(q, l_o)$
2	6	0.8296	6	16	0.5058
3	8	0.5883	7	19	0.6308
4	11	0.6937	8	22	0.7245
5	14	0.7722	9	25	0.7946

.

A convenient lower bound obtained by elementary algebra is that l_o may be any integer that satisfies

$$\left(\frac{eq}{l_o}\right)^{l_o} < \frac{1}{2}.$$

□

Proof. Proof of Lemma 6

An upper bound on $|\mathfrak{D}_n^*|$ follows from the fact that $\mathfrak{D}_n^* \subset \mathfrak{D}_n$. We require a lower bound. Our approach is to remove the discordant permutations with 2-cycles, S, from the set of discordant cycles.

Let us analyse $\pi \in S$. There are at most $\binom{n}{2}$ ways of choosing the indices of a 2-cycle i, j in π. The remaining entries must be discordant. We may build a $(n-2) \times (n-2)$ 0, 1-matrix M with 1 in entry a, b if $a \notin I(b)$ and 0 otherwise, and then deleting rows and columns i and j. The resulting M is symmetric. Note that the majority of the row sums r_i are $n - q - 2$: at most 4 may differ, and be at most $n - q$. Using Minc again, we have that

$$\begin{aligned}
\operatorname{per} M \leq & \prod_{i=1}^{n} (r_i!)^{\frac{1}{r_i}} \\
< & ((n-q-2)!)^{\frac{n-4}{n-q-2}} \cdot ((n-q)!)^{\frac{4}{n-q}} \\
= & L \cdot R \\
= & |\mathfrak{D}_{n-2}|(1 + \mathrm{o}(n))
\end{aligned}$$

where

$$\begin{aligned}
L \sim & \left(\sqrt{2\pi(n-q-2)}\, e^{-(n-q-2)} (n-q-2)^{n-q-2}\right)^{\frac{n-4}{n-q-2}} \\
\sim & \sqrt{2\pi(n-2)}\, e^{-(n-2)} (n-q-2)^{n-4} \\
\sim & \sqrt{2\pi(n-2)}\, e^{-(n-2)} n^{n-2} \left(1 - \frac{q}{n-2}\right)^{n-2} (n-q-2)^{-2} \\
= & (n-2)!\, e^{-q}\, (1 + \mathrm{o}(n))
\end{aligned}$$

and

$$\begin{aligned}
R \sim & \left(\sqrt{2\pi(n-q)}\, e^{-(n-q)} n^{n-q}\right)^{\frac{4}{n-q}} \\
\sim & e^{-4} n^4.
\end{aligned}$$

$$\begin{aligned}|\mathfrak{D}_n^*| =& |\mathfrak{D}_n| - S \\ <& |\mathfrak{D}_n| - \binom{n}{2}|\mathfrak{D}_{n-2}| \\ \sim& e^{-q}n! - \frac{1}{2}e^{-q}(n-2)! \\ =& \frac{1}{2}e^{-q}n!.\end{aligned}$$

□

Proof. Proof of Lemma 7

There are two changes to make. First, avoiding 2-cycles adds one to the number of positions we must not switch with, which relaxes our bound on l_{o}. Second, we must beat probability $(3/4)e^{-q}$ to accommodate the fact that there are fewer strongly discordant permutations (Lemma 6). In the terminology of Lemma 5, $t(q, l_{\mathrm{o}}) = 1 - s(q, l_{\mathrm{o}} + 1) > \frac{3}{4}$.

□