



UNIVERSITY OF LEEDS

This is an author produced version of *On the switch Markov chain for perfect matchings*.

White Rose Research Online URL for this paper:

<http://eprints.whiterose.ac.uk/111478/>

Article:

Dyer, M, Jerrum, M and Muller, H (2017) On the switch Markov chain for perfect matchings. *Journal of the ACM*, 64 (2). 12. ISSN 0004-5411

<https://doi.org/10.1145/2822322>

© 2017, ACM. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *Journal of the ACM*, VOL 64, ISS 2, June 2017, <https://doi.org/10.1145/2822322>.

On the switch Markov chain for perfect matchings*

Martin Dyer[†]

Mark Jerrum[‡]

Haiko Müller[†]

January 26, 2017

Abstract

We study a simple Markov chain, the switch chain, on the set of all perfect matchings in a bipartite graph. This Markov chain was proposed by Diaconis, Graham and Holmes as a possible approach to a sampling problem arising in Statistics. We ask: for which hereditary classes of graphs is the Markov chain ergodic and for which is it rapidly mixing? We provide a precise answer to the ergodicity question and close bounds on the mixing question. We show for the first time that the mixing time of the switch chain is polynomial in the case of monotone graphs, a class that includes examples of interest in the statistical setting.

1 Introduction

Counting perfect matchings in a bipartite graph is an important computational problem: the permanent of a 0-1 matrix. This, and the computationally equivalent problem of sampling matchings uniformly at random, has applications in Statistics, Statistical Physics and other areas. Diaconis, Graham and Holmes [9] considered applications of the 0-1 permanent to problems in Statistics, in particular where the 0-1 matrix has recognisable structure, which they called *truncated* or *interval-restricted*.

The truncated 0-1 matrices are those for which the columns can be permuted to give the *consecutive 1's* property on rows. That is, no two 1's in any row are separated by one or more 0's. Diaconis, Graham and Holmes [9] considered “one-sided” truncation, where the consecutive 1's appear at the left of each row, and “two-sided” truncation, where the consecutive 1s can appear anywhere in a row. For two-sided truncation, they considered two special cases. In the first, both the rows and columns can be permuted so that they have the consecutive 1's property. In the second, the rows and columns can be permuted so that the consecutive 1's have a “staircase” presentation. This is the *monotone* case, which is of particular interest in certain statistical applications [11].

Diaconis, Graham and Holmes [9] proposed a Markov chain for sampling perfect matchings in a bipartite graph, which we call the *switch chain*. They showed ergodicity of the chain for the truncated matrices considered in [9], and conjectured that it would converge rapidly. Computing the 0-1 permanent is a well-solved problem from a theoretical viewpoint. It is $\#P$ -complete to compute exactly [36], but there is a polynomial time approximation algorithm [19]. However, the switch chain gives a simpler and more practical algorithm than that of [19], making it worthy of consideration. Hence Diaconis, Graham and Holmes's conjecture was subsequently studied in the PhD theses of Matthews [26] and Blumberg [4]. We will discuss their results below.

A 0-1 matrix is equivalently the biadjacency matrix of a bipartite graph, and we will study the graphs corresponding to the matrices considered by Diaconis, Graham and Holmes [9]. We show that their questions are, in fact, questions about counting perfect matchings in *hereditary graph classes*. Hereditary classes are

*The work described here was partially supported by EPSRC grants “Computational Counting” (refs. EP/I011935/1 and EP/I012087/1) and “Randomized algorithms for computer networks” (ref. EP/M004953/1)

[†]School of Computing, University of Leeds, Leeds, LS2 9JT, UK.

[‡]School of Mathematical Sciences, Queen Mary University of London, Mile End Road, London, E1 4NS, UK.

of central interest in modern graph theory. In fact, this paper can be viewed as a first attempt to bring together two previously distinct streams of research in algorithms: structural graph theory and the complexity of counting and sampling.

We identify the largest hereditary graph class in which the switch chain is ergodic: *chordal bipartite* graphs. The classes considered in [9] form an ascending sequence within this class. We examine the mixing time behaviour of the switch chain for graphs from these classes, extending the work of [9], [26] and [4]. We also discuss the difficulty of computing the permanent *exactly* in these classes, where Valiant's result [36] does not necessarily imply #P-completeness.

In particular, we show for the first time that the mixing time of the switch chain is polynomial for monotone graphs. This is proved by a novel application of a simple combinatorial lemma: the solution to the so-called *mountain climbing problem* [15, 21, 28, 34, 37]. Though this lemma is well known, there appears to be no worst-case analysis of this problem in the literature. Therefore we provide such an analysis as an appendix.

After this paper was written, we learned that Bhatnagar, Randall, Vazirani and Vigoda [3] had used a similar approach for a rather different problem. They had analysed the Jerrum-Sinclair Markov chain [18] for generating random bichromatic matchings in a graph that has its edges partitioned into two colour classes.

For the necessary background information on Markov chains, see *inter alia* [1, 17, 23]. For the relevant graph-theoretic background, see [6, 37, 32, 14].

1.1 Notation and definitions

Let $\mathbb{N} = \{1, 2, \dots\}$ denote the natural numbers. If $n \in \mathbb{N}$, let $[n] = \{1, 2, \dots, n\}$ and, if $n_1, n_2 \in \mathbb{N}$, let $[n_1, n_2] = \{n_1, n_1 + 1, \dots, n_2\}$.

We will also use the notation $[n]' = \{1', 2', \dots, n'\}$ and $[n_1, n_2]' = \{n_1', (n_1 + 1)', \dots, n_2'\}$. Here the prime serves only to distinguish i from i' . Ordering and arithmetic for $[n]'$ elements follows that for $[n]$. Thus, for example, $1' < 2'$ and $1' + 2' = 3'$.

A graph $G = (V, E)$ is *bipartite* if its vertex set $V = [m] \cup [n]'$ and there is no (undirected) edge $(v, w) \in E$ such that $v, w \in [m]$ or $v, w \in [n]'$. Thus V comprises two independent sets $[m]$ and $[n]'$. Bipartite graphs $G_1 = ([m] \cup [n]', E_1)$ and $G_2 = ([m] \cup [n]', E_2)$ are *isomorphic* if there are permutations σ of $[m]$ and τ of $[n]'$ such that $(j, k') \in E_1$ if and only if $(\sigma_j, \tau_{k'}) \in E_2$.

Let $G = ([m] \cup [n]', E)$ be a bipartite graph. We consider $[m]$ and $[n]'$ to have the usual linear ordering, and we will abuse notation by denoting these ordered sets simply by $[m]$ and $[n]'$. Then let $A(G)$ denote the $m \times n$ *biadjacency matrix* of G , with rows indexed by $[m]$ and columns by $[n]'$, such that $A(i, j') = 1$ if $(i, j') \in E$, and $A(i, j') = 0$ otherwise. We will use the graph and matrix terminology interchangeably. For example, we refer to rows and columns of G , or edges in $A(G)$.

The neighbourhood in G of a vertex $v \in [m] \cup [n]'$ will be denoted by $\mathcal{N}(v)$. To avoid trivialities, we will assume that G has no isolated vertices, unless explicitly stated otherwise.

A *matching* in a bipartite graph $G = ([m] \cup [n]', E)$ is a set of independent edges, that is, no two edges in the set share an endpoint. A *perfect* matching is a set of edges such that every vertex of G lies in exactly one edge. For a bipartite graph $G = ([m] \cup [n]', E)$ this requires $m = n$, and n independent edges in E . In particular, G can have no isolated vertices. We will call a bipartite graph with $m = n$ *balanced*.

Equivalently, a perfect matching may be viewed as n independent 1's in the $n \times n$ 0-1 matrix $A(G)$. Thus a perfect matching M has edge set $\{(i, \pi'_i) : i \in [n]\}$, where π is a permutation of $[n]$. Equivalently, M has edge set $\{(\sigma'_j, j') : j \in [n]\}$, where σ is a permutation of $[n]$. Note that $\sigma = \pi^{-1}$ as elements of the symmetric group S_n . We may identify the matching M with the permutations π and σ . An example is shown in Fig. 1 below.

The total number of perfect matchings in a bipartite graph G is the *permanent*, which we denote by $\text{per}(A)$ when $A = A(G)$.

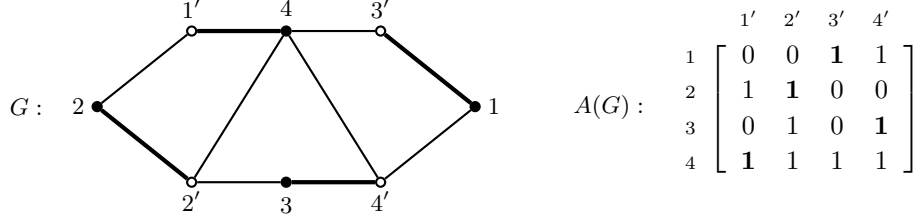


Figure 1: Bipartite graph with perfect matching $\sigma = (3241)$, $\pi = (4213)$.

1.2 Computing the permanent

The permanent has been studied extensively in Combinatorics and Computer Science. Valiant showed that computing the permanent *exactly* is #P-complete for a general 0-1 matrix [36]. No algorithm running in sub-exponential time is known for the exact evaluation of the permanent of 0-1 matrices.

Jerrum, Sinclair and Vigoda [19] showed that the 0-1 permanent has a *fully polynomial randomised approximation scheme* (FPRAS), using an algorithm for randomly sampling perfect matchings. This improved a Markov chain algorithm of Jerrum and Sinclair [18], which was not guaranteed to have polynomial time convergence for all bipartite graphs. The algorithm of [19] is simple, but involves polynomially many repetitions of a polynomial-length sequence of related Markov chains. The best bound known for the running time of this algorithm is $O(n^7 \log^4 n)$, due to Bezáková, Štefankovič, Vazirani and Vigoda [2].

Jerrum, Valiant and Vazirani [20] showed that sampling almost uniformly at random and approximate counting have equivalent computational complexity for many combinatorial problems, including the permanent. Technically, the problem must be *self-reducible*.

From the viewpoint of theoretical Computer Science, these results entirely settle the question of sampling and counting perfect matchings in bipartite graphs. However, simpler methods have been proposed for special cases of this problem, and here we consider one such proposal.

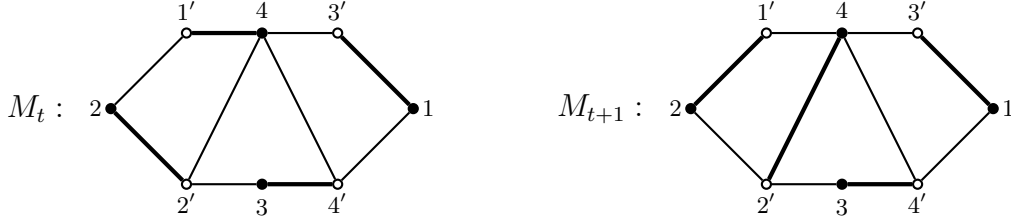
1.3 The switch chain

Diaconis, Graham and Holmes [9] proposed the following Markov chain for sampling perfect matchings from a balanced bipartite graph $G = ([n] \cup [n]', E)$ almost uniformly at random, which we will call the *switch chain*. A transition of the chain will be called a *switch*. Diaconis, Graham and Holmes [9] called this a “transposition”. The switch chain generalises the transposition chain for generating random permutations.

Switch chain

0. Let the perfect matching M_t at time t be the permutation π of $[n]$.
1. Set $t \leftarrow 0$, and let M_0 be any perfect matching of G .
2. Choose $i, j \in [n]$, uniformly at random, so $(i, \pi'_i), (j, \pi'_j) \in M_t$.
3. If $i \neq j$ and $(i, \pi'_j), (j, \pi'_i)$ are both in E ,
set $M_{t+1} \leftarrow M_t \setminus \{(i, \pi'_i), (j, \pi'_j)\} \cup \{(i, \pi'_j), (j, \pi'_i)\}$.
4. Otherwise, set $M_{t+1} \leftarrow M_t$.
5. Set $t \leftarrow t + 1$. If $t < t_{\max}$, repeat from step (2). Otherwise, stop.

Note that the switch chain is invariant under isomorphisms of G , so properties of the chain can be investigated from the viewpoint of graph theory. An example of a switch is shown below, with the edges $(4, 1')$, $(2, 2')$ being switched for $(4, 2')$, $(2, 1')$.



The remainder of the paper is arranged as follows. Section 2 introduces the hereditary graph classes relevant for our study. We determine the largest such class for which the switch chain given above is ergodic, that is, converges eventually to a stationary distribution. This is the class of chordal bipartite graphs. Unfortunately, the mixing time of the switch chain on chordal bipartite graph, though defined, is exponential. We then examine increasingly restricted graph classes, and determine that the mixing time remains exponential. Finally, we arrive at the classes of monotone graphs and chain graphs. For chain graphs, an explicit formula is known for the number of perfect matchings. Section 3 contains the main result of the paper, namely that the mixing time of the switch chain on monotone graphs is polynomial. Section 4 briefly considers possible directions for further work. The appendix contains a refined analysis of the “mountain climbing problem”, the solution of which provides a key element in the proof of polynomial-time mixing for monotone graphs.

2 Graph classes

2.1 Chordal bipartite graphs

The first question we might ask about the switch chain is: for which class of graphs is it ergodic? We wish to have a graph-theoretic answer to this question, so that we can recognise membership of graphs in the class. Therefore, we restrict attention to *hereditary* graph classes, that is, those for which all (vertex) induced subgraphs of every graph in the class are also in the class. Hereditary classes have become central to graph theory, and are most usually characterised by describing a minimal set of “excluded subgraphs”, that is, induced subgraphs which cannot be present. For example, *perfect graphs* are those which exclude all odd-length cycles (*odd holes*) of length at least 5, or their complements (*odd antiholes*) [7]. Thus, in particular, the class of perfect graphs contains all bipartite graphs, which exclude all odd holes and antiholes. All the graphs we consider here are bipartite, and hence perfect.

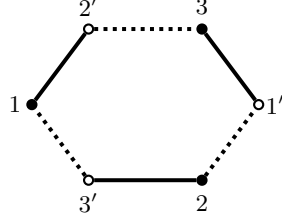
In our application, there is another technical reason for preferring to work with hereditary graph classes: in hereditary classes we have self-reducibility for most problems in $\#P$, including the permanent. This property implies equivalence between almost uniform sampling and approximate counting, referred to in Section 1.2. See [20] for details.

The switch chain is *ergodic* on a graph $G = (V, E)$ if the state space of the chain, the set of perfect matchings, is connected by switches. Importantly, we extend this to include graphs which have no perfect matching, where the state space is empty. Then we will say that a graph G is *hereditarily ergodic* if, for every $U \subseteq V$, the induced subgraph $G[U]$ is ergodic. A class of graphs will be called hereditarily ergodic if every graph in the class is hereditarily ergodic. For a hereditary graph class, this is clearly equivalent to saying that the switch chain is ergodic for every graph in the class.

Diaconis, Graham and Holmes [9] observed that the switch chain is not ergodic for all balanced bipartite graphs. They gave the example:

This graph has two perfect matchings, but the switch chain cannot move between them. This is because the graph is a chordless 6-cycle. In fact, this property characterises the class of graphs for which the switch chain is not ergodic, as we now show.

We say a graph G is *chordal bipartite* if it has no chordless cycle of length other than four. The class of chordal bipartite graphs is clearly hereditary. Note that the definition of chordal bipartite graphs is an excluded subgraph characterisation. To show that the switch chain is ergodic for this class, we require the



$$\begin{matrix} & 1' & 2' & 3' \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & \mathbf{1} & 1 \\ 1 & 0 & \mathbf{1} \\ \mathbf{1} & 1 & 0 \end{bmatrix} \end{matrix}$$

following “excluded submatrix” characterisation.

A Γ (Gamma) in a 0-1 matrix is an induced submatrix of the form

$$\Gamma : \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

A matrix is called Γ -free if it has no such induced submatrix. Lubiw [25] gave the following characterisation.

Theorem 1 (Lubiw). *A bipartite graph is chordal bipartite if and only if it is isomorphic to a graph G such that $A(G)$ is Γ -free.* \square

Moreover, Lubiw [25] showed that this characterisation can be used to recognise chordal bipartite graphs in $O(|E| \log |E|)$ time. This was subsequently improved to $O(|E|)$ time by Uehara [35]. For the switch chain we have:

Lemma 2. *The largest hereditary class of bipartite graphs for which the switch chain is ergodic is the class of chordal bipartite graphs. In this class, if $G = ([n] \cup [n]', E)$, the diameter of the chain is at most n .*

Proof. Clearly any graph with an induced cycle of length greater than 4 cannot be in the class, so we need only show ergodicity for chordal bipartite graphs. The chain is aperiodic, since there is a loop probability at least $1/n$ at each step, from choosing $i = j$ in step 2. Thus we must show that the chain is irreducible. From Theorem 1, we may suppose that $A(G)$ is given with a Γ -free presentation.

Let $\mathcal{G} = (\Omega, \mathcal{E})$ be the transition graph of the switch chain, with Ω the set of perfect matchings in G , and \mathcal{E} the set of transitions. We must show that \mathcal{G} is connected, and has diameter at most n . Let π and σ be any two perfect matchings in G , and let $\text{dist}(\pi, \sigma) = |\{i : \pi'_i \neq \sigma'_i\}|$. Note that $\text{dist}(\pi, \sigma) \leq n$, and $\text{dist}(\pi, \sigma) = 0$ implies $\pi = \sigma$.

Let k be the smallest index such that $\pi'_k \neq \sigma'_k$ and, without loss of generality, suppose $\pi'_k > \sigma'_k$. Then there exists $\ell > k$ such that $\pi'_\ell = \sigma'_k$, and hence $\pi'_\ell \neq \sigma'_\ell$. Then we have $(k, \sigma'_k), (k, \pi'_k), (\ell, \sigma'_k) \in E$, $\ell > k$ and $\pi'_k > \sigma'_k$.

$$\begin{matrix} & \pi'_\ell = \sigma'_k & \pi'_k \\ \begin{matrix} k \\ \ell \end{matrix} & \begin{bmatrix} 1 & 1 \\ 1 & ? \end{bmatrix} \end{matrix}$$

The Γ -free property of $A(G)$ then implies $(\ell, \pi'_k) \in E$. Thus we have $(k, \pi'_k), (\ell, \pi'_\ell) \in \pi$ and $(k, \pi'_\ell), (\ell, \pi'_k) \in E$. Therefore $\tau \in \Omega$ and $(\pi, \tau) \in \mathcal{E}$, where

$$\tau = \pi \setminus \{(k, \pi'_k), (\ell, \pi'_\ell)\} \cup \{(k, \pi'_\ell), (\ell, \pi'_k)\}.$$

Note that $\tau'_i = \pi'_i$ for $i \neq k, \ell$. However, $\pi'_k \neq \sigma'_k$, but $\tau'_k = \pi'_\ell = \sigma'_k$. Also $\pi'_\ell \neq \sigma'_\ell$, but $\tau'_\ell = \pi'_k = \sigma'_\ell$ if $\pi'_k = \sigma'_\ell$. Thus $\text{dist}(\pi, \sigma) - 2 \leq \text{dist}(\tau, \sigma) \leq \text{dist}(\pi, \sigma) - 1$. Hence there is a path of at most n edges in \mathcal{G} between any pair of matchings π, σ . Therefore the diameter of \mathcal{G} is at most n . \square

Computing the permanent exactly is known to be #P-complete for the class of chordal bipartite graphs [27], though this result does not extend even to chordal bipartite graphs of bounded degree. The complexity of exact computation of the permanent is unknown for all the subclasses of chordal bipartite graphs considered below, with the exception of bounded-degree *convex graphs*, which we consider in Section 2.2, and *chain graphs*, which we examine in Section 2.5.

2.2 Convex graphs

The largest class of graphs considered by Diaconis, Graham and Holmes [9] were those with “two-sided restrictions”. These are bipartite graphs G for which $A(G)$ has the *consecutive 1’s* property. These have been called *convex* graphs in the graph theory literature. They were introduced by Glover [13], who gave a simple greedy algorithm for finding a maximum matching in such a graph. The consecutive 1’s property can be recognised in $O(|E|)$ time by the well-known algorithm of Booth and Lueker [5].

A bipartite graph is *convex* if it is isomorphic to a graph $G = ([m] \cup [n]', E)$ such that $\mathcal{N}(i)$ is an interval $[\alpha'_i, \beta'_i] \subseteq [n]'$ for all $i \in [m]$. Note that this property remains true under an arbitrary permutation of $[m]$. Then

Lemma 3. *Convex graphs are a proper hereditary subclass of the class of chordal bipartite graphs.*

Proof. It is easy to see that the class CONVEX is hereditary. To see that it is a subclass of chordal bipartite graphs, we permute rows so that $\beta'_i \leq \beta'_j$ when $i < j$. Now we can show that $A(G)$ is Γ -free. If not, there is a Γ in some rows i, j and columns k', ℓ' .

$$\begin{array}{cc} & \begin{array}{cc} k' & \ell' \end{array} \\ \begin{array}{c} i \\ j \end{array} & \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \end{array}$$

We have $i < j$ but, since the rows of $A(G)$ have consecutive 1’s, $\beta'_i \geq \ell' > \beta'_j$. This contradicts our ordering of the rows of $A(G)$. To see that it is a proper subclass, note that there are at most $n! \binom{n}{2}^n = 2^{O(n \log n)}$ labelled convex graphs with n rows and columns, whereas Spinrad [30] has shown that there are $2^{\Theta(n \log^2 n)}$ chordal bipartite graphs. (Spinrad also gives in [30, Ex. 9.16(c)] an explicit example of a graph that is chordal bipartite but not convex.) \square

It is possible to give excluded subgraph and excluded submatrix characterisations of convex graphs, but we will not explore this here, since they are not easy to describe, and appear to have little algorithmic application. See [33] for details.

Blumberg [4] gave an $r^{O(r)} n^4$ bound on the mixing time of the switch chain for convex graphs with $r = \max_{i \in [n]} \deg(i)$. This is a hereditary subclass of convex graphs, since it is easy to see that graphs with bounded row- or column-degree form a hereditary subclass of any hereditary class. We will give *exact* algorithms for counting and sampling in this subclass of convex graphs. First we will show that these graphs also have bounded column degree.

Lemma 4. *Let $G = ([n] \cup [n]', E)$ be a convex graph containing a perfect matching. Let $r = \max_{i \in [n]} \deg(i)$ and $c = \max_{j \in [n]} \deg(j')$. Then we have $c \leq 2r - 1$.*

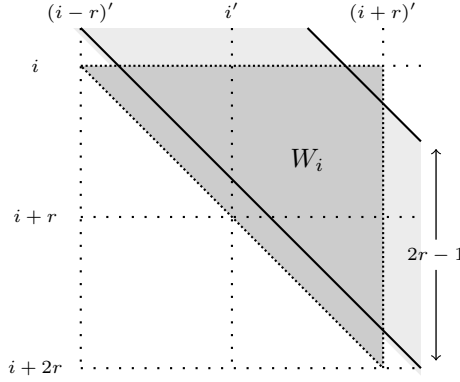
Proof. Let M be any perfect matching of G . We first permute the rows of $A(G)$ so that M is the diagonal of A , i.e. $M \leftarrow \{(i, i') : i \in [n]\}$. To bound c , consider any edge $(i, j') \in E$. Since G is convex, and $(i, i') \in E$, we have $i', j' \in [\alpha'_i, \beta'_i]$ and so $|i - j| \leq r - 1$. Hence $i \in [j - r + 1, j + r - 1] \cap [n]$, and so $\mathcal{N}(j') \subseteq [j - r + 1, j + r - 1] \cap [n]$. Therefore we have $c \leq 2r - 1$. \square

It is known that there is an exact algorithm for computing the permanent which is linear in n for all graphs of bounded *treewidth* [8, Thm. 1]. Convex graphs with $r = \max_{i \in [n]} \deg(i)$ have treewidth at most $2r - 1$. Unfortunately, the general algorithm of Courcelle, Makowsky and Rotics [8] is superexponential in the treewidth. An algorithm of Fürer [12, Thm. 3], for counting independent sets in graphs of bounded treewidth, could also be applied, since the treewidth of the line graph of a convex graph can be bounded by $8r^2$. (We will not prove these facts about treewidth here, since we do not use them.) Combined with Fürer’s algorithm, this gives an algorithm for the permanent which is linear in n , but exponential in r^2 .

However, we will not use either of these approaches, since the following dynamic programming algorithm has better time complexity for the problem at hand.

Lemma 5. *Let $G = ([n] \cup [n]', E)$ be a convex graph containing a perfect matching, and let $r = \max_{i \in [n]} \deg(i)$. Then, for any subgraph G^* of G , the permanent of $A(G^*)$ can be evaluated exactly in time $O(r^{2r}n)$. Hence the permanent can be evaluated in polynomial time for all convex graphs with degree bound $O(\log n / \log \log n)$.*

Proof. Let $A = A(G^*)$. The algorithm uses triangular windows W_i of width $2r + 1$ and height $2r + 1$, with corners at $A(i, (i - r)')$, $A(i, (i + r)')$ and $A(i + 2r, (i + r)')$. Note, from Lemma 4, that W_i cuts G as shown below. (When $i \leq r$ the window overhangs the left boundary of the matrix and when $i > n - 2r$ it overhangs the bottom boundary.) Moreover, for every edge of G there is an index i such that the corresponding entry of A appears in the window W_i . At iteration i of the algorithm, a *subperfect* matching Q will be a matching



of G^* , such that

1. Every row $j \leq i$ has a matching edge;
2. Every column $j' \leq \min\{(i + r)', n'\}$ has a matching edge;
3. No row $j > i + 2r$ has a matching edge;
4. No column $j' > (i + r)'$ has a matching edge.

Note that a subperfect matching cannot always be extended to a perfect matching of G^* . We consider the set

$$S_i = \{M : M = Q \cap W_i \text{ and } Q \text{ is a subperfect matching}\}.$$

Note that $|S_i| < (2r)!$, since each column of W_i is either empty or contains a unique edge in any of positions $1, 2, \dots, j$, for $j = 1, 2, \dots, 2r - 1$. For $M \in S_i$, let

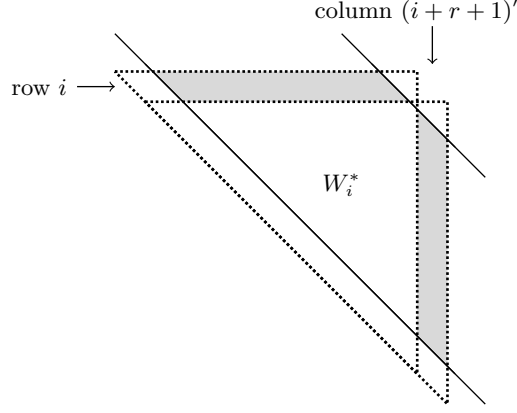
$$N_i(M) = |\{Q : Q \cap W_i = M\}|,$$

be the number of subperfect matchings represented by M . Initially, $i = 1$ and S_1 will be the set of all matchings in W_1 such that every vertex $j' \leq (r + 1)'$ has a matching edge. When $i = n - r$, all the subperfect matchings represented in W_{n-r} will be perfect matchings, and so we will have

$$\text{per}(A) = \sum_{M \in S_{n-r}} N_{n-r}(M).$$

We must show how to update the M and $N_i(M)$ from W_i to W_{i+1} . Let $W_i^* = W_i \cap W_{i+1}$. First we remove row i . We remove all $M \in S_i$ such that row i contains no matching edge, since they cannot correspond to a subperfect matching at iteration $(i + 1)$. Then we delete the matching edge in row i from M , for all $M \in S_i$. This will produce a set S_i^* of matchings in W_i^* ,

$$S_i^* = \{M : M = Q \cap W_i^* \text{ and } Q \text{ is a subperfect matching}\}.$$



We must now add column $(i + r + 1)'$ to W_{i+1} . For all $M^* \in S_i^*$, we attempt to augment each M^* with a matching edge e in column $(i + r + 1)'$. Note that e must be in W_{i+1} , and e can be in any row which has no matching edge in M^* . If no such row exists, we delete M^* from S_i^* , since it cannot correspond to a subperfect matching at iteration $i + 1$. Otherwise, for each possible choice of e , we add $M = M^* \cup \{e\}$ to S_{i+1} , and set

$$N_{i+1}(M) = \sum \{N_i(M^*) : M^* \in S_i, M^* \cap W_{i+1} = M \cap W_i\}.$$

This completes the description of the algorithm.

The operations in the update require $O(r|S_i|)$ time, except for the removal of duplicates, which can be implemented in $O(|S_i| \log |S_i|) = O(r^2|S_i|)$ time. Therefore, since

$$r^2|S_i| \leq r^2(2r)! \sim 2\sqrt{\pi} r^{5/2} (2r/e)^{2r} = O(r^{2r}),$$

and $O(n)$ updates must be performed, the overall time complexity of the algorithm is $O(r^{2r}n)$. This is polynomial in n if $r = O(\log n / \log \log n)$. \square

We can extend the algorithm of Lemma 5 to sample a matching uniformly at random. To do this, we must retain the sets S_i and the counts $N_i(M)$ ($M \in S_i$) used in the permanent evaluation. Then the sampling algorithm is a standard dynamic programming traceback through $S_{n-r}, \dots, S_i, \dots, S_1$, using the $N_i(M)$ to select matchings with the correct probability. See [10] for a more complete description of similar uses of traceback sampling. The time complexity for sampling a single matching is $O(\sum_i |S_i|) = O((2r)!n)$.

Thus dynamic programming seems preferable to Markov chain methods for sampling perfect matchings from convex graphs with small degree bound, at least if the chain must be run for its a guaranteed mixing time.

2.3 Biconvex graphs

Diaconis, Graham and Holmes [9] considered the following subclass of convex graphs. A graph $G = ([m] \cup [n]', E)$ is *biconvex* if it is convex and $\mathcal{N}(j')$ is an interval $[\alpha_{j'}, \beta_{j'}] \subseteq [n]$ for all $j' \in [n]'$. Thus $A(G)$ has the consecutive 1's property for both rows and columns.

Lemma 6. *Biconvex graphs are a proper hereditary subclass of convex graphs.*

Proof. It is easy to see that the class BICONVEX is a hereditary subclass of CONVEX. To see that it is a proper subclass, consider the example:

$$\begin{matrix} & 1' & 2' & 3' & 4' \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}.$$

In a biconvex ordering, row 2 must be adjacent to row 1 and row 3. Otherwise, columns 1' and 2' cannot be convex. But row 4 must also be adjacent to row 2, or column 3' cannot be convex. These conditions clearly cannot be satisfied simultaneously. \square

As with convex graphs, it is possible to give excluded subgraph and excluded submatrix characterisations of biconvex graphs. Since these are a little easier to describe than for convex graphs, we will give the excluded subgraph characterisation. Tucker shows [33, Thm. 10] that a bipartite graph is biconvex if and only if it does not contain the graphs I_n for $n \geq 1$, II_1 , II_2 , III_1 , III_2 and III_3 as induced subgraph. Here I_n is a chordless cycle C_{2n+4} , II_1 is the *triomino* and III_1 is the *tripod*.

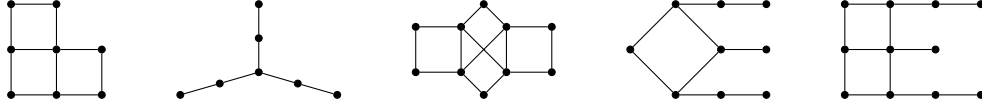


Figure 2: The triomino, the tripod and the graphs II_2 , III_2 and III_3

We know from Lemma 2 that the switch chain converges eventually on biconvex graphs, but how quickly is this guaranteed to occur? Unfortunately, the convergence may be exponentially slow. Both Matthews [26] and Blumberg [4] gave the following examples $\mathcal{G}_k = ([n] \cup [n]', \mathcal{E}_k)$, where $n = 2k - 1$:

$$(i, j') \in \mathcal{E}_k \iff \begin{cases} 1 \leq i < k \text{ and } k' \leq j' \leq (k+i)'; \\ i = k \text{ and } 1' \leq j' \leq n'; \\ k < i \leq n \text{ and } (i-k)' \leq j' \leq k'. \end{cases}$$

For example, \mathcal{G}_4 has biadjacency matrix

$$A(\mathcal{G}_4) = \begin{matrix} & \begin{matrix} 1' & 2' & 3' & 4' & 5' & 6' & 7' \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

Let π be any perfect matching. Then choosing $\pi'_k \leq k'$ forces $\pi'_i = (k+i)'$ for $i \in [k-1]$, and similarly choosing $\pi'_k \geq k'$ forces $\pi'_{k+i} = i'$ for $i \in [k-1]$. Thus the set of perfect matchings of \mathcal{G}_k is $S_1 \cup S_2$, where $S_1 = \{\pi : \pi'_k \leq k'\}$ and $S_2 = \{\pi : \pi'_k \geq k'\}$.

Clearly $S_1 \cap S_2 = \{\pi : \pi'_k = k'\} = \{\sigma\}$, for a single matching σ . Moreover, it is not difficult to show that there are 2^{k-1} ways to extend a partial matching π with $\pi'_i = (k+i)'$ for $i \in [k-1]$ to a perfect matching. One way is to note that the submatrix induced by rows $[k, n]$ and columns $[k]'$ is a *chain graph*, for which the permanent is easy to compute. (See Section 2.5 below and the formula presented there.) Thus $|S_1 \cap S_2| = 1$ and $|S_1| = |S_2| = 2^{k-1}$, and hence $|S_1 \cup S_2| = 2^k - 1$.

Therefore, if the switch chain is started at a random matching in S_1 , it will need $\Omega(2^n)$ time before it reaches σ , and it cannot enter S_2 before this occurs. This gives an $\Omega(2^n)$ lower bound on the mixing time of the chain. This argument can be made completely rigorous, see [4] or [26] for details.

2.4 Monotone graphs

Diaconis, Graham and Holmes [9] considered a structured subclass of biconvex graphs, which they called *monotone*, and showed that the switch chain is ergodic on graphs in this class. However, note that Lemma 2

gives a stronger result, for the larger class of chordal bipartite graphs. Diaconis, Graham and Holmes [9] conjectured further that the switch chain mixes rapidly in the class MONOTONE.

A bipartite graph $G = ([m] \cup [n]', E)$ will be called *monotone* if it is isomorphic to a convex graph such that $\alpha'_i \leq \alpha'_j$ and $\beta'_i \leq \beta'_j$ for all $i, j \in [m]$ with $i < j$. Thus $A(G)$ has a “staircase” structure. An example is shown in Fig. 3.

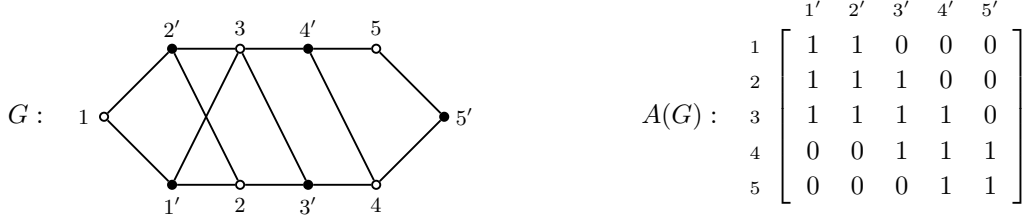


Figure 3: A monotone graph

We will always assume that G has the labelling of the definition, unless otherwise stated. First we show that, if G is row-monotone, it is also column-monotone. Recall that in a biconvex graph $\mathcal{N}(j')$ is an interval $[\alpha_{j'}, \beta_{j'}] \subseteq [n]$ for all $j' \in [n]'$.

Lemma 7. *A monotone graph is biconvex, and we have $\alpha_{i'} \leq \alpha_{j'}$, $\beta_{i'} \leq \beta_{j'}$ when $i', j' \in [n]'$ and $i' < j'$.*

Proof. For $j \in [n]$, let $s = \min\{i \in \mathcal{N}(j')\}$ and $t = \max\{i \in \mathcal{N}(j')\}$. If $s < i < t$, then $j' \geq \alpha'_t \geq \alpha'_i$ and $j' \leq \beta'_s \leq \beta'_i$, so $j' \in [\alpha'_i, \beta'_i] = \mathcal{N}(i)$ and hence $i \in \mathcal{N}(j')$. Thus $\mathcal{N}(j')$ is the interval $[s, t]$, so we may take $\alpha_{j'} = s$, $\beta_{j'} = t$. Hence $\alpha_{i'} = \min\{k : i' \in [\alpha'_k, \beta'_k]\}$ and $\alpha_{j'} = \min\{k : j' \in [\alpha'_k, \beta'_k]\}$, so $i' < j'$ implies $\alpha_{i'} \leq \alpha_{j'}$. Similarly, noting $\beta_{i'} = \max\{k : i' \in [\alpha'_k, \beta'_k]\}$ and $\beta_{j'} = \max\{k : j' \in [\alpha'_k, \beta'_k]\}$, we see that $i' < j'$ implies $\beta_{i'} \leq \beta_{j'}$. \square

Next we show a “forbidden submatrix” characterisation of monotone graphs, extending that of Lubiw [25] for chordal bipartite graphs.

Lemma 8. *A bipartite graph is monotone if and only if it is isomorphic to a graph G such that $A(G)$ has none of the following as an induced 2×2 submatrix:*

$$\Gamma \text{ (Gamma)} : \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \lrcorner \text{ (backwards L)} : \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad / \text{ (slash)} : \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Proof. Suppose G is monotone, but A contains Γ or $/$ in rows i and j , with $i < j$. Then row-convexity implies $\beta'_i > \beta'_j$, a contradiction. Similarly, if $A(G)$ contains a \lrcorner , then row-convexity implies $\alpha'_i > \alpha'_j$, again a contradiction. Thus, if G is a monotone graph, $A(G)$ cannot contain Γ , \lrcorner or $/$.

Now assume $A(G)$ contains no Γ , \lrcorner or $/$. Suppose $\mathcal{N}(i)$ is not an interval, so there exist $j' < k' < l'$ so that $(i, j'), (i, l') \in E$, but $(i, k') \notin E$. Since $\mathcal{N}(k') \neq \emptyset$, there exists $s \in [n]$ such that $(s, k') \in E$. If $s < i$, then $A(G)$ contains the first configuration below, which is either a Γ or a $/$, a contradiction. If $s > i$, then $A(G)$ contains the second configuration below, which is either a \lrcorner or a $/$, also a contradiction.

$$\begin{array}{cc} j' & k' \\ s & \begin{bmatrix} ? & 1 \\ 1 & 0 \end{bmatrix} \\ i & \end{array} \qquad \begin{array}{cc} k' & l' \\ i & \begin{bmatrix} 0 & 1 \\ 1 & ? \end{bmatrix} \\ s & \end{array}$$

Therefore suppose that $i < j$, but $\alpha'_i > \alpha'_j$. Then $A(G)$ contains the first configuration below, which is a \lrcorner or $/$, a contradiction. Similarly, if $\beta'_i > \beta'_j$, $A(G)$ contains the second configuration below, which is a Γ or $/$, again a contradiction. Hence G is monotone.

$$\begin{array}{c} \alpha'_j \quad \alpha'_i \\ i \begin{bmatrix} 0 & 1 \end{bmatrix} \\ j \begin{bmatrix} 1 & ? \end{bmatrix} \end{array} \qquad \begin{array}{c} \beta'_j \quad \beta'_i \\ i \begin{bmatrix} ? & 1 \end{bmatrix} \\ j \begin{bmatrix} 1 & 0 \end{bmatrix} \end{array}$$

□

□

A *bipartite permutation graph* is a permutation graph which is also bipartite. A graph $G = (V, E)$ is a *permutation graph* if there are permutations π, σ of V so that $(\pi_i, \pi_j) \in E$ if and only if $\pi_i < \pi_j$ and $\sigma_i > \sigma_j$. This can be given an *intersection* presentation, where π, σ are on parallel lines, and connected by lines (v, v) , for all $v \in V$. Then $(v, w) \in E$ if and only if corresponding lines (v, v) and (w, w) cross. An example is shown in Fig. 4 below.

Spinrad, Brandstädt and Stewart [31] studied this class of graphs, and gave $O(|E|)$ time algorithms for recognising membership in the class, and for constructing a crossing representation.

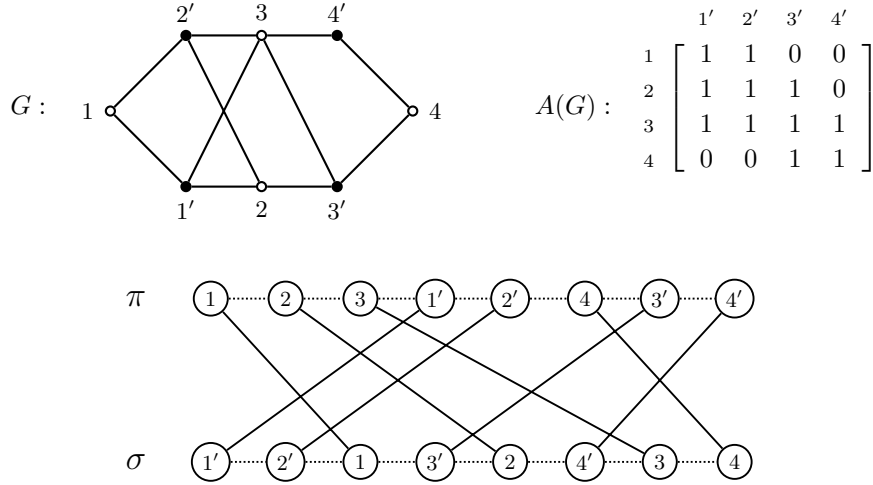


Figure 4: A bipartite permutation graph with its intersection representation

Our reason for introducing this class of graphs is that the bipartite permutation graphs are precisely the monotone graphs.

Lemma 9. *A graph is monotone if and only if it is a bipartite permutation graph.*

Proof. The condition of Lemma 8 is equivalent to the following. If $(i, k'), (j, \ell') \in E$ are such that $i < j$ and $k' > \ell'$, then $(i, \ell'), (j, k') \in E$. The conclusion now follows from the characterisation of bipartite permutation graphs given in [31], in particular Definition 3 and Theorem 1. □

Note that Lemma 8 is not a “forbidden subgraph” characterisation in the usual graph-theoretic sense. However, such a characterisation is known.

Lemma 10. [22, Lem. 1.46] *A graph is monotone if and only if it is chordal bipartite (i.e. it has no chordless cycle of length other than 4), and it contains none of the three graphs shown in Fig. 5 as an induced subgraph.* □

For example, the graph G given in Fig. 6 contains the armchair as a subgraph.

Lemma 11. *Monotone graphs are a proper hereditary subclass of biconvex graphs.*

Proof. The hereditary property follows easily from the definitions. The inclusion follows from Lemma 7, and strict inclusion follows from the example of Fig. 6. □

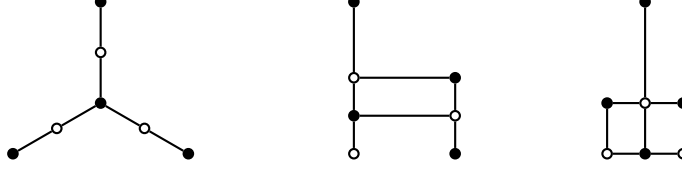


Figure 5: The tripod, the armchair and the stirrer.

$$A(G) = \begin{array}{c} \begin{array}{ccccc} & 1' & 2' & 3' & 4' & 5' \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array} \end{array} \quad \text{induced subgraph:} \quad \begin{array}{c} \begin{array}{ccc} & 3' & 4' & 5' \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array} \end{array}$$

Figure 6: A biconvex graph containing the armchair

To apply the switch chain to a monotone graph, we need to know whether it contains any perfect matching. If it does, we need to identify one efficiently, in order to start the chain. However, these are easy questions.

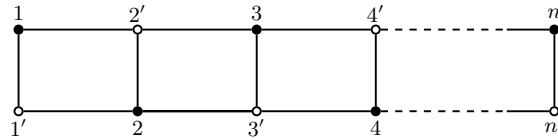
Lemma 12. *A monotone graph $G = ([n] \cup [n]', E)$ contains a perfect matching if and only if it contains the diagonal matching $\delta = \{(i, i') : i \in [n]\}$.*

Proof. We prove this by induction on n . If $n = 1$, then $E = \{(1, 1')\}$, and there is nothing to prove. So, suppose $n > 1$. Clearly $(1, 1') \in E$, or else either 1 or $1'$ is an isolated vertex, and hence G has no perfect matching. We will show that there is a perfect matching M^* which contains $(1, 1')$. Therefore, suppose that M is any perfect matching in G , with $(1, 1') \notin M$. Then $(1, j'), (i, 1') \in M$ for some $i \geq 2$, $j' \geq 2'$, and we have $(1, 1') \in E$. Hence $(i, j') \in E$, or else $A(G)$ would contain a Γ .

$$\begin{array}{c} \begin{array}{cc} & 1' & j' \\ \begin{array}{c} 1 \\ i \end{array} & \begin{bmatrix} 1 & \mathbf{1} \\ \mathbf{1} & ? \end{bmatrix} \end{array} \end{array}$$

Thus $M^* = M \setminus \{(1, j'), (i, 1')\} \cup \{(1, 1'), (i, j')\}$ is a perfect matching containing the edge $(1, 1')$. Now we use induction on the graph G^* given by deleting 1 and $1'$ from G , which contains the perfect matching $M^* \setminus \{(1, 1')\}$. \square

We will be particularly interested in *Hamiltonian* monotone graphs, and there is also an easy criterion for Hamiltonicity of a monotone graph. To state this, we consider the graph illustrated below, the *ladder* L_n .



Lemma 13. *L_n is a Hamiltonian monotone graph.*

Proof. Clearly L_n is bipartite, and $\mathcal{N}(1), \mathcal{N}(2), \mathcal{N}(3), \dots, \mathcal{N}(n-1), \mathcal{N}(n)$ are, respectively,

$$\{1', 2'\}, \{1', 2', 3'\}, \{2', 3', 4'\}, \dots, \{(n-2)', (n-1)', n'\}, \{(n-1)', n'\},$$

so are non-empty intervals satisfying the required ordering conditions. Finally, L_n has the Hamilton cycle $1' \rightarrow 2 \rightarrow 3' \rightarrow \dots \rightarrow n' \rightarrow n \rightarrow \dots \rightarrow 3 \rightarrow 2' \rightarrow 1 \rightarrow 1'$. \square

Then we have the following.

Lemma 14. *A monotone graph G is Hamiltonian if and only if it contains the ladder as a spanning subgraph.*

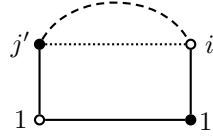
Proof. If G has a spanning ladder, the Hamilton cycle in the ladder is also a Hamilton cycle in G , and so G is Hamiltonian.

If $G = ([m] \cup [n]', E)$ is Hamiltonian, it has a perfect matching, so $m = n$ and G contains the diagonal matching δ , from Lemma 12. We will show by induction that G contains L_n , and so has a spanning ladder. The base case is $n = 2$. Then G must be a 4-cycle, so $G = L_2$.

If $n > 2$, consider any Hamilton cycle H in G . Vertices 1 and $1'$ lie on this cycle. There are two cases:

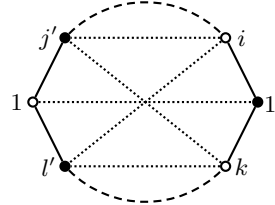
1. The cycle H contains the edge $(1, 1')$. Let $j' \neq 1'$ be adjacent on H to 1, and $i \neq 1$ be adjacent on H to $1'$. Since $i, j \geq 2$, biconvexity implies $(1, 2') \in E$ and $(2, 1') \in E$. Thus the three edges $(1, 1')$, $(1, 2')$, $(2, 1')$ of L_n are in E . Also $(i, j') \in E$, since G is Γ -free. Hence $i \rightarrow j' \rightarrow \dots \rightarrow i$ is a Hamilton cycle H^* in the monotone graph G^* obtained by deleting 1 and $1'$ from G .

$$1 \begin{array}{cc} & \begin{array}{cc} 1' & j' \end{array} \\ \left[\begin{array}{cc} \boxed{1} & 1 \\ 1 & 1 \end{array} \right] \\ i \end{array}$$



2. The cycle H does not contain the edge $(1, 1') \in E$. Let j', l' be the vertices of H adjacent to 1, and i, k the vertices of H adjacent to $1'$, so that H contains paths $i \rightarrow \dots \rightarrow j'$ and $k \rightarrow \dots \rightarrow l'$, avoiding 1 and $1'$. Now, since G is Γ -free, (i, j') , (i, l') , (k, j') , $(k, l') \in E$. Since $(1, 1') \in E$, and $(1, j') \in E$ for some $j \geq 2$, convexity implies that $(1, 2') \in E$. Similarly, since $(1, 1'), (i, 1') \in E$, with $i \geq 2$, convexity implies that $(2, 1') \in E$. Thus the three edges $(1, 1')$, $(1, 2')$, $(2, 1')$ of L_n are in E . Also $i \rightarrow \dots \rightarrow j' \rightarrow k \rightarrow \dots \rightarrow l' \rightarrow i$ is a Hamilton cycle H^* in the monotone graph G^* given by deleting 1 and $1'$ from G .

$$1 \begin{array}{ccc} & \begin{array}{ccc} 1' & j' & l' \end{array} \\ \left[\begin{array}{ccc} 1 & 1 & \boxed{1} \\ 1 & 1 & 1 \\ \boxed{1} & 1 & 1 \end{array} \right] \\ i \end{array}$$



In both cases, the edges $(1, 1')$, $(1, 2')$, $(2, 1')$ of L_n are in E , and we have a Hamiltonian monotone graph G^* with bipartition $[2, n] \cup [2, n]'$. It now follows by induction that G contains L_n . \square

2.5 Chain graphs

Diaconis, Graham and Holmes called the simplest class of graphs they considered “one-sided restriction” graphs. These are usually called *chain graphs* in the graph theory literature [39], and form a proper subclass of monotone graphs.

A chain graph is isomorphic to a graph $G = ([m] \cup [n]', E)$ where $\mathcal{N}(i) = [a_i]'$ for all $i \in [m]$, with $a_1 \leq a_2 \leq \dots \leq a_m$. Hence chain graphs are a subclass of monotone graphs, given by taking $\alpha'_i = 1'$, $\beta'_i = a'_i$, for all $i \in [n]$. An example is shown in Fig. 7.

The following easy fact is then true.

Lemma 15. $\mathcal{N}(j') = [b_j, m]$ for all $j' \in [n]'$ with $b_1 \geq b_2 \geq \dots \geq b_n$.

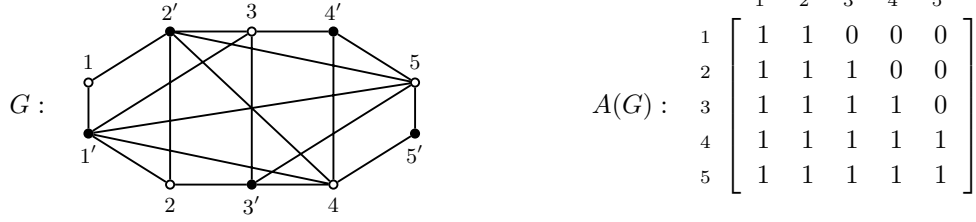


Figure 7: A chain graph

Proof. Since $a_i \leq a_{i+1}$, $(i, j') \in E$ implies $(i+1, j') \in E$. Let $b_j = \min\{i : a_i \geq j\}$. □

We note here that chain graphs have a very simple excluded subgraph characterisation. A bipartite graph is a chain graph if and only if it does not contain $2K_2$, the graph with four vertices and two disjoint edges, as an induced subgraph.

Lemma 16. *Chain graphs are a proper hereditary subclass of monotone graphs.*

Proof. The hereditary property and the inclusion are clear from the definition. The inclusion is strict because a chain graph is a monotone graph with $\alpha'_i = 1'$ for all $i \in [n]$. Hence the biadjacency matrix contains a column of 1's. But there exist monotone graphs which do not have a column of 1's. The graph in Fig. 3 is an example. □

Diaconis, Graham and Holmes [9] observed that there is a “classical” explicit formula for the permanent of a chain graph G . Of course, we must have $m = n$. Then, if $A = A(G)$,

$$\text{per}(A) = \begin{cases} 0, & \text{if } a_i < i \text{ for any } i \in [n]; \\ \prod_{i=1}^n (a_i - i + 1), & \text{otherwise.} \end{cases}$$

For example, if

$$A : \begin{matrix} & \begin{matrix} 1' & 2' & 3' & 4' & 5' \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}, \quad \text{then } \text{per}(A) = 2(3-1)(4-2)(5-3)(5-4) = 16.$$

After noting that the first a_1 columns of A are all 1's, and hence identical, this formula can be proved by an easy induction on the row order. The proof method can also be used to sample a perfect matching uniformly at random in $O(n)$ time.

Matthews [26] showed, using a coupling argument, that the mixing time of the switch chain for chain graphs is bounded by $O(n^3 \log n)$. Blumberg [4] gave a detailed study of the eigenvalues of the transition matrix of the switch chain for this class, based on earlier work of Hanlon [16].

These results clearly have little computational application, but establishing the mixing time of the switch chain for graphs in the class CHAIN is far from trivial. For example, there are chain graphs for which the original Jerrum-Sinclair Markov chain [18] has exponential mixing time. Consider the graph G for which $A(G)$ is lower triangular, so $A(i, j) = 1$ if $i \leq j$, $A(i, j) = 0$ otherwise. Then $\text{per}(A) = 1$, from the formula above, but the graph G^* given by deleting vertices 1 and n' has $\text{per}(A^*) = 2^{n-3}$ by the same formula, where $A^* = A(G^*)$. Thus G has one perfect matching, but an exponential number of near-perfect matchings. Therefore the algorithm of [18] will need exponential time to sample a perfect matching almost uniformly.

3 Analysis of the switch chain

In Section 2 we have seen that the hereditary graph classes considered by Diaconis, Graham and Holmes [9] form an ascending sequence:

$$\text{CHAIN} \subset \text{MONOTONE} \subset \text{BICONVEX} \subset \text{CONVEX} \subset \text{CHORDAL BIPARTITE}.$$

We know from Lemma 2 that the switch chain is ergodic for (balanced) bipartite graphs in all these classes, and has diameter at most n . Here we consider the mixing time of the switch chain for these classes, that is, the time required to reach near-stationarity. We will define this more formally below.

Here we take the Computer Science viewpoint of trying to distinguish only between polynomial time (rapid) mixing, and superpolynomial time (slow) mixing. We have observed that the switch chain may have exponential mixing time in the class BICONVEX, and the switch chain has mixing time $O(n^3 \log n)$ in the class CHAIN [26]. Therefore we need only determine whether or not the class MONOTONE has polynomial mixing time. The remainder of this section will be devoted to showing that this class does indeed have polynomial mixing time.

It is usual to measure deviation from the stationary distribution in terms of variation distance. For an ergodic Markov chain on state space Ω , transition probabilities $\mathcal{P} : \Omega^2 \rightarrow [0, 1]$ and stationary distribution π , the distance to stationary at time t , starting in state $x \in \Omega$, is $\Delta_x(t) = \max_{A \subset \Omega} |\mathcal{P}^t(x, A) - \pi(A)|$. Then the *mixing time*, $T_{\text{mix}}(\varepsilon)$, is the first time at which the Markov chain is ε -close to stationarity, maximised over the choice of starting state x :

$$T_{\text{mix}}(\varepsilon) = \max_{x \in \Omega} \min\{t : \Delta_x(t) \leq \varepsilon\}.$$

Two remarks about this definition. First, the function $\Delta_x(t)$ is monotonically non-increasing in t , so once the t -step distribution of the Markov chain is ε -close to stationarity, it remains that way. Second, the dependence on of T_{mix} on ε is weak, typically a multiplicative factor $\log \varepsilon^{-1}$, so it is usual to quote the mixing time as a function of the size of the problem instance, in our case n , and ignore the dependence on ε (or set ε to some conventional value, usually $\varepsilon = 1/e$). For more on mixing time, and the general context for this section of the paper, refer to [1, Chaps. 3 & 4] and [17, Chaps. 3 & 5].

3.1 Canonical paths and flows

Although there are other approaches to bounding the mixing time of Markov chains, here we will attempt only to apply the canonical paths approach of Jerrum and Sinclair [18]. For any symmetric Markov chain, this may be described briefly as follows.

Suppose the problem size is n . The method requires constructing paths of transitions of the chain $X = Z_1 \rightarrow Z_2 \rightarrow \dots \rightarrow Z_\ell = Y$, between every pair of states X and Y in the state space Ω , such that the length ℓ of each path is at most polynomial in n . This is often easy to achieve, but to obtain a good upper bound on mixing time it is essential that the paths are “spread out” over the state space, and do not overload any particular transition. The degree of success in achieving this end is measured by the congestion of the set of paths. Denote the (canonical) path from X to Y by γ_{XY} . Then the congestion ϱ of the chosen paths is given by

$$\varrho = \max_{(Z, Z^\dagger)} \left\{ \frac{1}{\pi(Z)\mathcal{P}(Z, Z^\dagger)} \sum_{X, Y : \gamma_{XY} \ni (Z, Z^\dagger)} \pi(X)\pi(Y) |\gamma_{XY}| \right\}, \quad (1)$$

where $|\gamma_{XY}|$ denotes the length of the path γ_{XY} , and the maximisation is over all transitions $Z \rightarrow Z^\dagger$, i.e., all pairs with $\mathcal{P}(Z, Z^\dagger) > 0$ and $Z^\dagger \neq Z$, and the sum is over all paths that use the transition $Z \rightarrow Z^\dagger$.

Polynomial mixing time will follow from a polynomial upper bound on congestion. For the switch Markov chain, where π is uniform over Ω , it can be seen from the definition of ϱ that this is equivalent to ensuring that the number of canonical paths through any transition is bounded by $|\Omega|$, possibly multiplied by a polynomial factor. A possible strategy for obtaining a good bound on congestion is therefore the following. Fix a transition $Z \rightarrow Z^\dagger$. For every canonical path γ_{XY} from X to Y that uses transition (Z, Z^\dagger) , specify an

encoding $W \in \Omega$, such that, given W and g additional bits of information, we can identify X and Y uniquely. It is common to refer to the additional information as “guesses”. In this setting, the mixing time T_{mix} of the chain can be bounded by $2^{O(g)} \text{poly}(n)$. Ideally, we seek $g = O(\log n)$, since this will give a polynomial bound on the mixing time.

The strategy just described is the one we will apply in this section to the switch chain on monotone graphs. In the proof of Lemma 2 we implicitly constructed canonical paths for the more general situation of chordal bipartite graphs. The counterexample \mathcal{G}_k from Section 2.3 demonstrates that the congestion of those canonical paths is in fact exponentially large, necessarily so even for biconvex graphs. Achieving low congestion in the monotone setting is a delicate matter, and will be addressed in the following section.

It only remains to present a precise relationship between congestion and mixing time, as given in [17, Cor. 5.9], based on Sinclair [29].

Lemma 17. *The mixing time of a symmetric Markov chain with uniform stationary distribution is bounded by*

$$T_{\text{mix}}(\varepsilon) \leq 2\rho(\ln |\Omega| + 2 \ln \varepsilon^{-1}),$$

where ρ is the congestion with respect to any set of canonical paths.

Technically, this result applies to a so-called “lazy” version of the Markov chain, in which which a fair coin is flipped at each step. A transition is made if the coin comes up “heads”, and no transition is made if the coin comes up “tails”. In our case, the loop probabilities, which are at least $1/n$, are sufficient to avoid the introduction of the lazy version. Then the factor 2 could be saved in the mixing time bound, but we will not use this refinement.

3.2 Construction of canonical paths

Our goal is to construct canonical paths between arbitrary pairs X, Y of perfect matchings in G . In general, $(V, X \cup Y)$ is a subgraph of G that is composed of alternating cycles $C_1 \cup \dots \cup C_s = X \oplus Y$, and isolated edges $X \cap Y$. Assume that these cycles are ordered deterministically in some way. (For example, order the cycles according to the smallest unprimed vertex in each cycle.) Then we may switch each of the cycles in order, using the procedure we will describe below. The isolated edges can be left untouched. Thus, it is sufficient to construct the canonical path for a single alternating cycle.

In fact, we may specialise the canonical path construction even further. Since MONOTONE is a hereditary class, if H is any alternating cycle in G , it is a Hamilton cycle in a smaller monotone graph $G[V(H)]$. Thus we will assume that $G[V(H)] = G$ in the remainder of this section, and let H be the (Hamilton) cycle with vertices $(u_1, v'_1, u_2, v'_2, \dots, u_n, v'_n)$. So our initial and final matchings are as follows:

$$X = \{(u_1, v'_1), (u_2, v'_2), \dots, (u_n, v'_n)\}$$

and

$$Y = \{(u_2, v'_1), (u_3, v'_2), \dots, (u_1, v'_n)\},$$

where we will choose $u_1 = n$ as the initial vertex of the cycle.

With each pair $(u, v') \in [n] \times [n]'$ (which may or may not correspond to an edge of G) we associate a point $p = (v, n - u + 1)$ in \mathbb{R}^2 . In particular, we associate the edges in X with the points $\{p_i = (v_i, n - u_i + 1) : i \in [n]\}$ and those in Y with points $\{q_i = (v_i, n - u_{i+1} + 1) : i \in [n]\}$ (interpreting u_{n+1} as u_1). We can think of this mapping as assigning Cartesian coordinates to the entries of the adjacency matrix $A(G)$ in such a way that the pattern of entries plotted in \mathbb{R}^2 looks exactly like the pattern of entries in the matrix $A(G)$ as conventionally written: the x coordinate increases with increasing column number, and the y -coordinate decreases with increasing row number. Denote the x - and y -coordinates of point $p \in \mathbb{R}^2$ by $x(p)$ and $y(p)$, so that $p = (x(p), y(p))$.

Let $P = \{p_1, p_2, \dots, p_n\} \cup \{q_1, q_2, \dots, q_n\} \subset [n]^2$. The alternating (Hamilton) cycle $X \cup Y$ corresponds to the cyclic sequence $(p_1, q_1, p_2, q_2, \dots, p_n, q_n)$ of points. Join the adjacent points in this sequence by line

segments (omitting the return segment from q_n to p_1) to yield a continuous path Π from p_1 to q_n . This path consists of alternating horizontal and vertical segments. By the choice $u_1 = n$ for the initial vertex we have that $y(p_1) = y(q_n) = 1$, i.e., that the path begins and ends at the lowest point (corresponding to the final row of the matrix). Somewhere along the way the path reaches the highest point $y(q_k) = y(p_{k+1}) = n$ (corresponding to the first row of the matrix).

The following lemma is inspired by the “mountain climbing problem” (see, for example [34]), and is proved in the Appendix.

Lemma 18. *Suppose Π is as above. There are continuous functions $\alpha, \beta : [0, 1] \rightarrow \Pi$ satisfying $\alpha(0) = p_1$, $\alpha(1) = q_k$, $\beta(0) = q_n$, $\beta(1) = p_{k+1}$, and $y(\alpha(t)) = y(\beta(t))$ for all $t \in [0, 1]$. Moreover the set of events*

$$T = \{t \in [0, 1] : \alpha(t) \in P \text{ or } \beta(t) \in P\}$$

has cardinality at most n^2 .

The lemma is tight in the sense that $|T|$ may be within $O(n)$ of the claimed upper bound; see Theorem 21, noting that n in that theorem is larger than n here by a factor 2. Note that the trajectories of $\alpha(t)$ and $\beta(t)$ will not, in general, move in a uniform sense along Π : it may be necessary for either or both of $\alpha(t)$ and $\beta(t)$ to retreat along Π for periods in order to make progress later.

Before entering into the details of the construction of canonical paths, here is a high-level overview. We regard the general points $[n]^2$, the special points P , and the path Π as forming a *board*, on which we play out a game involving n *tokens*. Movements of the tokens will mirror switches in the canonical path from X to Y . Place the tokens initially on the points p_1, p_2, \dots, p_n that represent the initial matching X . Our aim is to move these n tokens to the points q_1, q_2, \dots, q_n in a manner consistent with switches in the original graph. Thus, at each step, we will take two tokens and move them to new locations. Note that the two tokens we choose form the endpoints of the diagonal of some axis-aligned rectangle, say R . We remove these tokens and replace them on the endpoints of the opposite diagonal of R . For this operation to represent a valid switch in the original graph, it is necessary for the new locations for the tokens to correspond to 1’s in the matrix $A(G)$. Ensuring that this happens at every step requires the tokens to be moved in a very particular order. Describing this order is at the heart of the matter.

We think of each of $\alpha(t)$ and $\beta(t)$ as being a *focus* [of activity]. As time passes from 0 to 1, and points $\alpha(t)$ and $\beta(t)$ trace out their trajectories along Π , we shall move tokens in the neighbourhood of the foci according to certain rules. (Refer to Figure 8 for a pictorial description of such a rule.) If we remove the points $\alpha(t)$ and $\beta(t)$ from the path Π , we separate Π into three connected pieces. Denote the points in P lying in the middle piece by $P_U = P_U(t)$ and the remaining points by $P_L = P_L(t)$. Note that $P_L(t) \cup P_U(t) = P$, except at events $t \in T$, when $\alpha(t) \in P$ or $\beta(t) \in P$. We shall ensure that tokens lying in P_U are in their original locations (i.e., points of the form p_i), while those in P_L are in their final locations (i.e., points of the form q_i). As time passes, $P_L(t)$ tends to increase in size and $P_U(t)$ tends to decrease, corresponding to tokens passing from the initial points $\{p_1, \dots, p_n\}$ to the final points $\{q_1, \dots, q_n\}$.

The arrangement of tokens on the board at time t (viewed as a subset of points in $[n]^2$) will be called the *configuration* of tokens and denoted $\sigma = \sigma(t)$. As we want σ to correspond to a perfect matching in G , we insist that it contain one point from every row $y = i \in [n]$ of the board and one point from every column $x = j \in [n]$. In what follows, it may be worth bearing in mind the basic underlying strategy, which is to keep the tokens as far as possible on the points P . If this can be done effectively, it will be possible to provide an encoding of the current state (in the sense described of the previous Section 3.1) by forming a perfect matching W guided by the points of P that are not in the current configuration σ .

As the foci trace out their trajectories, there will be periods (open time intervals) when both $\alpha(t)$ and $\beta(t)$ are both on (open) vertical segments, and during these periods $y(\alpha(t))$ and $y(\beta(t))$ are either both monotonically increasing or both monotonically decreasing; call these *v-periods*. The v-periods are separated by *h-periods*, closed time intervals during which one of $\alpha(t)$ or $\beta(t)$ is stationary and the other is moving horizontally. During v-periods, $\sigma(t)$ is constant and well defined, in a manner that will be explained presently. We do not examine the configuration during h-periods, so its definition there is not important; one could decree, e.g., that the configuration there is the same as during the previous v-period.

During a v-period, neither $\alpha(t)$ nor $\beta(t)$ is a member of P ; as a consequence, $P_L(t) \cup P_U(t) = P$, so that every point in P is assigned either to $P_L(t)$ or $P_U(t)$. So assume that $\alpha(t)$ and $\beta(t)$ are both on (open) vertical segments. For convenience we introduce a local naming scheme around $\alpha(t)$ and $\beta(t)$. Denote by a_1 and a_2 the lower and upper ends of the line segment containing $\alpha(t)$. Continue the numbering $\dots, a_0, a_1, a_2, a_3, \dots$ along the path Π as far as necessary; this convention provides a local numbering of some subsequence of $p_1, q_1, \dots, p_n, q_n$, or its reversal. In a similar way, introduce names for the points around $\beta(t)$, with b_1 and b_2 being the lower and upper ends of the line segment containing $\beta(t)$.

Suppose $\sigma \subset [n]^2$ is a configuration of tokens. A *hole-pair* is a pair H of adjacent points $H = \{p_i, q_i\}$ or $H = \{q_i, p_{i+1}\}$ of P such that $\sigma \cap H = \emptyset$. As the foci move, we attempt to maintain the following Invariant I, which is a conjunction of I1–I3:

I1 $\{a_1, a_2\}$ is a hole-pair.

I2 If $x(a_1) < x(b_1)$ then $\{b_2, b_3\}$ is a hole-pair; otherwise $\{b_0, b_1\}$ is a hole-pair.

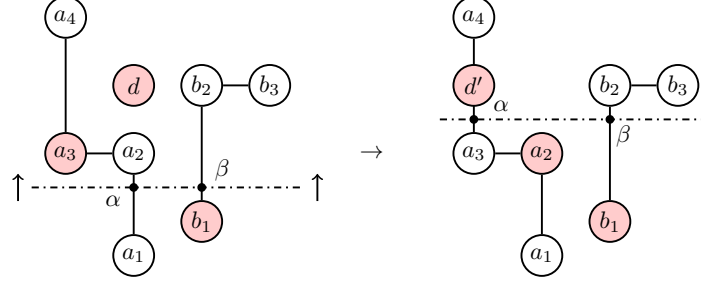
I3 There are no hole-pairs beyond these two.

A number of consequences follow from I1–I3: (i) $\sigma(t)$ is completely determined by $\alpha(t)$ and $\beta(t)$, (ii) $|\sigma \cap P| = n - 1$, (iii) $\sigma \cap P_L \subseteq \{q_1, q_2, \dots, q_n\}$, and (iv) $\sigma \cap P_U \subseteq \{p_1, p_2, \dots, p_n\}$. (Starting from a hole-pair, and working around Π , successive points in P are forced to be alternately in and out of σ ; this demonstrates (i), and the other conclusions follow as byproducts of this argument.) Invariant I may fail after a token-switch, but when this happens it will be immediately reinstated at the following switch, as we shall see. There will generally be a single token (and exceptionally three) lying outside P (i.e., $|\sigma \setminus P| = 1$ or exceptionally $|\sigma \setminus P| = 3$). The position of such a token(s) will be called a *dislocation*, and is denoted d (or d' or d'') in Figures 8–11 below.

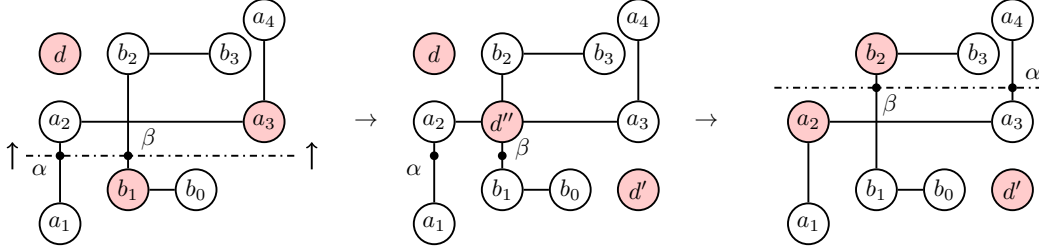
Initially $\sigma = \{p_1, \dots, p_n\}$ so Invariant I will certainly not be satisfied at the outset. A similar remark applies to the final configuration $\sigma = \{q_1, \dots, q_n\}$. We will see how to finesse this issue later. For the time being, we assume that we are under way, that Invariant I is satisfied, and that t is in a v-period, so that $\alpha(t)$ and $\beta(t)$ are on vertical line segments (a_1, a_2) and (b_1, b_2) . We assume first that $\alpha(t)$ (and hence $\beta(t)$) is moving upwards, i.e., that the rate of change of $y(\alpha(t))$ is positive. (It transpires that the situation when $\alpha(t)$ is moving downwards creates little extra work, as it can be handled by symmetry, specifically by rotating the board through an angle π . More detail will be provided immediately after we have dealt with the focus-moving-upwards situation.) Depending on the ordering of $y(a_2)$ and $y(b_2)$ one of two events occurs first: either $y(\alpha(t)) = y(a_2)$ (an α -event) or $y(\beta(t)) = y(b_2)$ (a β -event). We consider the situation just before this event and just after, and, in particular, what action needs to be taken to maintain the invariant. (By “just after” here, we mean “at a point in time just after the start of the next v-period”.)

We proceed by case analysis. There are eight cases I–IV and I*–IV*. First we split on whether $x(a_1) < x(b_1)$ (Cases I and II) or $x(a_1) > x(b_1)$ (Cases III and IV), i.e., on the horizontal relationship between $\alpha(t)$ and $\beta(t)$ before the event. Then we split on whether $y(a_2) < y(b_2)$ (Cases I and III) or $y(a_2) > y(b_2)$ (Cases II and IV), i.e., on whether it is an α -event or a β -event that occurs. Finally we split on whether $\alpha(t)$ and $\beta(t)$ have the same horizontal relationship after the event (unstarred cases) or opposite (starred cases). Note that the eight Cases I–IV* are exhaustive. We consider the eight cases in turn below. Cases I and I* illustrate most of the issues, so we treat them in more detail; the remaining cases will leave more for the reader. Some shorthand terminology will be useful in describing these cases. Let $p, q \in \mathbb{R}^2$ be points. We say that p is *to the left of* q (and q is *to the right of* p) if $y(p) = y(q)$ and $x(p) < x(q)$. We say that p is *above* q (and q is *below* p) if $x(p) = x(q)$ and $y(p) > y(q)$.

Case I. This case is characterised by $x(a_1) < x(b_1)$, $y(a_2) < y(b_2)$ and $x(a_3) < x(b_1)$. (Refer to Figure 8, which illustrates the layout of the board in the vicinity of the foci.) In this figure and the later ones the conventions are as follows. The dotted-and-dashed line is $y = y(\alpha(t)) (= y(\beta(t)))$ and marks the current y -coordinate of the foci. Points with tokens are grey/red and points without tokens are white. The “before” picture is to the left and the “after” to the right. The current location of the dislocation is marked d . Note that d must always be the intersection of the vertical line through a_1 and a_2 , and either the horizontal line through b_0 and b_1 , or through b_2 and b_3 , as appropriate; this is a consequence of the fact that the board contains a token on every horizontal and vertical line.



Case I applies when $x(a_1) < x(b_1)$, $y(a_2) < y(b_2)$ and $x(a_3) < x(b_1)$.



Case I* applies when $x(a_1) < x(b_1)$, $y(a_2) < y(b_2)$ and $x(a_3) > x(b_1)$.

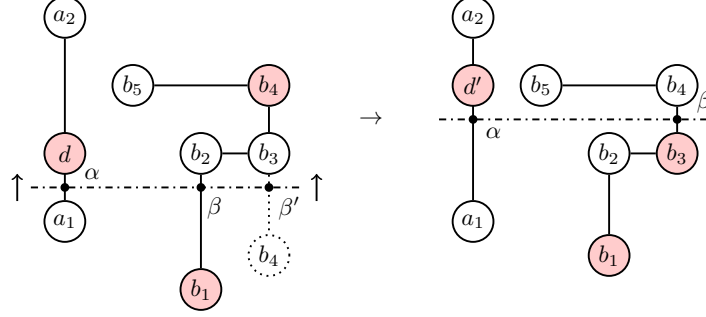
Figure 8: Cases I and I*

The action to be taken in this case is to switch the tokens at a_3 and d . Recall that “switch”, in the token interpretation, means drawing the axis-aligned rectangle with a_3 and d at the endpoints of one diagonal, and moving the tokens to the endpoints of the other diagonal. In this case, the tokens end up at a_2 and a new dislocation d' . The most demanding check to make is that the dislocation d' is legally placed after the switch. Recall that, to be legal, d' must be a point corresponding to a 1 in the matrix $A(G)$. This part of the argument will always appeal to monotonicity. In this instance we see that the new dislocation d' is above a_3 and to the left of b_2 ; since a_3 and b_2 both correspond to 1s in the matrix then so does d' , by monotonicity (Lemma 8). Once this point has been established, and noting that the new hole-pairs are $\{a_3, a_4\}$ and $\{b_2, b_3\}$, it is easy to check that I1–I3 have been preserved.

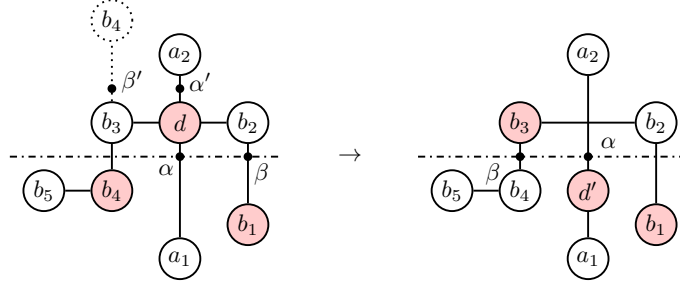
A note of caution. When checking the invariant, it is important to bear in mind that there are no relations between points beyond the ones explicitly stated, and the ones implied by the line segments being parallel to the axes. Thus, in the first row of Figure 8, d' is left of b_2 since $x(d') = x(a_3) < x(b_1) = x(b_2)$. However, a_3 (respectively, b_3) may be to the left or right of a_2 (respectively b_2), and a_4 may be above or below a_3 . (In case a_4 is below b_4 , the foci will end up below where they have been drawn, but on the same vertical lines; that is, α and a_3 will be on the same vertical line, as will β and b_1 .) It is crucial not to make use of any unintended relationships between points. It would have been impractical (and unnecessary) to display all the possible combinations.

Case I.* This case is similar to the previous one, but with $x(a_3) > x(b_1)$ replacing $x(a_3) < x(b_1)$. The fact that $\alpha(t)$ starts to the left of $\beta(t)$ and ends up on the right creates an additional complication, whose resolution necessitates an extra switch.

The first action to be taken is to switch the tokens at a_3 and b_1 creating two new dislocations at d' and d'' . This is legal because d' is below a_3 and to the right of b_1 , and because d'' is to the left of a_3 and above b_1 . (Note that we use here both of the valid modes of reasoning arising from monotonicity.) At this point we have three dislocations, which is a worst case. Note that we have temporarily violated I3, but we will put this right at the following step, when we switch d and d'' . This switch requires no justification, as we are moving tokens to their official locations in P . As an aside, we remark that a_4 may be below a_3 , in which case $\alpha(t)$ and $\beta(t)$ will end up lower than drawn in the figure. In any case, it is routine to check that Invariant I is satisfied after the second switch.



Case II applies when $x(a_1) < x(b_1)$, $y(a_2) > y(b_2)$ and $x(a_1) < x(b_3)$. Only perform this switch if $y(b_4) > y(b_3)$; otherwise, just move focus β to β' .



Case II* applies when $x(a_1) < x(b_1)$, $y(a_2) > y(b_2)$ and $x(a_1) > x(b_3)$. Only perform this switch if $y(b_4) < y(b_3)$; otherwise, just move the foci to α' and β' .

Figure 9: Cases II and II*

Case II. This case is characterised by $x(a_1) < x(b_1)$, $y(a_2) > y(b_2)$ and $x(a_1) < x(b_3)$. (Refer to Figure 9.) A new feature here is that the switch is conditional, and is only made when b_4 is above b_3 . In that case, we switch b_4 and d ; this is legal because the new dislocation d' is above a_1 and to the left of b_4 .

If b_4 is below b_3 (indicated by the ghostly b_4 in the figure) then no switch is required to maintain the invariant; all that is needed is to move the focus β rightwards to β' .

Case II.* This case is similar to the previous, but with $x(a_1) > x(b_3)$ replacing $x(a_1) < x(b_3)$. Again the switch is conditional, this time only being made when b_4 is below b_3 . In that case we switch b_4 and d ; this is legal because the new dislocation d' is below a_2 and to the right of b_4 . Otherwise we just move the foci to α' and β' .

Case III. This case is characterised by $x(a_1) > x(b_1)$, $y(a_2) < y(b_2)$ and $x(a_3) > x(b_1)$. (Refer to Figure 10.) Here we switch a_3 and d ; this is legal because the new dislocation d' is below a_3 and to the right of b_1 .

Case III.* This case is similar to the previous, but with $x(a_3) < x(b_1)$ replacing $x(a_3) > x(b_1)$. First we switch a_3 and b_2 , creating new dislocations at d' and d'' ; this is legal because the positions of d' and d'' relative to a_3 and b_2 . We finish by switching d and d'' .

Case IV. This case is characterised by $x(a_1) > x(b_1)$, $y(a_2) > y(b_2)$ and $x(a_1) > x(b_3)$. (Refer to Figure 11.) Here and in the final case, we have an obligatory switch followed by a conditional one. First we switch b_2 and d , which is legal because the new dislocation d'' is below a_2 and to the right of b_2 . If b_4 is above b_3 (indicated by the ghostly b_4 in the middle frame of the sequence), we stop at this point, with the foci at α' and β' . Otherwise (b_4 is below b_3), we switch b_4 and d'' , which is legal as the new dislocation d' is below a_2 and to the right of b_4 .

Case IV.* This case is similar to the previous, but with $x(a_1) < x(b_3)$ replacing $x(a_1) > x(b_3)$. First we switch b_2 and d , which is legal because the new dislocation d'' is below a_2 and to the right of b_2 . If b_4 is

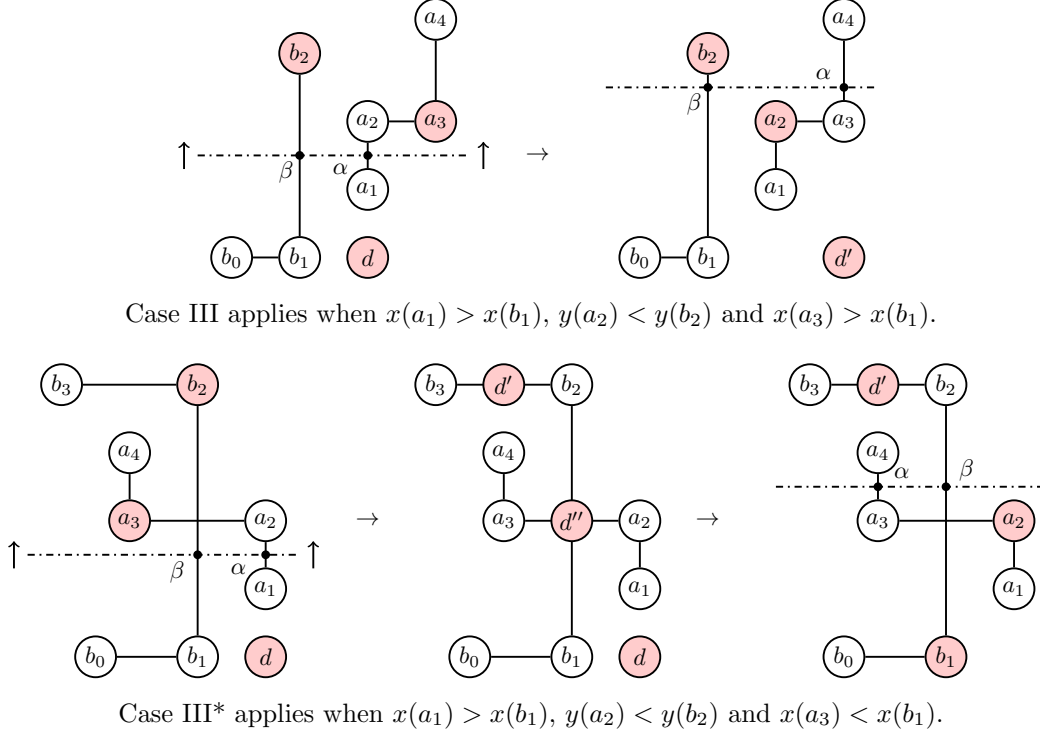


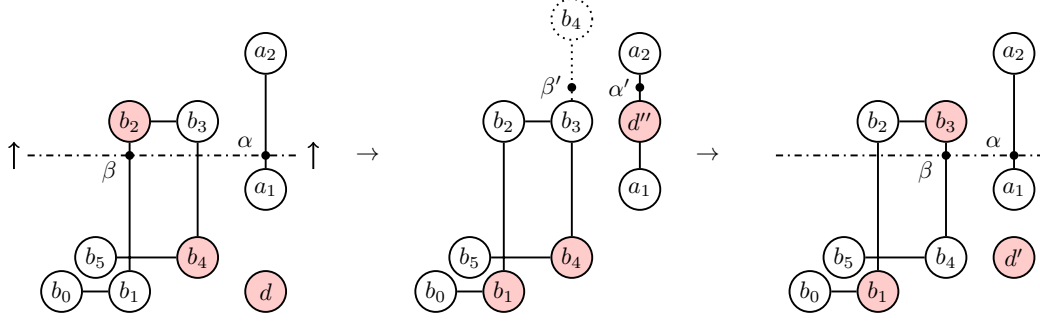
Figure 10: Cases III and III*

below b_3 , we stop at this point, with the foci at α' and β' . Otherwise (b_4 is above b_3), we switch b_4 and d'' , which is legal as the new dislocation d' is above a_1 and to the left of b_4 .

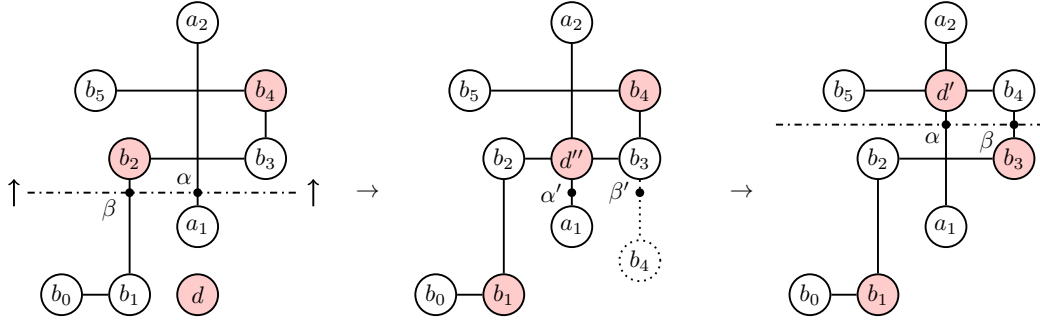
Cases I–IV* exhaust all the possibilities with $\alpha(t)$ is moving upwards. To deal with the situation when $\alpha(t)$ is moving downwards, we appeal to symmetry. The argument is as follows. Rotate the board through an angle of π , an operation which is equivalent to the change of coordinates $x := n - x + 1$ and $y := n - y + 1$. Note that, relative to the new coordinates, the configuration is still legal, satisfying Invariant I. Note also that the underlying matrix is still monotonic, since monotonicity of a matrix is preserved by rotation through π . However, $\alpha(t)$ is now moving upwards in the new coordinate system. Now apply the switch or switches dictated by the relevant one of the Cases I–IV* described above. Finally rotate the board back through an angle π to obtain a new configuration satisfying Invariant I.

We now have to deal with the issue that the initial and final configurations do not satisfy the invariant. (There are no hole-pairs.) To get the procedure above going, we need to create one horizontal and one vertical hole-pair. We make space for this by extending the board and adding three “virtual points” p'_1 , p_{n+1} and q_{n+1} . Suppose the coordinates of p_1 are $(h, 1)$. Delete the point p_1 and add points $p'_1 = (h, 0)$, $p_{n+1} = (n + 1, 1)$ and $q_{n+1} = (n + 1, 0)$. (In terms of the graph G , this corresponds to adding new vertices $n + 1$ and $(n + 1)'$ and new edges $(n + 1, h')$, $(n, (n + 1)')$ and $(n + 1, (n + 1)')$, together with any further edges implied by monotonicity.) Integrate these new points into the path Π as illustrated in Figure 12. Add a token to q_{n+1} and leave the existing tokens in place (including the one at the point formerly known as p_1 , which now becomes dislocation d). Place the foci α and β just below d and p_{n+1} . The invariant is now satisfied, and the arrangement of tokens within the original extent of the board is unchanged. We can now start the canonical path construction as described earlier. A similar construction (rotated through π) works to finish the path.

Two important points to note. The new board is consistent with monotonicity. (In this context, recall that the edge (n, n') must be present in G , and hence the point $(n, 1)$ is a valid position for a token.) Moreover, even though p_1 has been placed to the left of q_n in the figure, the construction works equally well with p_1 to the right.



Case IV applies when $x(a_1) > x(b_1)$, $y(a_2) > y(b_2)$ and $x(a_1) > x(b_3)$. Only perform the second switch if $y(b_4) < y(b_3)$; otherwise, stop after the first with the foci at α' and β' .



Case IV* applies when $x(a_1) > x(b_1)$, $y(a_2) > y(b_2)$ and $x(a_1) < x(b_3)$. Only perform the second switch if $y(b_4) > y(b_3)$; otherwise, stop after the first with the foci at α' and β' .

Figure 11: Cases IV and IV*

3.3 Encoding and congestion

In Lemma 18, reverse the role of α and β , so that $\alpha(0) = q_n$, $\alpha(1) = p_{k+1}$, $\beta(0) = p_1$ and $\beta(1) = q_k$. Place the n tokens initially on the points $\{q_1, \dots, q_n\}$, and denote the configuration at time t by $\sigma'(t)$. Since the trajectories of $\alpha(t)$ and $\beta(t)$ are oblivious of the tokens, $P_L = P_L(t)$ and $P_U = P_U(t)$ are unchanged. According to the invariant, the configuration σ' satisfies $\sigma' \cap P_U \subseteq \{q_1, \dots, q_n\}$, $\sigma' \cap P_L \subseteq \{p_1, \dots, p_n\}$ and $|\sigma' \cap P| = n - 1$. At any legal time t , then,

$$|(\sigma(t) \cup \sigma'(t)) \cap P| = 2n - 2. \quad (2)$$

Consider a canonical path $X = Z_0 \rightarrow Z_1 \rightarrow \dots \rightarrow Z_\ell = Y$ constructed by the method of Section 3.2. Some of Cases I–IV* involved making two switches; in these cases, call the middle configuration between the two switches (and the corresponding perfect matching Z_i) *transitory*. Note the configurations that are not transitory are of the form $\sigma(t)$ for some t , and these configurations satisfy Invariant I. If Z_i is not transitory, consider a time t at which configuration $\sigma(t)$ corresponds to Z_i . Then $\sigma'(t)$ is a near complement to $\sigma(t)$ with respect to P , and its corresponding perfect matching Z'_i is a near complement to Z_i with respect to $X \cup Y$; specifically, from equation (2),

$$|(Z_i \cup Z'_i) \cap (X \cup Y)| = 2n - 2.$$

Suppose (Z, Z^\dagger) is a valid transition (switch) of the Markov chain. In the usual way, we wish to provide every canonical path through (Z, Z^\dagger) with a unique encoding, and hence bound the total number of paths using (Z, Z^\dagger) . In fact, our encoding will be an element of $\Omega \times [8n^2]$. So suppose the transition of interest occurs at (Z_i, Z_{i+1}) in a canonical path of the above form from X to Y . We suppose first that $C = X \cup Y$ is a single

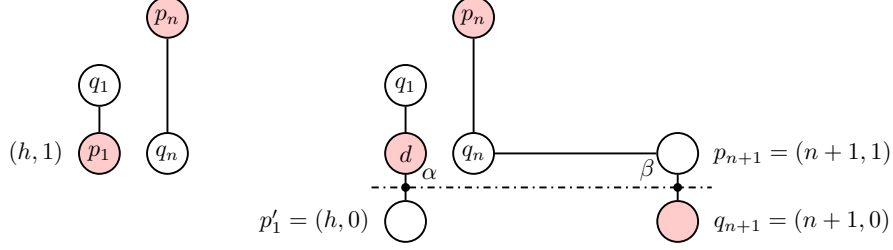


Figure 12: Starting the canonical path with the aid of extra “virtual points”

cycle. At least one of Z_i and Z_{i+1} is not transitory, say, Z_i . Our encoding will be the near complement Z'_i together with some additional data. Consider $C' = Z_i \cup Z'_i$. Noting that $|C' \setminus C| \leq 2$, the first piece of extra data we provide is the identity of the (at most) two edges in $C' \setminus C$ that need to be deleted from C' in a bid to recover C . These edges cannot be either the top or bottom horizontal edges, since these remain in C' throughout. Thus there are at most $\binom{2n-2}{2} + (2n-2) + 1 = 2(n^2 - n + 1) \leq 2n^2$ possibilities for $C' \setminus C$. Now we need to add two edges, but there are only two ways this can be done so that the result is a cycle. Finally, we need to signal that the near complement was taken with respect to Z_i and not Z_{i+1} : a further two possibilities. This gives us our encoding within the set $\Omega \times [8n^2]$.

In general, the symmetric difference of X and Y is a union of several cycles, say $X \oplus Y = C_1 \cup \dots \cup C_k \cup \dots \cup C_s$, where the cycles are presented in the agreed order, and C_k is the cycle currently being switched using the procedure described above. To construct the perfect matching $W \in \Omega$ that forms the main element of the encoding, proceed as follows. Except on the cycle C_k , the matching W is given by the symmetric difference of X , Y and Z_i . Thus, denoting the vertex set of C_k by V_k , and the complement $V \setminus V_k$ by $\overline{V_k}$, we have $W[\overline{V_k}] = (X \oplus Y \oplus Z_i)[\overline{V_k}]$. On C_k itself, $W[V_k]$ is given by the single-cycle construction from Section 3.2. The additional data is encoded by an integer in $[8n^2]$ as before. Note that we can reconstruct the common edges $X \cap Y$ of the initial and final states, and all the cycles of $X \oplus Y$ except C_k , by examining $Z_i \cap W$ and $Z_i \oplus W$. Knowing that C_1, \dots, C_{k-1} have been switched and C_{k+1}, \dots, C_s have not, we may then recover $X[\overline{V_k}]$ and $Y[\overline{V_k}]$. Finally, we recover $X[V_k]$ and $Y[V_k]$ as in the single-cycle case. Summarising, there are at most $8n^2|\Omega|$ canonical paths using any given transition.

We have all the quantities needed for the calculation of the congestion ϱ . From the definition of the switch chain, $\mathcal{P}(Z, Z^\dagger) = 2n^{-2}$. From Lemma 18, the maximum length of a canonical path is n^2 . Substituting these values into (1) yields

$$\varrho \leq |\Omega|^{-1} (n^2/2) (8n^2) |\Omega| n^2 = 4n^6.$$

Then, from Lemma 17, noting that the state space Ω has cardinality at most $n!$, we obtain the sought for bound on mixing time.

Theorem 19. *The switch Markov chain has mixing time $\tau(\varepsilon) < 8n^6(n \ln n + 2 \ln \varepsilon^{-1}) = O(n^7 \log n)$ for any graph $G = ([n] \cup [n]', E)$ in the class MONOTONE.*

The big-O estimate comes from the observation that, if we were to set ε smaller than $1/n!$, say, we could not distinguish between the output of the chain and the uniform distribution within polynomial time. By comparison with Theorem 19, the algorithm of [2] has running time $O(n^7 \log^4 n)$, with no bound given on the implied constant. It may be possible to improve the analysis here, but it is highly unlikely that we could match the $O(n^2 \log n)$ bound conjectured by Diaconis, Graham and Holmes [9]. Currently, this has not even been established for chain graphs.

4 Conclusions

The hierarchy of hereditary graph classes studied in this paper provides a framework for understanding the ergodicity and mixing time of the switch chain for perfect matchings on bipartite graphs. Unfortunately,

while we have identified the largest class for ergodicity, it seems difficult to characterise the largest hereditary class that supports polynomial time mixing. Indeed, it is not even clear that such a class exists.

A simpler question would be to exhibit proper superclasses of monotone graphs for which the switch chain mixes rapidly. From the examples \mathcal{G}_k given in subsection 2.3, we see that such a superclass cannot contain all biconvex graphs. More precisely, it cannot contain \mathcal{G}_k for all k . Thus we might consider the class which excludes \mathcal{G}_k for some k , and try to determine the largest k for which this class has rapid mixing. Note that the class excluding \mathcal{G}_k also excludes \mathcal{G}_ℓ for all $\ell > k$, since \mathcal{G}_k is an induced subgraph of \mathcal{G}_ℓ .

The graph \mathcal{G}_2 is monotone. Therefore the class of \mathcal{G}_2 -free biconvex graphs, that is, biconvex graphs that have no induced subgraph isomorphic to \mathcal{G}_2 , is not a superclass of monotone graphs. However, for every fixed $k > 2$ the class of all \mathcal{G}_k -free biconvex graphs does contain all monotone graphs. This can be proved by observing inclusions among the forbidden subgraphs in Figs. 2 and 5. Unfortunately, we know very little about the structure of the graphs in these classes, let alone whether their structure might support rapid mixing.

APPENDIX

A Proof of Lemma 18

A.1 The parallel mountain climbers problem

Let $P = (p_1, p_2, \dots, p_n)$ be an ordered sequence of points $p_i = (x_i, y_i) \in \mathbb{R}^2$ such that $y_1 = y_n$ and $y_i \geq y_1$ ($i \in [n]$). The line segments $L_i = [p_i, p_{i+1}]$ ($i \in [n-1]$) will be called *slopes*. We will assume that $y_i \neq y_j$ for $i \neq j$, unless $\{i, j\} = \{1, n\}$. Otherwise, we replace y_i by $y_i + \varepsilon^i$ ($1 < i < n$), for small enough $\varepsilon > 0$. (It is not really necessary to remove vertical slopes, since we will see that they can be identically to the others.) Similarly, we can avoid vertical slopes by replacing the x_i by $x_i + \delta^i$ for some small $0 < \delta < \varepsilon$. The *range* Π is then the point set $\bigcup_{i=1}^{n-1} L_i \subset \mathbb{R}^2$. The set $P \subset \Pi$ will be called the *nodes* of the range. Observe that Π is a connected one-dimensional manifold with boundary $\{p_1, p_n\}$, in which the nodes of P appear in order.

The *height* $y : \Pi \rightarrow \mathbb{R}$ ($p \in \Pi$) is the piecewise linear function defined by interpolation on the y_i ($i \in [n]$). Thus, if $p \in L_i$ is such that $p = \lambda p_i + (1 - \lambda)p_{i+1}$, then $y(p) = \lambda y_i + (1 - \lambda)y_{i+1}$. A node p_i will be called a *peak* if it is a local maximum of y , a *valley* if it is a local minimum of y , and *monotone* otherwise. The *summit* p_s ($s \in [n]$) of Π is the highest peak, so $y_s = \max\{y(p) : p \in \Pi\}$.

We will say that two ranges Π, Π' are *isomorphic* if there is bijection between P and P' such that $y'_i < y'_j$ if and only if $y_i < y_j$. The bijection is extended linearly to the L_i . If $p = \lambda p_i + (1 - \lambda)p_{i+1} \in L_i$, then $p' = \lambda p'_i + (1 - \lambda)p'_{i+1} \in L'_i$. Hence, for any range Π , there is an isomorphic range Π' such that $p_i = (i, y_i - y_1)$ ($i \in [n]$). For illustration, we will generally use this *standard presentation* of Π . Then, for example, we will say a point $v_1 \in \Pi$ is to the left or right of a point $v_2 \in \Pi$ if it is true in the standard presentation.

An example range with $n = 11$ is shown below, first in the standard presentation, and then in an isomorphic presentation. Here p_3, p_5, p_7, p_{10} are peaks, $p_1, p_4, p_6, p_9, p_{11}$ are valleys, and p_2, p_8 are monotone. The summit is p_7 . We wish to show the existence of continuous functions $\alpha, \beta : [0, 1] \rightarrow \Pi$, with $\alpha(0) = p_1$, $\beta(0) = p_n$, $\alpha(1) = \beta(1) = p_s$, such that $y(\alpha(t)) = y(\beta(t))$ for all $t \in [0, 1]$. An *event* will be any value of t such that $\alpha(t) \in P$ or $\beta(t) \in P$, and we will let T be the set of events. Our assumption that $y_i = y_j$ only if $i = j$ or $\{i, j\} = \{1, n\}$ implies that, for any p_i ($1 < i < n$), $y(p) = y_i$ only if $p = p_i$ or p is in the interior of some line segment L_j . This can be viewed as the intersection of a horizontal line through p_i with the line segment L_j . If this intersection exists, we will denote it by $p_i L_j$. The set of such intersections determines all potential events. It is a superset of T , and is clearly preserved under isomorphism.

Note that, in the interval (t_j, t_{j+1}) between two successive events, we may take α, β to be functions defined by linear interpolation:

$$\alpha(t) = \frac{(t_{j+1} - t)\alpha(t_j) + (t - t_j)\alpha(t_{j+1})}{t_{j+1} - t_j}, \quad \beta(t) = \frac{(t_{j+1} - t)\beta(t_j) + (t - t_j)\beta(t_{j+1})}{t_{j+1} - t_j}.$$

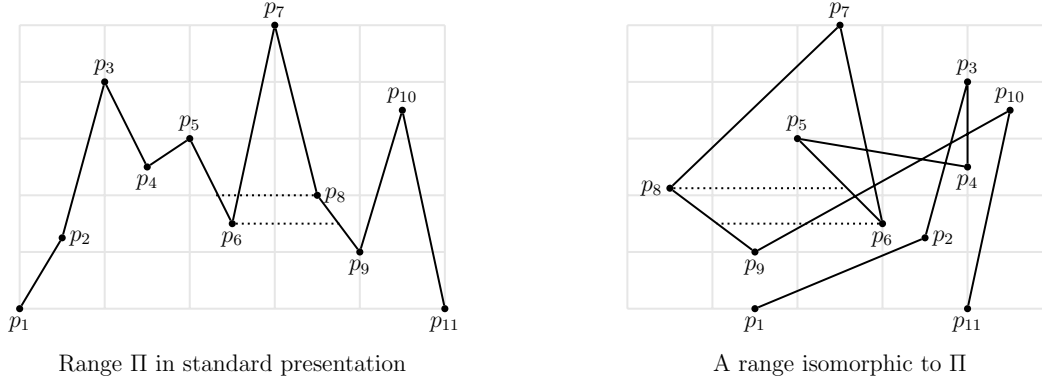


Figure 13: Example

Thus α, β may be taken as piecewise linear functions. These functions clearly transform in an obvious way, under isomorphism. They are completely determined by the set T . Hence we wish to determine a tight upper bound on $|T|$.

This is usually posed as the (discrete parallel) *mountain climbers problem*, and we will use its terminology here. (See, for example, [15, 28, 34, 37].) For example, we refer to the parameter t as “time”. Two climbers, starting at p_1 and p_n , attempt to meet by moving continuously on the slopes of the standard presentation of the range Π , so that they are always at the same height. Note that one climber must always remain to the left of the summit, and the other to the right, since if either reaches the summit, so must the other. Thus, if they meet, they must meet at p_s . Can they always meet and, if so, how many events are required to ensure that they meet?

Note that (a version of) this problem first arose in analysis, as a problem concerning continuous functions. (See, for example, [21, 24, 38].) This continuous version is somewhat different. There are pathological cases in which the climbers cannot meet and, even when they can, there may be no finite bound on the number of events.

There appears to be no worst-case analysis of the discrete version of this problem in the literature, although [15] gave an $O(n^2)$ upper bound on the number of events. Since this problem is central to the canonical path construction in this paper, we give an analysis here.

A.2 Upper bound

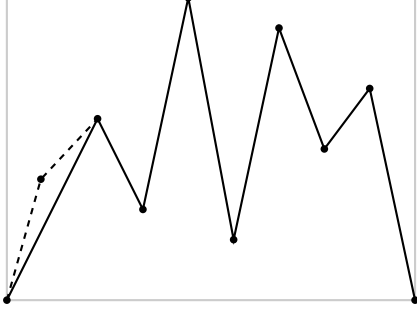
The *range graph* [34] $\mathcal{RG}(\Pi)$ has a vertex for each intersection $p_i L_j$, with $1 < i < s \leq j < n$ or $1 \leq j < s < i < n$. These are intersections from points to the left of the summit with slopes to the right, and vice versa, representing all possible events during the climb. We also add vertices for the initial and final states $p_1 p_n$ and $p_s p_s$. It is easy to see that this vertex set is the same for isomorphic ranges and can have cardinality at most $(n-1)(n-2) + 2$, by simple counting. However, this bound can be improved.

Lemma 20. *There are at most $(n-1)^2/4 + 1$ vertices in $\mathcal{RG}(\Pi)$.*

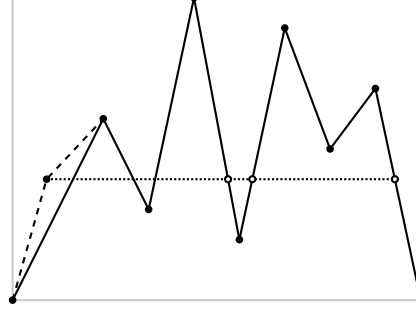
Proof. Suppose there are n_1 slopes to the left of the summit, and n_2 to the right, so $n_1 + n_2 = n - 1$. Let $N(n_1, n_2)$ be the maximum number of vertices in $\mathcal{RG}(\Pi)$. We will show by induction that $N(n_1, n_2) \leq n_1 n_2 + 1$. First, it is easy to see that $N(1, n_2) \leq n_2 + 1$.

Next, if any of the nodes from p_2 to p_{s-1} is monotone, consider the range given by removing such a node. Its range graph will have at most $N(n_1 - 1, n_2) \leq (n_1 - 1)n_2 + 1$ vertices, by induction. Adding back the removed node gives rise to at most n_2 new vertices, as shown below (in the case that the removed node is p_2), so $N(n_1, n_2) \leq (n_1 - 1)n_2 + 1 + n_2 = n_1 n_2 + 1$ vertices. Finally, if there are no monotone nodes and

hence p_2 is a peak, consider the range given by removing both node p_2 and node p_3 , which is necessarily a

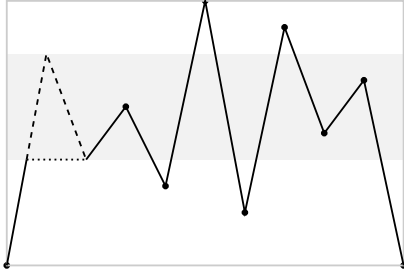


Adding a leftmost monotone node

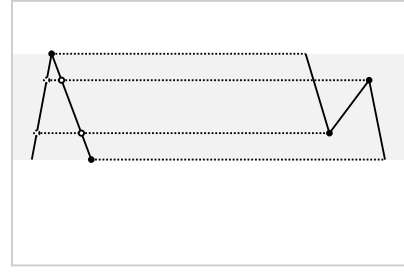


Vertices arising from the new node

valley. Its range graph will have $N(n_1 - 2, n_2) \leq (n_1 - 2)n_2 + 1$ vertices, using the induction. Adding back p_2 and p_3 gives the situation shown below.



Adding a new leftmost peak



Vertices arising from the j th path

The only new vertices will be in the window determined by the slope to the right of the new first peak, i.e., between p_2 and p_3 . The window partitions the range to the right of the summit into some number r of disjoint paths. Suppose that the j th path has $n_{2,j}$ slopes ($j = 1, \dots, r$), so $\sum_{i=1}^r n_{2,i} \leq n_2$. Then it is easy to see that the j th path gives rise to exactly $2n_{2,j}$ new vertices. Thus the total number of vertices added is $2\sum_{i=1}^r n_{2,i} \leq 2n_2$. Thus $N(n_1, n_2) \leq N(n_1 - 2, n_2) + 2n_2 \leq (n_1 - 2)n_2 + 1 + 2n_2 = n_1n_2 + 1$, and we need only observe that n_1n_2 has a maximum of $(n - 1)^2/4$, when $n_1 = n_2 = (n - 1)/2$. \square

The edges of $\mathcal{RG}(\Pi)$ correspond to possible intervals between successive events. There is an edge between two vertices in $\mathcal{RG}(\Pi)$ if and only if there is a parallel move of the two climbers which would take them from one vertex to the other. Thus there is an edge between vertices p_iL_j and p_kL_r if either $k \in \{i - 1, i + 1\}$ and $j = r$, or $j \in \{k - 1, k\}$ and $r \in \{i - 1, i\}$. For example, p_6L_8 is adjacent to p_8L_5 in the range graph of the range of Fig. 13 above, as illustrated.

It is clear that each vertex, other than p_1p_n and p_sp_s , is adjacent to exactly two other vertices, corresponding to moving left or right on the range at a peak or valley, and up or down from a monotone node. Thus all vertices in the graph have degree 2, apart from p_1p_n and p_sp_s , which have degree 1, since it is only possible to move up from p_1p_n , or down from p_sp_s . Hence the range graph comprises a unique path joining p_1p_n to p_sp_s , and possibly some disjoint cycles. The number of edges on this path is at most $(n - 1)^2/4$. Note that the path from p_1p_n to p_sp_s in the range graph corresponds precisely to the sequence of events in the canonical path of Section 3.2.

The full range graph for the example with $n = 11$ in Fig. 13 above has 22 vertices, and comprises a path and a single cycle, with 14 and 8 vertices, respectively:

The upper bound on the number of vertices with $n = 11$ is $10^2/4 + 1 = 26$. So the 22-vertex range graph has fewer vertices than the upper bound, and the 14-vertex path has considerably fewer.

This example illustrates that $\mathcal{RG}(\Pi)$ need not be connected, and may have fewer vertices than the upper bound. In fact, the range graph may have as few as $n - 1$ vertices, for example when $s = 2$. There cannot

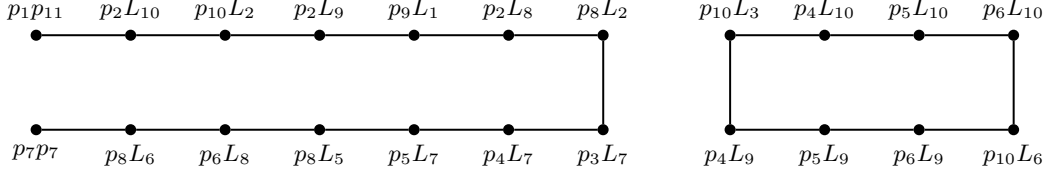


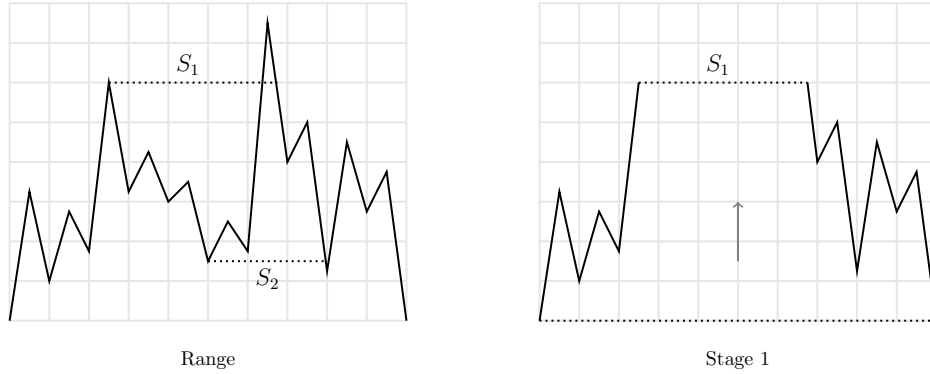
Figure 14: The range graph for the example of Fig. 13

be fewer, since there must be at least one vertex for each of the $(n - 3)$ nodes other than p_1, p_s, p_n , plus the two vertices p_1p_n, p_sp_s .

Note also that the effect on $\mathcal{RG}(\Pi)$ of removing the perturbations ε^j , used to make heights unique, is to merge pairs of vertices p_iL_{j-1} and p_iL_j into a single vertex p_ip_j . The merged vertex may have degree 4 in the range graph, so the component of $\mathcal{RG}(\Pi)$ containing p_1p_n and p_sp_s may contain cycles. In this case, the path from p_1p_n to p_sp_s in $\mathcal{RG}(\Pi)$ may no longer be unique. See [34] for an example.

A.3 Alternative proof

The proof given above is essentially that of Tucker [34]. Other authors (e.g. [15]) define the range graph to include all intersections p_iL_j , and even all “intersections” p_ip_i , but use essentially the same proof. Unfortunately, the range graph proof does not provide a lower bound on the worst case, since the component structure of $\mathcal{RG}(\Pi)$ is opaque. To do this, we give an equally easy proof of the existence of a path from p_1p_n and p_sp_s , by induction on n , which will lead to a lower bound in Section A.4 below. If $n = 3$, there is a single peak, and nothing to prove. Otherwise, we divide the problem into three stages. Assume, without loss of generality, that the second highest peak p_r is to the left of the summit p_s , so $r < s$. Let the deepest valley between p_r and p_s be p_t , so $r < t < s$. We construct the horizontal line S_1 from p_r to the slope L_s to the right of p_s , and the horizontal line S_2 from p_t to the first slope encountered to its right. This produces three subproblems, climbing up from p_1p_n to summit S_1 , climbing down from S_1 to “summit” S_2 , and then climbing up from S_2 to the true summit p_sp_s . By “climbing down” we mean replacing the height function y with $-y$.

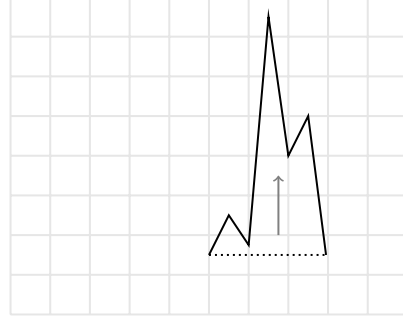


Each stage has a range with fewer nodes than the original problem, so each will have a path from the base to the summit in the corresponding range graph, by induction. Joining these paths together gives a path for the original problem. Note that the summits S_1 and S_2 in stages 1 and 2 must be shrunk to a point so that the ranges correspond to the definition, but this does not affect the path.

The uniqueness of the path is less easy to establish using this proof. However, we have already shown uniqueness in A.2 above, using the properties of the range graph, so we will not consider this further here.



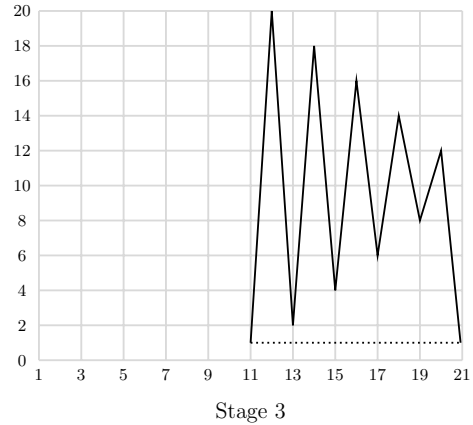
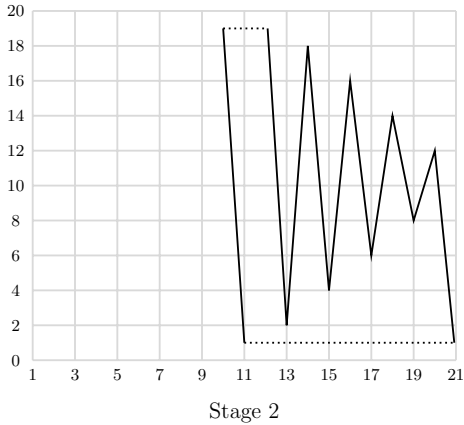
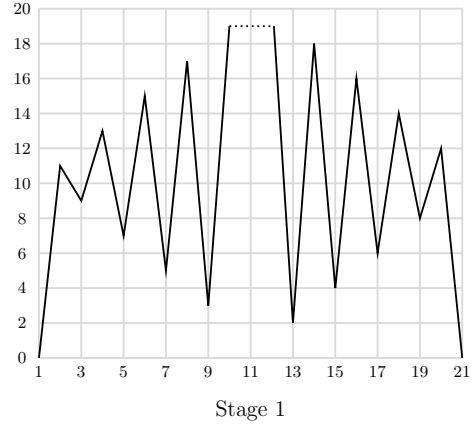
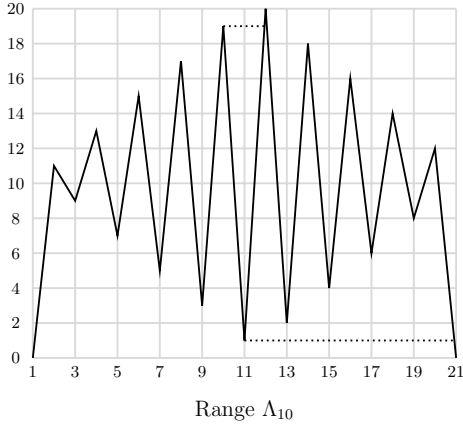
Stage 2



Stage 3

A.4 Lower bound

For $n = 2k + 1 \geq 3$, define the range Λ_k with $p_1 = (0, 0)$, $p_{2k+1} = (2k, 0)$, k peaks p_i at $(i, \min\{k + i - 1, 3k - i + 2\})$ ($i = 2, 4, \dots, 2k - 2$), and $k - 1$ valleys p_i at $(i, \max\{k - i + 2, i - k - 1\})$ ($i = 1, 3, \dots, 2k - 1$). If k is odd, $p_k = (k + 1, 2k)$ is the summit, and $p_{k+2} = (k + 2, 2k - 1)$ is the second highest peak. If k is even, $p_{k+2} = (k + 2, 2k)$ is the summit, and $p_k = (k, 2k - 1)$ is the second highest peak. Thus $s = k + 1$ if k is odd, and $s = k + 2$ if k is even. The case $n = 21$ is shown below, together with the three ranges constructed in the inductive proof above.



The range of stage 1 can be transformed to Λ_{k-1} by minor changes of coordinates. Clearly stages 2 and 3

each contribute $(k-1)$ edges to the path in $\mathcal{RG}(\Lambda_k)$. Thus, if T_k is the number of edges in the path from $p_1 p_{2k+1}$ to $p_s p_s$ in $\mathcal{RG}(\Lambda_k)$, we have $T_1 = 1$, and $T_k = T_{k-1} + 2(k-1)$, from which we can easily deduce $T_k = k(k-1) + 1 = (n-1)(n-2)/4 + 1$. The path from $p_1 p_n$ to $p_s p_s$ is unique, and has path length $(n-1)(n-2)/4$ in $\mathcal{RG}(\Lambda_k)$. Observe that this closely matches our upper bound $(n-1)^2/4$.

We can compute the number of vertices in $\mathcal{RG}(\Lambda_k)$ exactly. If k is even (resp. odd), the total number of horizontal intersections from peaks to the left (resp. right) of p_s with slopes to its right (resp. left) is

$$(k-1) + (k-3) + (k-5) + \cdots + 3 + 1.$$

The total number of horizontal intersections from valleys to the left (resp. right) of p_s with slopes to its right (resp. left) is

$$(k-2) + (k-4) + (k-6) + \cdots + 4 + 2.$$

These total $k(k-1)/2$, whether k is odd or even. The total number of horizontal intersections from peaks to the right (resp. left) of p_s with slopes to its left (resp. right) is

$$(k-1) + (k-3) + (k-5) + \cdots + 3 + 1,$$

and the total number of horizontal intersections from valleys to the right (resp. left) of p_s with slopes to its left (resp. right) is

$$(k-2) + (k-4) + (k-6) + \cdots + 4 + 2.$$

These also total $k(k-1)/2$. So the number of vertices in $\mathcal{RG}(\Lambda_k)$ is $k(k-1) + 2$. Hence $\mathcal{RG}(\Lambda_k)$ is a single path from $p_1 p_n$ to $p_s p_s$.

To summarise, we have proved

Theorem 21. *Let $R(n)$ be the maximum possible number of events required for two parallel climbers to meet at the summit on a range Π with n nodes. Then $R(n) = n^2/4 - O(n)$.*

From this we may deduce

Lemma 18. *Let Π be as defined in Section A.1. There are continuous functions $\alpha, \beta : [0, 1] \rightarrow \Pi$ satisfying $\alpha(0) = p_1$, $\alpha(1) = q_k$, $\beta(0) = q_n$, $\beta(1) = p_{k+1}$, and $y(\alpha(t)) = y(\beta(t))$ for all $t \in [0, 1]$. Moreover the set of events*

$$T = \{t \in [0, 1] : \alpha(t) \in P \text{ or } \beta(t) \in P\}$$

has cardinality at most n^2 .

Proof. Π is a range with $2n$ nodes $\{p_1, q_1, p_2, q_2, \dots, p_n, q_n\}$. Hence, by Lemma A.1, $\mathcal{RG}(\Pi)$ has at most $(2n)^2/4 = n^2$ vertices. Observe, from Theorem A.2, that this bound cannot be significantly improved in the worst case. \square

Acknowledgements

We thank Mary Cryan for useful discussions.

References

- [1] D. ALDOUS AND J. A. FILL, *Reversible Markov chains and random walks on graphs*, 2002. <http://www.stat.berkeley.edu/~aldous/RWG/book.html>. Unfinished monograph, recompiled 2014.
- [2] I. BEZÁKOVÁ, D. STEFANKOVIC, V. V. VAZIRANI, AND E. VIGODA, *Accelerating simulated annealing for the permanent and combinatorial counting problems*, SIAM Journal on Computing, 37 (2008), pp. 1429–1454.

- [3] N. BHATNAGAR, D. RANDALL, V. VAZIRANI, AND E. VIGODA, *Random bichromatic matchings*, in LATIN 2006: Theoretical Informatics, vol. 3887 of Lecture Notes in Computer Science, Springer, 2006, pp. 190–201.
- [4] O. BLUMBERG, *Permutations with interval restrictions*, PhD thesis, Stanford University, 2012.
- [5] K. S. BOOTH AND G. S. LUEKER, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, Journal of Computer and System Sciences, 13 (1976), pp. 335–379.
- [6] A. BRANDSTÄDT, V. B. LE, AND J. P. SPINRAD, *Graph classes: a survey*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.
- [7] M. CHUDNOVSKY, N. ROBERTSON, P. SEYMOUR, AND R. THOMAS, *The strong perfect graph theorem*, Annals of Mathematics, 2006 (164), pp. 51–229.
- [8] B. COURCELLE, J. A. MAKOWSKY, AND U. ROTICS, *On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic*, Discrete Applied Mathematics, 108 (2001), pp. 23–52.
- [9] P. DIACONIS, R. GRAHAM, AND S. P. HOLMES, *Statistical problems involving permutations with restricted positions*, in State of the art in probability and statistics, vol. 36 of Lecture Notes–Monograph Series, Institute of Mathematical Statistics, 2001, pp. 195–222.
- [10] M. DYER, *Approximate counting by dynamic programming*, in Proc. 35th Annual ACM Symposium on Theory of Computing, ACM Press, 2003, pp. 693–699.
- [11] B. EFRON AND V. PETROSIAN, *Nonparametric methods for doubly truncated data*, Journal of the American Statistical Association, 94 (1999), pp. 824–834.
- [12] M. FÜRER, *A natural generalization of bounded tree-width and bounded clique-width*, CoRR, abs/1402.1810 (2014).
- [13] F. GLOVER, *Maximum matching in a convex bipartite graph*, Naval Research Logistics Quarterly, 14 (1967), pp. 313–316.
- [14] M. C. GOLUMBIC, *Algorithmic graph theory and perfect graphs*, vol. 57 of Annals of Discrete Mathematics, North-Holland, Amsterdam, 2004.
- [15] J. E. GOODMAN, J. PACH, AND C. K. YAP, *Mountain climbing, ladder moving, and the ring width of a polygon*, Amer. Math. Monthly, 96 (1989), pp. 494–510.
- [16] P. HANLON, *A random walk on the rook placements on a Ferrer’s board*, The Electronic Journal of Combinatorics, 3 (1996), p. R26.
- [17] M. JERRUM, *Counting, sampling and integrating: algorithms and complexity*, ETH Zürich Lectures in Mathematics, Birkhäuser, Basel, 2003.
- [18] M. JERRUM AND A. SINCLAIR, *Approximating the permanent*, SIAM Journal on Computing, 18 (1989), pp. 1149–1178.
- [19] M. JERRUM, A. SINCLAIR, AND E. VIGODA, *A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries*, Journal of the ACM, 51 (2004), pp. 671–697.
- [20] M. JERRUM, L. VALIANT, AND V. VAZIRANI, *Random generation of combinatorial structures from a uniform distribution*, Theoretical Computer Science, 43 (1986), pp. 169–188.
- [21] T. KELETI, *The mountain climbers’ problem*, Proceedings of the American Mathematical Society, 117 (1993), pp. 89–97.
- [22] E. KÖHLER, *Graphs without asteroidal triples*, PhD thesis, Technische Universität Berlin, 1999.

- [23] D. A. LEVIN, Y. PERES, AND E. L. WILMER, *Markov chains and mixing times*, American Mathematical Society, 2006.
- [24] V. J. LÓPEZ, *An elementary solution to the mountain climbers' problem*, *aequationes mathematicae*, 57 (1999), pp. 45–49.
- [25] A. LUBIW, *Doubly lexical orderings of matrices*, *SIAM Journal on Computing*, 16 (1987), pp. 854–879.
- [26] J. MATTHEWS, *Markov chains for sampling matchings*, PhD thesis, Edinburgh University, 2008.
- [27] Y. OKAMOTO, R. UEHARA, AND T. UNO, *Counting the number of matchings in chordal and chordal bipartite graph classes*, in *Graph-Theoretic Concepts in Computer Science*, vol. 5911 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 296–307.
- [28] I. PAK, *Lectures on discrete and polyhedral geometry*, 2010. <http://www.math.ucla.edu/~pak/geomp018.pdf>
- [29] A. SINCLAIR, *Improved bounds for mixing rates of Markov chains and multicommodity flow*, *Combinatorics Probability and Computing*, 1 (1992), pp. 351–370.
- [30] J. SPINRAD, *Nonredundant 1's in Γ -free matrices*, *SIAM J. Discrete Math.*, 8 (1995), pp. 251–257.
- [31] J. SPINRAD, A. BRANDSTÄDT, AND L. STEWART, *Bipartite permutation graphs*, *Discrete Applied Mathematics*, 18 (1987), pp. 279 – 292.
- [32] J. P. SPINRAD, *Efficient Graph Representations.: The Fields Institute for Research in Mathematical Sciences.*, Fields Institute monographs, American Mathematical Soc., 2003.
- [33] A. TUCKER, *A structure theorem for the consecutive 1's property*, *Journal of Combinatorial Theory, Series B*, 12 (1972), pp. 153 – 162.
- [34] ———, *The parallel climbers puzzle*, *Math Horizons*, (1995), pp. 22–24.
- [35] R. UEHARA, *Linear time algorithms on chordal bipartite and strongly chordal graphs*, in *Automata, Languages and Programming*, 29th International Colloquium, 2002, pp. 993–1004.
- [36] L. G. VALIANT, *The complexity of computing the permanent*, *Theoretical Computer Science*, 8 (1979), pp. 189–201.
- [37] D. B. WEST, *Introduction to Graph Theory*, Prentice Hall, 2nd ed., 2001.
- [38] J. V. WHITTAKER, *A mountain-climbing problem*, *Canad. J. Math.*, 18 (1996), pp. 873–882.
- [39] M. YANNAKAKIS, *Computing the minimum fill-in is NP-complete*, *SIAM J. Alg. Disc. Meth.*, 2 (1981), pp. 77–79.