Article:

Dyer, M orcid.org/0000-0002-2018-0374 and Muller, H orcid.org/0000-0002-1123-1774 (2019) Counting Perfect Matchings and the Switch Chain. SIAM Journal on Discrete Mathematics, 33 (3). pp. 1146-1174. ISSN 0895-4801

https://doi.org/10.1137/18M1172910

COUNTING PERFECT MATCHINGS AND THE SWITCH CHAIN*

MARTIN DYER† AND HAIKO MÜLLER‡

Abstract. We examine the problem of exactly or approximately counting all perfect matchings in hereditary classes of nonbipartite graphs. In particular, we consider the switch Markov chain of Diaconis, Graham and Holmes. We determine the largest hereditary class for which the chain is ergodic, and define a large new hereditary class of graphs for which it is rapidly mixing. We go on to show that the chain has exponential mixing time for a slightly larger class. We also examine the question of ergodicity of the switch chain in an arbitrary graph. Finally, we give exact counting algorithms for three classes.

Key words. Hereditary Graph Classes, Matchings, Approximate Counting, Markov Chains

AMS subject classifications. 68Q25, 68R10, 60J10

1. Introduction.

In [11], we examined (with Jerrum) the problem of counting all perfect matchings in some particular classes of bipartite graphs, inspired by a paper of Diaconis, Graham and Holmes [10] which gave applications to Statistics. That is, we considered the problem of evaluating the *permanent* of the biadjacency matrix. This problem is well understood for general graphs, at least from a computational complexity viewpoint. Exactly counting perfect matchings has long been known to be #P-complete [34], and this remains true even for graphs of maximum degree 3 [8]. The problem is well known to be in FP for planar graphs [23]. For other graph classes, less is known, but #P-completeness is known for chordal and chordal bipartite graphs [29]. In Section 6, we give positive results for three graph classes. Definitions and relationships between the classes we study are given in the Appendix. See also [3] and [9]. The Appendix also gives a convenient summary of results.

Approximate counting of perfect matchings is known to be in randomized polynomial time for bipartite graphs [21], but the complexity remains open for nonbipartite graphs. The algorithm of [21] is remarkable, but complex. It involves repeatedly running a rapidly mixing Markov chain on a sequence of edge-weighted graphs, starting from the complete bipartite graph, and gradually adapting the edge weights until they approximate the target graph. Simpler methods have been proposed, but do not lead to polynomial time approximation algorithms in general.

In [11], we studied a particular Markov chain on perfect matchings in a graph, the *switch chain*, on some *hereditary* graph classes.[1] That is, classes of graphs for which any vertex-induced subgraph of a graph in the class is also in the class. For reasons given in [11], we believe that hereditary classes are the appropriate objects of study in this context. For the switch chain, we asked: for which hereditary classes is the Markov chain ergodic and for which is it rapidly mixing? We provided a precise answer to the ergodicity question and close bounds on the mixing question. In particular, we showed that the mixing time of the switch chain is polynomial for the class of *monotone graphs* [10] (also known as *bipartite permutation graphs* [32] and *proper interval bigraphs* [18]).

In this paper, we extend the analysis of [11] to hereditary classes of nonbipartite graphs. In Section 2 we consider the question of ergodicity, and in Section 3 we consider rapid mixing of the switch chain. In both cases, we introduce corresponding new graph classes, and examine their relationship to known classes. In particular, we introduce a class generalising monotone graphs, quasimonotone graphs, and show that the switch chain mixes rapidly in this class. In a companion paper [13], we show that this class can be recognised in polynomial time. In Section 4, we show that the switch chain can have exponential mixing time in some well known hereditary graph classes.

*The work reported in this paper was partially supported by EPSRC research grants EP/M004953/1 "Randomized algorithms for computer networks" and EP/S016562/1 Sampling in hereditary classes".

†School of Computing, University of Leeds, Leeds LS2 9JT, UK. Email: m.e.dyer@leeds.ac.uk.

‡School of Computing, University of Leeds, Leeds LS2 9JT, UK. Email: h.muller@leeds.ac.uk.

[1] A preliminary version of [11] appeared as [12]. Inter alia, this used a different, and seemingly less successful, approach to proving rapid mixing of the switch chain for monotone graphs.

Some reasons for restricting attention to hereditary classes are given in [11]. However, we might consider the class of *all* graphs on which the switch chain is ergodic. In Section 5, we discuss the question of deciding ergodicity of the switch chain for an arbitrary graph. We give no definitive answer, but give some evidence that polynomial time recognition is unlikely.

Finally, in Section 6, we give positive results for *exactly* counting perfect matchings in some "small" graph classes, namely cographs, graphs with bounded treewidth and complements of chain graphs. See [3, 9], Section 6, and the Appendix for definitions.

1.1. Notation and definitions. Let $\mathbb{N} = \{1, 2, \ldots\}$ denote the natural numbers, and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. If $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$. For a set S, $S^{(2)}$ will denote the set of subsets of V of size exactly 2. For a singleton set, we will generally omit the braces, where there is no ambiguity. Thus, for example $S \cup x$ will mean $S \cup \{x\}$.

Let $G = (V, E)$ be a (simple, undirected) graph with $|V| = n$. More generally, if H is any graph, we denote its vertex set by $V(H)$, and its edge set by $E(H)$. We write an $e \in E$ between v and w in G as $e = vw$, or $e = \{v, w\}$ if the vw notation is ambiguous. The degree of a vertex $v \in V$ will be denoted by $\deg(v)$, and its neighbourhood by $\mathcal{N}(v)$.

The *empty graph* $G = (\varnothing, \varnothing)$ is the unique graph with $n = 0$. We include the empty graph in the class of connected graphs. Also, $G = (V, V^{(2)})$, is the complete graph on n vertices. The *complement* of any graph $G = (V, E)$ is $\overline{G} = (V, V^{(2)} \setminus E)$. We use the notation $G_1 \simeq G_2$ to indicate graph isomorphism.

If $U \subseteq V$, we will write $G[U]$ for the subgraph of G induced by U. Then a class $\mathcal{C}$ of graphs is called *hereditary* if $G[U] \in \mathcal{C}$ for all $G \in \mathcal{C}$ and $U \subseteq V$. For a cycle C in G, we will write $G[C]$ as shorthand for $G[V(C)]$. Definitions of the hereditary graph classes we consider, and relationships between them, are given in the Appendix.

Let L, R be a *bipartition* of V, i.e. $V = L \cup R$, $L \cap R = \varnothing$. Then we will denote the L, R cut-set by $L{:}R = \{vw \in E : v \in L, w \in R\}$. The associated bipartite graph $(L \cup R, L{:}R)$ will be denoted by $G[L{:}R]$. If C is an even cycle in G, then an *alternating bipartition* of C assigns the vertices of C alternately to L and R as the cycle is traversed.

A *matching* M is an independent set of edges in G. That is $M \subseteq E$, and $e \cap e' = \varnothing$ for all $\{e, e'\} \in M^{(2)}$. A *perfect* matching M is such that, for every $v \in V$, $v \in e$ for some $e \in M$. For a perfect matching M to exist, it is clearly necessary, but not sufficient, that n is even. Then $|M| = n/2$. A *near-perfect* matching M' is one with $|M'| = n/2 - 1$. The empty graph has the unique perfect matching $\varnothing$.

A *hole* in a graph G will mean a chordless cycle of length greater than 4, as in e.g. [3, Definition 1.1.4]. Note that the term has been also used to mean a chordless cycle of length at least 4, as in e.g. [35]. An odd hole is a hole with an odd number of edges and vertices, and an even odd hole is a hole with an even number of edges and vertices. More generally, an i-cycle is a cycle with i edges.

1.2. Approximate counting and the switch chain. Sampling a perfect matching almost uniformly at random from a graph $G = (V, E)$ is known to be computationally equivalent to approximately counting all perfect matchings [22]. The approximate counting problem was considered by Jerrum and Sinclair [19], using a Markov chain similar to that suggested by Broder [5]. They showed that their chain has polynomial time convergence if ratio of the number of near-perfect matchings to the number of perfect matchings in the graph is polynomially bounded as a function of n. They called graphs with this property *P-stable*, and it was investigated in [20]. However, many simple classes of graphs fail to have this property (e.g. chain graphs; in the Appendix, we indicate which of the classes we consider are P-stable.) A further difficulty with this algorithm is that the chain will usually produce only a near-perfect matching, and may require many repetitions before it produces a perfect matching.

For any *bipartite* graph, the Jerrum, Sinclair and Vigoda algorithm [21] referred to above gives polynomial time approximate counting of all perfect matchings. This is a major theoretical achievement, though the algorithm seems too complicated to be used in practice. Moreover, the approach does not appear to extend to nonbipartite graphs, since odd cycles are problematic. Indeed, Štefankovič, Vigoda and Wilmes [33] have recently shown that the algorithm of [21] can fail for nonbipartite graphs. They also give positive results for some graph classes, seemingly different from those discussed here.

For this reason, a simpler Markov chain was proposed in [10], which was called the *switch chain* in [11]. This mixes rapidly in cases that the Jerrum-Sinclair chain does not, and vice versa, so the two cannot be compared. For a graph G possessing some perfect matching M_0, the switch chain maintains a perfect matching M_t for each $t \in [t_{\max}]$, whereas the Jerrum-Sinclair chain does not. It may be described as follows.

Switch chain

(1) Set $t \leftarrow 0$, and find any perfect matching M_0 in G.
(2) Choose $v, v' \in V$, uniformly at random. Let $u, u' \in V$ be such that uv, $u'v' \in M_t$.
(3) If $u'v$, $uv' \in E$, set $M_{t+1} \leftarrow \{u'v, uv'\} \cup M_t \setminus \{uv, u'v'\}$.
(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.

A transition of the chain is called a *switch*. This chain is clearly symmetric on the set of perfect matchings, and hence will converge to the uniform distribution on perfect matchings, provided the chain is ergodic. It is clearly aperiodic, since there is delay probability of at least $1/n$ at each step, from choosing $v = v'$ in step (2). For any $v \neq v'$ the transition probability is at most $4/n^2$, since the choice v, v' can also appear as v', v, and the choice of u, u' as u', u, but the transition may fail in step (3).

2. Ergodicity of the switch chain. For a graph G, we define the *transition graph* $\mathcal{G}(G)$ of the switch chain on G as having a vertex for each perfect matching M in G, and an edge between every two perfect matchings M, M' which differ by a single switch. Then we will say G is *ergodic* if $\mathcal{G}(G)$ is connected. Since the switch chain is aperiodic, this corresponds to the usual definition of ergodicity when $\mathcal{G}(G)$ is non-empty. A class $\mathcal{C}$ of graphs will be called ergodic if every $G \in \mathcal{C}$ is ergodic.

As in [11], we say that a graph $G = (V, E)$ is *hereditarily ergodic* if, for every $U \subseteq V$, the induced subgraph $G[U]$ is ergodic. As discussed in [11], this notion is closely related to that of *self-reducibility* (see, for example [19]). A class of graphs $\mathcal{C}$ will be called hereditarily ergodic if every $G \in \mathcal{C}$ is hereditarily ergodic. We characterise the class of all hereditarily ergodic graphs below. This is the largest hereditary subclass of the (non-hereditary) class of ergodic graphs.

If $\mathcal{G}(G)$ is the empty graph, then G is ergodic. If G has a unique perfect matching, G is ergodic, since $\mathcal{G}(G)$ has a single vertex, and so is connected. Otherwise, let X, Y be any two distinct perfect matchings in $G = (V, E)$. Then X is connected to Y in $\mathcal{G}(G)$ if there is a sequence of switches in G which *transforms* X to Y. Since, $X \oplus Y$ is a set of vertex-disjoint alternating even cycles, it suffices to transform X to Y independently in each of these cycles. Thus, since we are dealing with a hereditary class, we must be able to transform X to Y in the graph induced in G by every even cycle. Therefore, it is sufficient to decide whether or not we can transform X to Y when $X \cup Y$ is an alternating Hamilton cycle in G.

Thus, let H: $v_1 \to v_2 \to \ldots \to v_{2r} \to v_1$ be a Hamilton cycle in the graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_{2r}\}$. Let X, Y be the two perfect matchings which form H and suppose, without loss of generality, that $X = \{v_{2i-1}v_{2i} : i \in [r]\}$. Now the first switch in a sequence from X to Y must use a 4-cycle in G with two edges $v_{2i-1}v_{2i}$, $v_{2j-1}v_{2j} \in X$, with $1 \leq i < j \leq r$. The other two edges of the cycle must be either $v_{2i-1}v_{2j}$, $v_{2i}v_{2j-1}$ or $v_{2i-1}v_{2j-1}$, $v_{2i}v_{2j}$. We call the first an *odd* switch, and the second an *even* switch, see Fig. 2.1.

The only switch that can change an edge in X to an edge in Y must have $v_{2i}v_{2j-1} \in Y$, and hence $j = i+1$. We will call this a *boundary* switch. Clearly a boundary switch is an odd switch, see Fig. 2.1.

The edges $v_{2i-1}v_{2i}$, $v_{2j-1}v_{2j}$ divide H into two vertex-disjoint paths $P_1 : v_{2i} \to v_{2j-1}$ and $P_2 : v_{2j} \to v_{2i-1}$, see Fig. 2.1. Thus performing an odd switch on a 4-cycle $(v_{2i-1}, v_{2i}, v_{2j-1}, v_{2j})$ results in two smaller alternating cycles $P_1 \cup v_{2i-1}v_{2j}$ and $P_2 \cup v_{2i}v_{2j-1}$, on which we can use induction, since we are in a hereditary class. However, performing an even switch on a 4-cycle $(v_{2i-1}, v_{2j-1}, v_{2i}, v_{2j})$ simply produces a new Hamilton cycle $H' = X' \cup Y$, where $X' = X \setminus \{v_{2i-1}v_{2i},\, v_{2j-1}v_{2j}\} \cup \{v_{2i-1}v_{2j-1},\, v_{2i}v_{2j}\}$, see Fig. 2.1.

An edge $v_iv_j \in E \setminus C$ is a *chord* of a cycle C. If C is an even cycle, it is an *odd* chord if $j - i = 1 \pmod 2$ and *even* if $j - i = 0 \pmod 2$. Note that $j - i = i - j \pmod 2$, so the definition is independent of the order

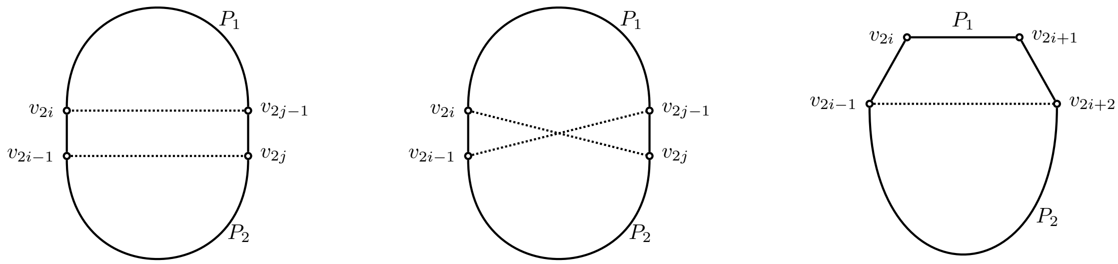


Fig. 2.1: An odd switch, an even switch and a boundary switch

of i and j on C. Note that even and odd chords are not defined for odd cycles.

An odd chord divides an even cycle C into two even cycles, sharing an edge. Thus an odd switch involves two odd chords, and an even switch involves two even chords. However, a 4-cycle with two odd chords may not be an odd switch and a 4-cycle with two even chords may not be an even switch, if the cycle edges involved are not both in X or Y. We call these *illegal* switches, see Fig.2.2.

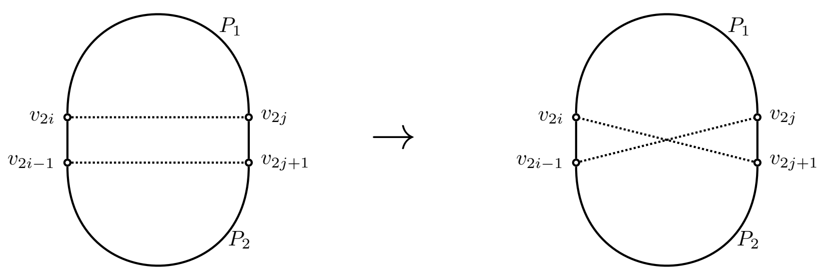


Fig. 2.2: An illegal switch

We define the graph class OddChordal as follows. A graph $G = (V, E)$ is odd chordal if and only if every even cycle C in G of length six or more has an odd chord. Note that this is a hereditary graph property. The switch chain is hereditarily ergodic on the class OddChordal, but it is not the largest class with this property.

Let $(v_{2i-1}, v_{2i}, v_{2j}, v_{2j-1})$ be an even switch for the even cycle C, with cycle segments P_1, P_2 as above. Then a *crossing chord* is an edge (v_k, v_l) such that $v_k \in P_1$, $v_l \in P_2$, see Fig. 2.3.

We can now define our target graph class Switchable. A graph $G = (V, E)$ is switchable if and only if every even cycle C in G of length 6 or more has an odd chord, or has an even switch with a crossing chord. Clearly we may assume that the crossing chord is an even chord. This class is also hereditary, and the definition implies OddChordal $\subseteq$ Switchable.

Our choice of names for the classes OddChordal and Switchable is obvious from the above, and Theorem 2.1 below.

Theorem 2.1. *A graph $G = (V, E)$ is hereditarily ergodic if and only if $G \in$ Switchable.*

Proof. Suppose $G \in$ Switchable and G has a Hamilton cycle H which is the union of two perfect matchings X and Y. We wish to show that X can be transformed to Y using switches in G. We will argue inductively on the size of G. If H is a 4-cycle, we can transform X to Y with a single switch. Suppose then that we can transform X' to Y' for every two perfect matchings X', Y' in any graph $G' \in$ Switchable that has fewer vertices than G. First suppose C has an odd chord (v_i, v_j). Then $H \cup v_i v_j$ forms two even cycles C_1,

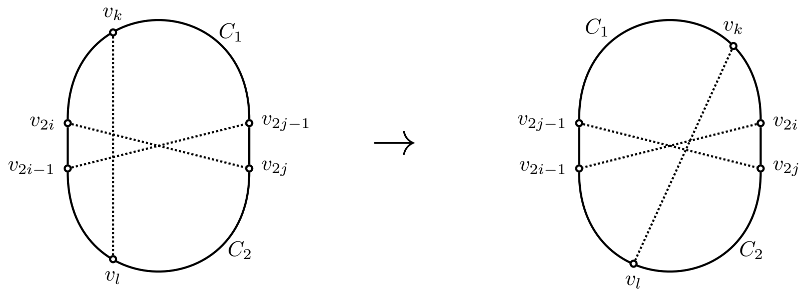


Fig. 2.3: An even switch with a crossing chord

C_2, with $v_i v_j$ as a common edge, so that $v_{i+1} \in C_1$ and $v_{i-1} \in C_2$, see Fig. 2.4. If i is odd and j is even, then $v_i v_{i+1}$, $v_{j-1} v_j \in C_1 \cap X$, and if i is even and j is odd, then $v_{i-1} v_i$, $v_j v_{j+1} \in C_2 \cap X$. In the first case C_1 is an alternating cycle for $X' = X \cap C_1$, $Y' = C_1 \setminus X'$, and in the second C_2 is an alternating cycle for $X' = X \cap C_2$, $Y' = C_2 \setminus X'$. Consider the first case, the second being symmetrical. Then, since C_1 is shorter than H, we can transform X' to Y' by induction. After this, C_2 is an alternating cycle shorter than C, with $X'' = (X \cap C_2) \cup v_i v_j$, $Y'' = Y \cap C_2$, so we can transform X'' to Y'' by induction. This transforms X to Y for the whole cycle H, and we are done.

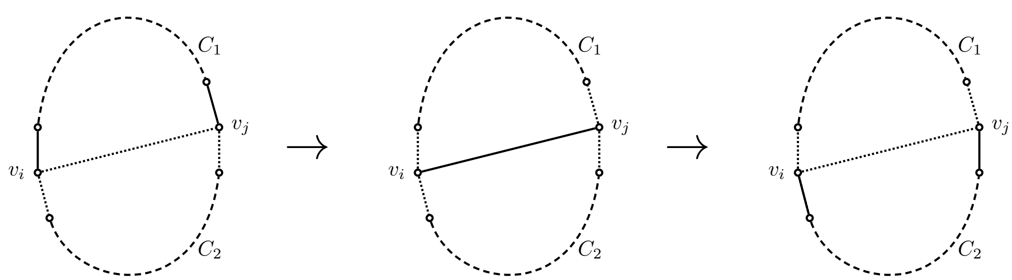


Fig. 2.4: Switching a cycle using an odd chord

Now suppose H has no odd chord, so it has an even switch with an even crossing chord. The even switch gives another Hamilton cycle $H' = P_1 \cup v_{2j-1} v_{2i-1} \cup P_2 \cup v_{2j} v_{2i}$, and suppose its vertices are numbered in the implied order. Now, in this numbering, the parity of vertices in P_1 remains as in C, but the edge $v_{2j-1} v_{2i-1}$ changes the parity of all vertices in P_2. Finally, the edge $v_{2j} v_{2i}$ restores the parity in P_1. Thus, in particular the crossing chord $v_k v_l$ changes from being an even chord in H to an odd chord in H'. (See Fig. 2.3.) Now, since H' has an odd chord, we can use the argument above to show that its matching $X' = X \setminus \{v_{2i-1} v_{2i},\, v_{2i-1} v_{2j}\} \cup \{v_{2i-1} v_{2j-1},\, v_{2i} v_{2j}\}$ can be transformed to Y.

Suppose $G \notin$ Switchable. Then we may assume that there is a Hamilton cycle H_0 in G, of length $2r \geq 6$, which has only even chords. Let X_0, Y be the two perfect matchings such that $H_0 = X_0 \cup Y$. Then H_0 has only even switches, and no even switch can have a crossing chord. Therefore, any switch from X_0 to X_1 in G produces a new Hamilton cycle $H_1 = X_1 \cup Y$ in G. Since there are no crossing chords, the switch does not change the parity of any chord from H_0 to H_1, so H_1 also has only even chords, and hence only even switches. The switch cannot produce a crossing chord for any even switch in H_1, since this chord would also have been crossing in H_0. Thus H_1 is a Hamilton cycle with no odd chord and no even switch with a crossing chord.

Therefore, suppose there is a sequence of switches, $X_0, X_1, \ldots, X_i, \ldots, X_l = Y$. Let s be the smallest i such that $X_i \cap Y \neq \varnothing$. The switch from X_{s-1} to X_s introduces an edge of Y, and so requires a boundary switch in H_{s-1}, which is odd switch. However, by induction, no Hamilton cycle in the sequence $H_0, H_1, \ldots, H_{s-1}$

has an odd chord, and so there can be no odd switch in H_{s-1}. Hence $X_s \cap Y \neq \varnothing$, a contradiction. So the switch chain is not ergodic on G, as required. ∎

Lemma 2.2. *If $G \in$ Switchable, the diameter of $\mathcal{G}(G)$ is at most $(n-3)$.*

Proof. Remember $n = |V(G)|$. Let D_n be the diameter of $\mathcal{G}(G)$. Clearly $D_4 = 1 = 4-3$, and this will be the basis for an induction. Using the construction in the proof of Theorem 2.1, the graph G is decomposable into two smaller graphs G_1 and G_2, which have a common edge, after one switch. Let G_1 and G_2 have $t+1$ and $n-t+1$ vertices, for some t. Thus by induction, $D_n \leq D_{t+1} + D_{n-t+1} + 1 \leq (t-2)+(n-t-2)+1 = n-3$. ∎

For the class OddChordal, we can prove a stronger bound, which also gives a characterisation of the class in terms of the switch chain,

Lemma 2.3. *$G \in$ OddChordal if and only if $\operatorname{diam}(\mathcal{G}(C)) = |C|/2-1$ for every even cycle C in G.*

Proof. Let C be any even cycle in $G \in$ OddChordal. Then C has a boundary switch. First consider the matchings X, Y for which C is an alternating cycle. To switch X to Y, we first perform the boundary switch, leaving an alternating cycle C' with $|C'| = |C|-2$, Assume by induction that $\operatorname{dist}(X,Y) = |C|/2-1$, the base case being for a quadrangle, here $\operatorname{dist}(X,Y) = 1 = 4/2-1$. Then $\operatorname{dist}(X,Y) = 1 + (|C'|/2 - 1) = (|C|-2)/2 = |C|/2-1$, continuing the induction. Thus $\operatorname{diam}(\mathcal{G}(C)) \geq |C|/2-1$. Now, if X, Y are any two matching in $\mathcal{G}(C)$, $X \oplus Y$ can be divided into alternating cycles $C_1, C_2, \ldots, C_k$, say. Then $\operatorname{dist}(X,Y) \leq \sum_{i=1}^{k}(|C_i|/2-1) = |C|/2-k \leq |C|/2-1$, with equality if and only if $X \cup Y = C$. Thus $\operatorname{diam}(\mathcal{G}(C)) = |C|/2-1$.

Conversely, suppose C is an even cycle with no odd chord, but $\mathcal{G}(C) \in$ Switchable, so $\operatorname{diam}(\mathcal{G}(C))$ is well defined. First consider matchings X, Y such that C is an alternating cycle. Then the first switch on the path from X to Y must be an even switch, giving matchings X', Y' which form an alternating cycle C' with $|C'| = |C|$. Now, from above, $\operatorname{dist}(X',Y') \geq |C'|/2-1 = |C|/2-1$, and thus $\operatorname{dist}(X,Y) \geq |C|/2$. Hence $\operatorname{diam}(\mathcal{G}(C)) \neq |C|/2-1$. ∎

Corollary 2.4. *If $G = (V,E) \in$ OddChordal, with $n = |V|$, then $\operatorname{diam}(\mathcal{G}(G)) \leq n/2-1$.*

Proof. If X, Y are any two matching in $\mathcal{G}(G)$, $X \oplus Y$ can be divided into alternating cycles $C_1, C_2, \ldots, C_k$, say. Then $\operatorname{dist}(X,Y) \leq \sum_{i=1}^{k}(|C_i|/2-1) \leq n/2-k \leq n/2-1$. ∎

Finally, we note that, even if the switch chain is not ergodic on a graph G, it may still be able to access an exponential number of perfect matchings from any given perfect matching. Thus the graphs which are not ergodic for the switch chain do not necessarily have "frozen" perfect matchings. Since the existence of frozen states is the most usual criterion for non-ergodicity of large Markov chains, deciding non-hereditary ergodicity seems problematic.

Example 1. *The graph G in Fig. 2.5 has $n = 4k$ vertices, and a Hamilton cycle $H = X \cup Y$, where X, Y are the following two perfect matchings:*

$$X = \big\{\{1,2\},\{3,4\},\ldots,\{4k-1,4k\}\big\}, \quad Y = \big\{\{2,3\},\{4,5\},\ldots,\{4k-2,4k-1\},\{4k,1\}\big\}.$$

From either X or Y, there are k even switches, each without a crossing chord. Each of these switches can be made independently, leading to $2^k = 2^{n/4}$ different matchings. However Y cannot be reached from X, or vice versa. Note that there are $4(k-1)$ illegal switches for H, for example $(1,2,4k-2,4k-1)$ and $(2,3,4k-1,4k)$.

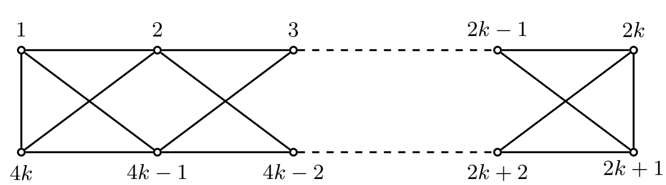
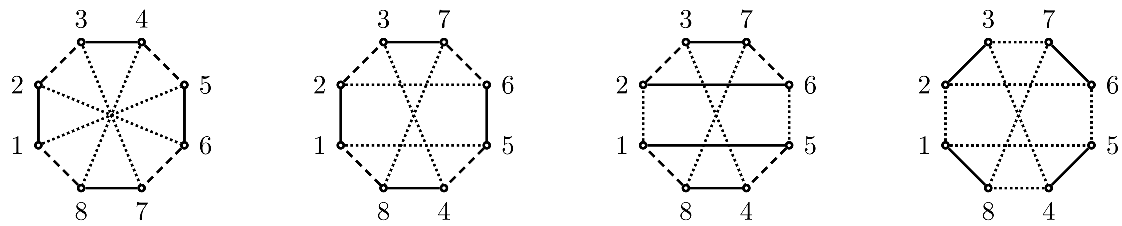


Fig. 2.5: *Example 1*

2.1. Relationship to known graph classes. We will now consider the relationship between the classes defined above and known hereditary graph classes, which are defined in the Appendix. First we will show that OddChordal ⊂ Switchable, by means of the following example.

Example 2. *The graph G in Fig. 2.6 has an (even) Hamilton cycle H is $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 8 \to 1$, and no odd chords, but there is a sequence of switches which transforms the perfect matching $X = \{(1,2),(3,4),(5,6),(7,8)\}$, shown in solid line, to the perfect matching $Y = \{(2,3),(4,5),(6.7),(8,1)\}$, shown dashed. Other edges of G are shown dotted. The switch used to obtain the (solid) perfect matching from its predecessor is shown below each graph. The first switch is an even switch $(3,7,8,4)$ with two crossing even chords $(1,5)$, $(2,6)$. In the last step two disjoint odd switches have been made simultaneously.*



G (3,7,8,4) (1,2,6,5) (2,6,7,3), (1,5,4,8)

Fig. 2.6: *Example* 1.

This example may be extended so that the outer cycle has any even number of vertices. These graphs are the *Möbius ladders*, which appear in [13], in a related context.

Next we consider the simple class Cograph. (See Section 6.1 for definitions and the notation used here.) We may show that Cograph ⊈ Switchable, and hence the switch chain is not necessarily ergodic even in this class. Consider the graph G shown in Fig. 2.7:

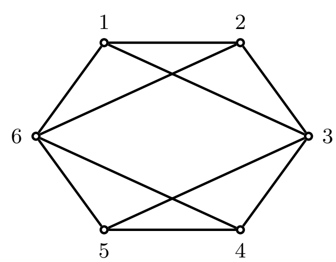


Fig. 2.7: A non-ergodic cograph

This is a cograph, since $G = G_1 \bowtie G_2$, where $G_1 = G[\{1,2,4,5\}]$, $G_2 = G[\{3,6\}]$ are cographs, since $G_1 \simeq (K_1 \bowtie K_1) + (K_1 \bowtie K_1)$ and $G_2 \simeq K_1 \uplus K_1$. However G is not odd chordal, since the 6-cycle $(1,2,3,4,5,6)$ spans G and has only even chords: $\{1,3\}$, $\{2,6\}$, $\{3,5\}$ and $\{4,6\}$. Moreover, G is not switchable, since the two even switches $\{1,3\}$, $\{2,6\}$ and $\{3,5\}$, $\{4,6\}$ have no crossing chord. However, we will show in Section 6, that perfect matchings in a cograph can be counted exactly, and hence a random matching can be generated, in polynomial time.

It is known that Cograph ⊆ Permutation, the class of permutation graphs [9]. Thus we know that Permutation ⊈ Switchable. However, there are permutation graphs which are not cographs and are not switchable. Consider the graph G shown in Fig. 2.8. It has the intersection model shown, so $G \in$ Permutation. However, G is not a cograph, since $G[\{2,3,4,5\}] \simeq P_4$. The 6-cycle $(1,2,4,3,5,6)$ spans G, and has no odd chord or even switch. So, by Theorem 2.1, $G \notin$ Switchable. This example can be extended to an infinite sequence of connected non-ergodic permutation graphs on $2(k+1)$ vertices $(k > 1)$ which are a $2k$-ladder with a triangle at each end. These graphs also appear, in a related context in [13].

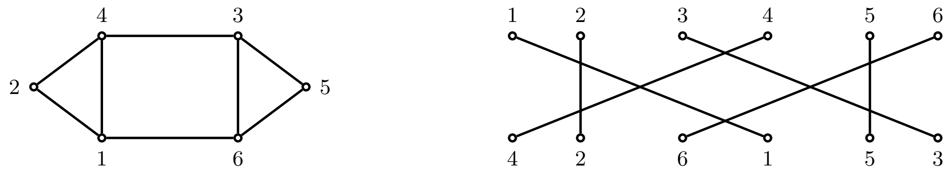


Fig. 2.8: A non-ergodic permutation graph

The class OddChordal does not seem to have been studied previously in the graph theory literature. We have the following relationships to the known classes ChordalBipartite, Interval, Strongly chordal, Chordal, EvenHoleFree and OddChordal. (For definitions, see the Appendix, [9], and Section 2 for OddChordal.)

$$\text{ChordalBipartite} \subset \text{OddChordal} \subset \text{Switchable} \subset \text{EvenHoleFree}$$
$$\text{Interval} \subset \text{StrongChordal} = \text{Chordal} \cap \text{OddChordal}$$
$$\text{ChordalBipartite} = \text{Bipartite} \cap \text{OddChordal}.$$

The inclusions are strict, as illustrated in Example 2 above and Fig. 2.9 below. Fig. 2.9(a) contains a triangle, so cannot be chordal bipartite, but has no odd hole. The only 6-cycle has an odd chord $\{1, 4\}$, so the graph is odd chordal. In Fig. 2.9(b), the 6-cycle has an even chord $\{1, 5\}$, but no odd chord, so the graph has no even hole, but is not odd chordal. In Fig. 2.9(c), the graph is odd chordal, since the only even cycle is the 4-cycle (3,6,7,8), but it has an odd hole (1,2,3,4,5). In Fig. 2.9(d), the outer 6-cycle has no odd chord, but is chordal, since all other chordless cycles are triangles. So, in addition to the inclusions, we see that Chordal and OddHoleFree are incomparable with OddChordal. Thus, from [6], odd chordal graphs are not perfect in general. Note that, except for OddChordal, Switchable and EvenHoleFree, all these classes are

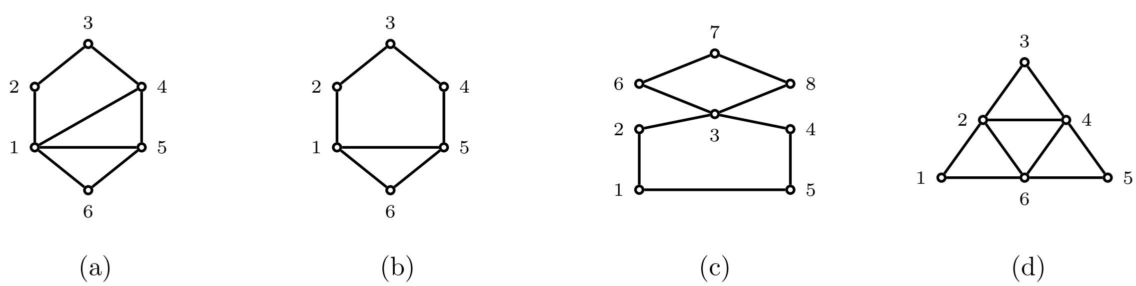


Fig. 2.9: Distinction between graph classes

known to be recognisable in polynomial time. Thus, an obvious question is: does OddChordal have a polynomial time recognition algorithm? We conjecture that the answer is "yes", but currently we cannot prove this.

3. Rapid mixing and quasimonotone graphs. The switch chain for *monotone* graphs (also known as *bipartite permutation graphs*, or *proper interval bigraphs*), was studied in [11]. These are graphs for which the biadjacency matrix has a "staircase" structure. See [11] for precise definitions. The chain was shown to have polynomial mixing time. As far as we are aware, that is the only proof of rapid mixing of the switch chain for a nontrivial class of graphs. Thus, we consider here extending the proof technique of [11] to a much larger class of graphs, which are not necessarily bipartite. To define this class, we need the following definition.

3.1. Quasiclasses. Let $\mathcal{C} \subseteq$ Bipartite, where Bipartite denote the class of bipartite graphs. Then we will define the class quasi-$\mathcal{C}$ as follows. A graph G is in quasi-$\mathcal{C}$ if $G[L{:}R] \in \mathcal{C}$ for all bipartitions L, R of V. This may seem a very demanding definition, but it is not so for most classes of interest, as we shall see.

LEMMA 3.1. *If $\mathcal{C} \subseteq$ BIPARTITE is a hereditary class, then so is quasi-$\mathcal{C}$.*

Proof. Suppose $G = (V, E) \in$ quasi-$\mathcal{C}$ and $v \in V$. We wish to show that $G[V \setminus v] \in$ quasi-$\mathcal{C}$. Let L, R be any bipartition of $V \setminus v$. Then L, R can be extended to a bipartition of $L \cup v, R$ of V. Thus $G[L \cup v{:}R] \in \mathcal{C}$ and, since $\mathcal{C}$ is hereditary, $G[L{:}R] \in \mathcal{C}$. Thus $G[V \setminus v] \in$ quasi-$\mathcal{C}$. ☐

LEMMA 3.2. *Let $\mathcal{C} \subseteq$ BIPARTITE be a graph class that is hereditary and closed under disjoint union, then $\mathcal{C} =$ BIPARTITE $\cap$ quasi-$\mathcal{C}$.*

Proof. First let $G = (L \cup R, E)$ be any bipartite graph that does not belong to $\mathcal{C}$. Since $G = G[L{:}R]$, G does not belong to quasi-$\mathcal{C}$. Hence $\mathcal{C} \supseteq$ BIPARTITE $\cap$ quasi-$\mathcal{C}$.

Next we show $\mathcal{C} \subseteq$ BIPARTITE $\cap$ quasi-$\mathcal{C}$. Let $G = (X \cup Y, E)$ be a graph in $\mathcal{C}$ and let $L{:}R$ be any bipartition of $X \cup Y$. Now $G[L{:}R]$ is the disjoint union of $G_1 = G[(X \cap L) \cup (Y \cap R)]$ and $G_2 = G[(X \cap R) \cup (Y \cap L)]$. The graphs G_1 and G_2 belong to $\mathcal{C}$ since the class is hereditary, and hence $G[L{:}R] \in \mathcal{C}$, because $\mathcal{C}$ is closed under disjoint union. Thus $G \in$ quasi-$\mathcal{C}$. ☐

We also have

LEMMA 3.3. *quasi*-CHORDALBIPARTITE = ODDCHORDAL.

Proof. $G \notin$ ODDCHORDAL if it has an even cycle C with only even chords. Then C is a hole in $G[L{:}R]$, for any bipartition L, R of V which is alternating on C. Thus $G[L{:}R] \notin$ CHORDALBIPARTITE, so $G \notin$ quasi-CHORDALBIPARTITE. Conversely, suppose that $G \notin$ quasi-CHORDALBIPARTITE. Then there is some bipartition L, R of V such that $G[L{:}R]$ contains a hole C. The edges of $G[C]$ that are not in $G[L{:}R]$ must be even chords of C, so C has only even chords in G. Thus $G \notin$ ODDCHORDAL. ☐

In [13], some other examples of quasi--classes are discussed. As a final example here, in Section 3.4 we consider the quasi-class corresponding to the class CHAIN, of *chain graphs*. See [11], [9] or the Appendix for definitions.

Our motivation for introducing this concept is that methods and results for bipartite graph classes may be easily extendible to the corresponding quasi-class. In particular, we are interested in the case of *monotone graphs*.

3.2. Quasimonotone graphs. For the class MONOTONE [11], of monotone graphs, we will denote the hereditary (by Lemma 3.1) class quasi-MONOTONE by QMONOTONE, and a graph $G \in$ QMONOTONE will be called *quasimonotone*. All monotone graphs are quasimonotone, by Lemma 3.2. Since MONOTONE $\subset$ CHORDALBIPARTITE, QMONOTONE $\subset$ ODDCHORDAL, by Lemmas 3.1 and 3.3. So the switch chain is ergodic on quasimonotone graphs, since we have ODDCHORDAL $\subset$ SWITCHABLE.

3.2.1. Unit interval graphs. A *unit interval graph* G (also called a *proper interval graph*, *claw-free interval graph* or *indifference graph*) is the *intersection graph* of a set of unit intervals $v_i = [x_i, x_i + 1]$ ($i \in [n]$) on the real line. That is, $G = (V, E)$, where $V = \{v_i : i \in [n]\}$ and $v_i v_j \in E$ if and only if $i \neq j$ and $v_i \cap v_j \neq \varnothing$. The class of unit interval graphs will be denoted by UNITINTERVAL.

UNITINTERVAL is a hereditary class, with the following forbidden subgraphs: all chordless cycles C_k of length $k \geq 4$, the *claw*, the *3-sun* and its complement, the *net*, as shown in Fig. 3.1. Our interest in this class results from the following.

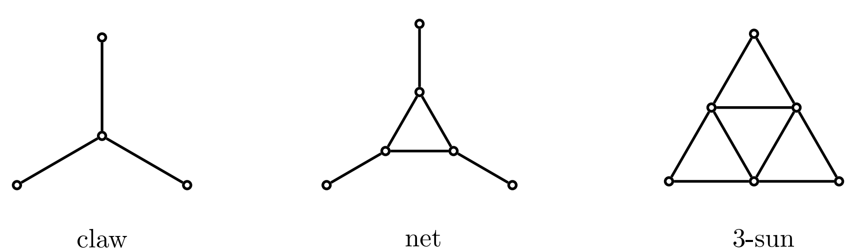
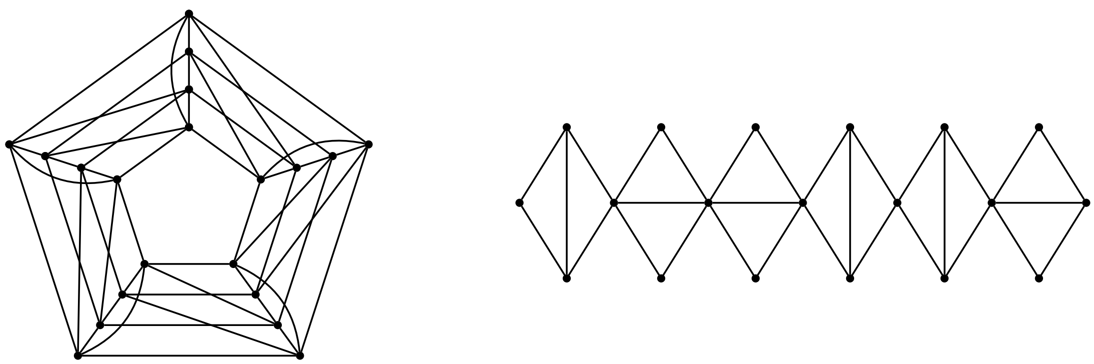


Fig. 3.1: Forbidden subgraphs for unit interval graphs

THEOREM 3.4. UNITINTERVAL $\subset$ QMONOTONE.

Proof. Let $G = (V, E) \in$ UNITINTERVAL, and suppose that L, R is any bipartition of V. Then, by definition, $G[L{:}R]$ is a *unit interval bigraph* [18]. It is shown in [18] that the class of unit interval bigraphs coincides with the class MONOTONE. Thus $G[L{:}R]$ is a monotone graph, and hence G is quasimonotone. ∎

Clearly MONOTONE $\cup$ UNITINTERVAL $\subseteq$ QMONOTONE, but the class is considerably larger than this, and there seems to be no simple characterisation of all graphs in the class. In Fig. 3.2(a), we give an example of a quasimonotone graph which is not monotone (because it is nonbipartite) and not unit interval (because it is not chordal). In Fig. 3.2(b), we give an example of a quasimonotone graph which is chordal (so not monotone) but not unit interval (because it contains claws). We omit the proof that these graphs are quasimonotone. For the recognition algorithm, see [13]. We show in Section 3.3 below that the switch chain is rapidly mixing in the class QMONOTONE. Therefore, the applicability of the switch chain requires a recognition problem for quasimonotone graphs. In particular, can we recognise a quasimonotone graph in polynomial time? Trivially, this problem is only in co-NP, by guessing a bipartition L, R, and using an algorithm for recognising monotone graphs [32] to show that $G[L{:}R]$ is not monotone. However, we show in [13] that the problem of quasimonotone graph recognition is in P.



(a) (b)

Fig. 3.2: Two quasimonotone graphs

3.3. Rapid mixing of the switch chain.

We will now show that the switch chain has polynomial time convergence on the class of quasimonotone graphs. Here we asssume some familiarity with [11].

To do this, we simply extend to quasimonotone graphs the analysis for monotone graphs given in [11, Sec. 3]. We construct a canonical path between any pair of perfect matchings X, Y in G by considering the set of alternating cycles in $X \oplus Y$. Since quasimonotone graphs form a hereditary class, we can reduce the problem to constructing a canonical path for switching each of these cycles, taken in some canonical order. Each such cycle H is an alternating Hamilton cycle in the graph $G' = G[H]$. Note that G' is quasimonotone, by heredity, and has an even number n of vertices, since H is alternating. We will denote the restrictions of X and Y to G' by X' and Y'.

Now consider the alternating bipartition L, R of H, which gives a bipartition of G' such that $|L| = |R| = n/2$. Since G' is quasimonotone, $G'[L{:}R]$ is monotone, and we have $H \subseteq G'[L{:}R]$. Hence we can use the "mountain climbing" technique of [11] to construct a canonical path and an encoding for switching X' to Y' in $G'[L{:}R]$. This is also a canonical path for switching X' to Y' in G', with length $O(n^2)$, as in [11].

The rest of the analysis follows closely that in [11, Sec. 3], noting only that L, R each have at most $n/2$ vertices, rather than n, as in [11]. However the conclusion, that the mixing time is $O(n^7 \log n)$, remains the same. See [11] for further details.

Of course, the starting configuration for the switch chain must be a perfect matching. In the case of monotone graphs, a simple linear time algorithm was given in [11]. This does not extend to quasimonotone graphs, but the $O(n^3)$ algorithm of [26] for general graphs suffices to obtain $O(n^7 \log n)$ mixing time. However, we know that an $O(n^2)$ algorithm exists, by making use of the quasimonotone structure. We will not give details here, since this is not a critical issue. We leave open the question of the existence of a $o(n^2)$ algorithm for quasimonotone graphs.

3.3.1. Forbidden subgraphs of quasimonotone graphs. The forbidden subgraphs for the class Monotone are all (even) holes, together with the three 7-vertex graphs shown in Fig. 3.3, as shown in [25].

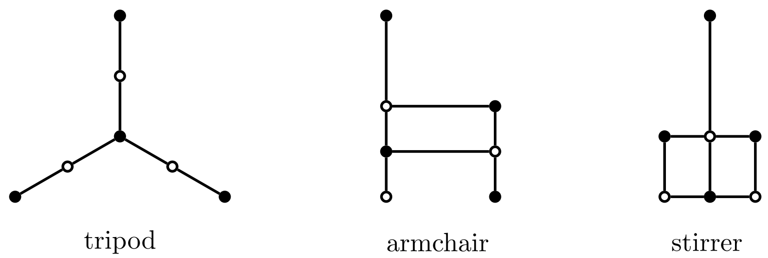


Fig. 3.3: Forbidden subgraphs for monotone graphs

If H is a bipartite graph, a graph H' will be called *pre-H* if it has a bipartition L, R such that $H'[L{:}R] \cong H$. Thus, if a class $\mathcal{C}$ of bipartite graphs can be characterised by the set $\mathcal{F}$ of forbidden subgraphs then quasi-$\mathcal{C}$ can be characterised by forbidding all pre-F ($F \in \mathcal{F}$) as induced subgraphs.

We will call any pretripod, prearmchair or prestirrer, a *flaw*. Referring to the three graphs in Fig. 3.3, these are graphs whose only edges additional to those shown join two white vertices or two black vertices. There are clearly too many of these for us to exhibit them all. A *flawless* graph G will be one which contains no flaw as an induced subgraph. Let us call this (hereditary) class Flawless. Since all flaws have only seven vertices, we can test in $O(n^7)$ time whether an input graph G on n vertices is flawless. Thus membership in Flawless is certainly in P.

However, preholes can have unbounded size. It is easy to see that the preholes are all even cycles that have no odd chord, which is an infinite class. These preholes are clearly the forbidden subgraphs for the class OddChordal. Thus quasimonotone graphs are characterised by the absence of preholes, pretripods, prestirrers and prearmchairs, which is equivalent to the statement QMonotone = Flawless $\cap$ OddChordal.

Unfortunately, this characterisation of QMonotone does not seem to lead to polynomial time recognition. We have observed above that we do not know whether the class OddChordal can be recognised in polynomial time, so we cannot simply test whether G is flawless and odd chordal. However, we show in [13] that quasimonotone graphs can be recognised in polynomial time.

3.4. Quasi-chain graphs. Chain graphs form a subclass of monotone graphs, and there is a trivial algorithm (see [11]) for counting matchings in such graphs. However, this does not extend to the quasi-class. We know of no better analysis of the switch chain, and no better algorithm for either approximately or exactly counting matchings, than those we have given for quasimonotone graphs. However, we will show that there is a simpler recognition algorithm for graphs in this quasi-class than that given in [13].

This class also illustrates a definitional issue. The class of chain graphs, Chain, is not closed under disjoint union, so the quasi-class does not include the class itself. For example, in the simple chain graph of Fig. 3.4, $G[L{:}R]$ is a union of two chain graphs, so quasi-Chain does not contain Chain. Therefore, we define instead the class Chains, which is the class of graphs such that every connected component is a chain graph.

Now quasi-Chains will contain Chains, by Lemma 3.2. Clearly Chains $\subseteq$ Monotone, since each chain graph is monotone, and Monotone is closed under disjoint union. However, it is known that the forbidden subgraph for the class Chain is $2K_2$, but we see from Fig. 3.4 that $2K_2 \in$ Chains. (The graph $2K_2$ comprises two disjoint edges.) So we need a different characterisation of Chains, in which the forbidden subgraphs are connected. This is not difficult. The following result seems to be folklore. It is stated without proof in [4, Proposition 2] and [15, Property 1]. (In fact [4, Proposition 2] does have a two-line proof, but it is a proof

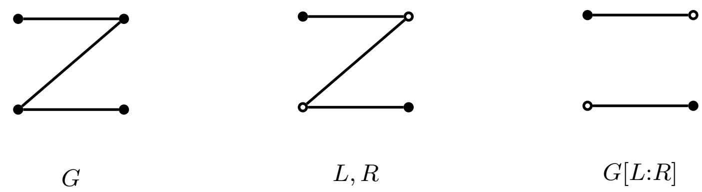


Fig. 3.4

of Corollary 3.6 below.) It seems to be attributed to [1], which does contain related results, but not this. Therefore, we will give a short proof via monotone graphs. The graphs P_5 and C_5 are shown in Fig. 3.6. The graphs P_k, C_k denote, respectively, a path and cycle with k vertices.

Lemma 3.5. *$G \in$ Chains if and only if it is bipartite and P_5-free.*

Proof. If G is not bipartite, it is clearly not in Chains. If it has an induced P_5, this must be entirely in some component G' of G. But P_5 contains an induced $2K_2$, by deleting its middle vertex. So G' cannot be a chain graph, and hence $G \notin$ Chains. Conversely, suppose G is bipartite and P_5-free. It cannot contain a flaw, or a k-hole for $k > 4$, since the flaws contain an induced P_5, and so does every k-hole for any $k \geq 6$. See Fig. 3.5. Thus $G \in$ Monotone.

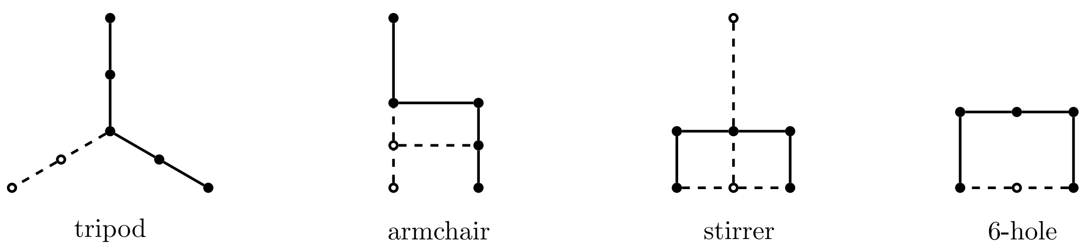


Fig. 3.5

Now consider any connected component G' of G, with monotone biadjacency matrix A'. If G' is not a chain graph, then A' must contain a submatrix of the form below, or its transpose.

	x	y
u	1	0
v	1	1
w	0	1

But this corresponds to an induced path (u, x, v, y, w), which is a P_5, giving a contradiction. Thus G' must be a chain graph, and $G \in$ Chains. □

Corollary 3.6. *$G \in$ Chains if and only if it is (triangle,C_5, P_5)-free.*

Proof. If G is P_5-free, it has no holes of size 6 or more. Therefore, unless it has a triangle or a 5-hole, it must be bipartite. So, if we exclude these two possibilities, Lemma 3.5 implies $G \in$ Chains. The converse is also clear from Lemma 3.5. All graphs in Chains are P_5-free and bipartite, so cannot have a triangle or a 5-cycle. □

Corollary 3.7. *quasi-Chains is precisely the class of (pre-P_5)-free graphs, and membership can be recognised in $O(n^5)$ time.*

Proof. From Lemma 3.5, the forbidden subgraphs for quasi-Chains are pre-P_5's and preholes. But any prehole has size 6 or more, and so induces a pre-P_5. Thus preholes give no new forbidden subgraphs. The pre-P_5's have only 5 vertices, so they can be searched for by brute force in $O(n^5)$ time. □

The class pre-P_5 contains the ten graphs shown in Fig. 3.6, up to isomorphism. They are in order of the number of edges added to P_5, which is given an alternating bipartition.

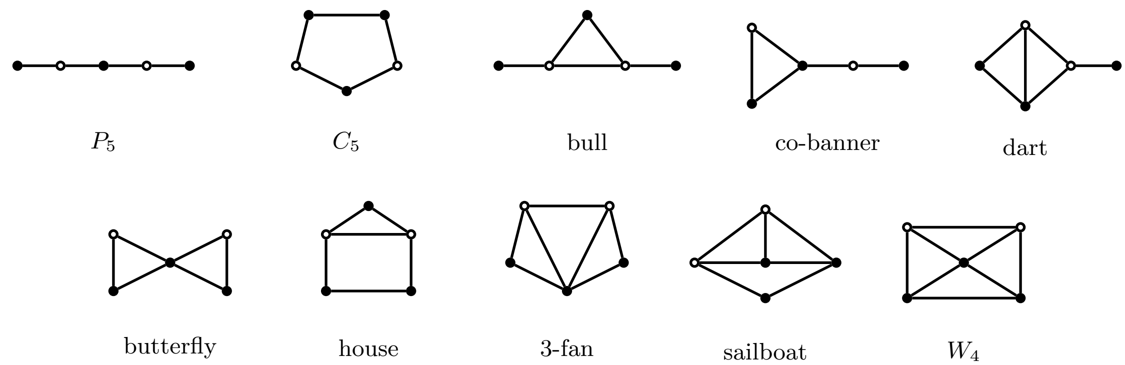


Fig. 3.6: Forbidden subgraphs for quasi-Chains

Lemma 3.8. *Let $\mathcal{C}$ be a hereditary bipartite graph class. Then quasi-$\mathcal{C}$ $\subseteq$ HoleFree if and only if $\mathcal{C} \subseteq$ Chains.*

Proof. If $G \notin$ quasi-HoleFree, then G contains a hole H. Then any bipartition of G which extends an alternating bipartition of H contains P_5 as a subgraph, Thus $G \notin$ quasi-Chains, by Lemma 3.5. Thus quasi-Chains $\subseteq$ HoleFree.

Now suppose $G \notin$ Chains. Then G contains a P_5 by Lemma 3.5. Thus, by heredity, $\mathcal{C}$ contains P_5 and all its subgraphs. Then quasi-$\mathcal{C}$ contains C_5, since every bipartition of C_5 gives P_5 or its subgraphs. Thus quasi-$\mathcal{C} \nsubseteq$ HoleFree. ∎

In particular, we see that if $\mathcal{C} \nsubseteq$ Chains, then quasi-$\mathcal{C} \nsubseteq$ Perfect, by [6].

Thus we have completely settled the ergodicity of the switch chain on hereditary graph classes. Since this has given rise to new classes, the question of efficient recognition of these classes is an interesting open question.

4. Slow mixing of the switch chain. Unfortunately, the switch chain appears to mix slowly in the worst case on graphs in many hereditary classes of interest. In this Section we consider the two classes Interval and Permutation, by showing that even their intersection ChordalPermutation exhibits slow mixing.

4.1. Chordal permutation graphs. The examples we present here are inspired by those given for biconvex graphs in [2, 28].

4.1.1. Construction. For every integer $k \geq 1$ let G_k be the graph with vertex set $U \cup W \cup X \cup Y \cup Z$ and edge set $E_{\mathrm{UW}} \cup E_{\mathrm{W}} \cup E_{\mathrm{WX}} \cup E_{\mathrm{X}} \cup E_{\mathrm{XY}} \cup E_{\mathrm{Y}} \cup E_{\mathrm{YZ}}$ defined by

$$\begin{aligned} U &= \{u_i \mid 1 \leq i \leq k\} & Z &= \{z_i \mid 1 \leq i \leq k\} \\ W &= \{w_i \mid 1 \leq i \leq k\} & Y &= \{y_i \mid 1 \leq i \leq k\} \\ X &= \{x_i \mid 1 \leq i \leq 2\} & E_{\mathrm{X}} &= \{x_1 x_2\} \\ E_{\mathrm{UW}} &= \{u_i w_j \mid 1 \leq i \leq j \leq k\} & E_{\mathrm{YZ}} &= \{z_i y_j \mid 1 \leq i \leq j \leq k\} \\ E_{\mathrm{W}} &= \{vw \mid w \in W, v \in W \setminus \{w\}\} & E_{\mathrm{Y}} &= \{vy \mid y \in Y, v \in Y \setminus \{y\}\} \\ E_{\mathrm{WX}} &= \{vx \mid x \in X, v \in U \cup W\} & E_{\mathrm{XY}} &= \{vx \mid x \in X, v \in Y \cup Z\} \end{aligned}$$

Thus G_k has $n = 4k + 2$ vertices.

Using the notation for cographs, i.e. $\uplus$ for disjoint union and $\bowtie$ for complete join, we have $G_k = (X, E_{\mathrm{X}}) \bowtie \big((U \cup W, E_{\mathrm{UW}} \cup E_{\mathrm{W}}) \uplus (Y \cup Z, E_{\mathrm{YZ}} \cup E_{\mathrm{Y}})\big)$. The graphs $G_k[U \cup W]$ and $G_k[Y \cup Z]$ are threshold graphs, that is, these graphs are both interval and permutation graphs, see Fig. A.1 in the Appendix. Since both these classes are closed under disjoint union and join with complete graphs, G_k too is both an interval graph and a permutation graph. For illustration, Fig. 4.1 gives G_4, where w_2, w_3, y_2, y_3 are not labelled for clarity. Then Fig. 4.2 gives an interval model of G_4, and Fig. 4.3 gives a permutation model.

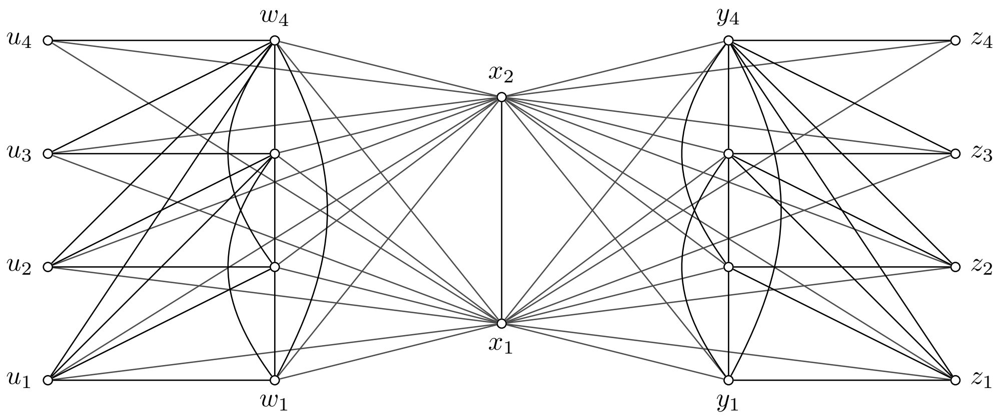


Fig. 4.1: The graph G_4

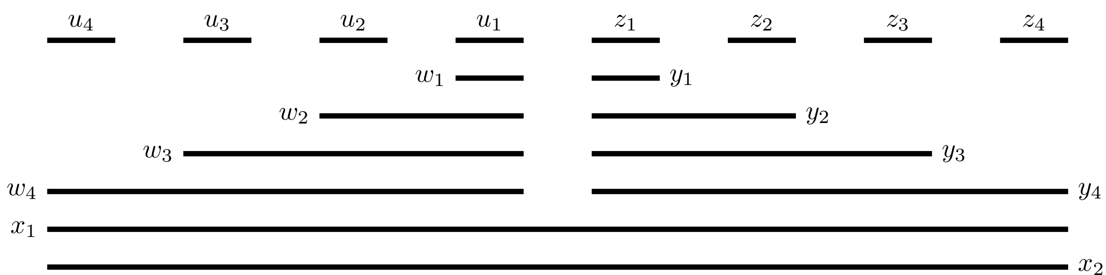


Fig. 4.2: An interval model of G_4

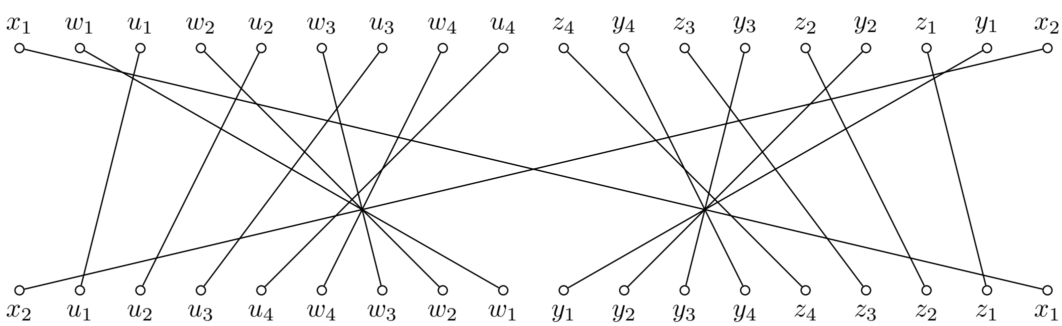


Fig. 4.3: A permutation model of G_4

4.1.2. Perfect matchings of G_k. Now we consider the perfect matchings of G_k. One of them is

$$M_0 = \{u_i w_i, y_i z_i \mid i \in [k]\} \cup \{x_1 x_2\}$$

and plays a special role. If a perfect matching M of G_k contains the edge x_1x_2 then $M = M_0$ because the threshold graphs $G_k[U \cup W]$ and $G_k[Y \cup Z]$ have only one perfect matching, namely $M_0 \cap E_{\mathrm{UW}}$ and $M_0 \cap E_{\mathrm{YZ}}$.

No perfect matching of G_k matches one vertex in X to a vertex in $U \cup W$ and the other to a vertex in $Y \cup Z$, because, for every $v_1 \in U \cup W$ and $v_2 \in Y \cup Z$, the graph $G_k \setminus \{v_1, x_1, v_2, x_2\}$ contains two odd components. For $v_1 \neq w_k$ and $v_2 \neq y_k$ it consists of two connected components that contain $2k-1$ vertices each.

That is, every perfect matching M of G_k either contains the edge x_1x_2 or it contains edges x_1v_1 and x_2v_2 where either $v_1, v_2 \in U \cup W$ or $v_1, v_2 \in Y \cup Z$. We call this the one-sided property of the perfect matchings of G_k.

Let $\mathcal{M}$ be the set of perfect matchings of G_k and let $\mathcal{M}'_1$ and $\mathcal{M}'_2$ be the set of perfect matchings of $G_k[U \cup W \cup X]$ and $G_k[X \cup Y \cup Z]$, respectively. With

$$\mathcal{M}_1 = \{M \cup (M_0 \cap E_{\mathrm{XY}}) \mid M \in \mathcal{M}'_1\}$$
$$\mathcal{M}_2 = \{M \cup (M_0 \cap E_{\mathrm{UW}}) \mid M \in \mathcal{M}'_2\}$$

we have

$$\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2 \qquad\qquad \{M_0\} = \mathcal{M}_1 \cap \mathcal{M}_2$$

by the one-sided property shown above. From $|\mathcal{M}'_i| = 3^k$ for $i = 1, 2$ follows $|\mathcal{M}| = 2 \cdot 3^k - 1$.

4.2. Mixing time. Note that $\mathcal{G}(G_k)$ is connected, but $\mathcal{G}(G_k) \setminus M_0$ is not. By induction we show that every matching $M \in \mathcal{M}$ is at most k switches away from M_0. This is obvious for $M = M_0$. In the inductive step we may assume $M \in \mathcal{M}_1 \setminus M_0$ by symmetry. We consider the maximal index i such that $u_i w_i \notin M$. Let $u_i x$ and $w_i v$ be the two edges in M that saturate u_i and w_i. Since $u_j w_j \in M$ for $i < j \leq k$ we have $x \in X$ and $v \in W \cup X$. Hence vx is an edge of G_k. Switching the 4-cycle (u_i, w_i, v, x) transforms M into a matching containing the edges $u_j w_j$ for all indices j with $i \leq j \leq k$. By induction, the distance from M to M_0 in $\mathcal{G}_k$ is at most k. We may also observe that, $|\mathcal{N}(M_0) \cap \mathcal{M}_1| = k$ in $\mathcal{G}(G_k)$, and $|\mathcal{N}(M_0) \cap \mathcal{M}_2| = k$.

Now, similarly to [2, 28], we upper bound the conductance of the switch chain by computing the flow through the cut $\mathcal{M}_1 \setminus M_0 : \mathcal{M}_2$. There are only k edges in the cut, those from M_0 to $\mathcal{M}_1$, and each has transition probability $2/n^2$. The uniform equilibrium distribution π of the chain gives every state $M \in \mathcal{M}$ probability $\pi(M) = 1/|\mathcal{M}|$, and thus $\pi(\mathcal{M}_1 \setminus M_0) < 1/2$. Thus the flow through the cut is at most $2k/(n^2|\mathcal{M}|) < 1/(8k|\mathcal{M}|)$, and hence the conductance of the chain is

$$\Phi \;\leq\; \frac{2}{8k|\mathcal{M}|} \;=\; \frac{1}{4k(2.3^k - 1)}\,.$$

Now, for example from [27, Thm. 7.3], the mixing time τ_{mix} for the chain to reach variation distance $1/4$ from π satisfies $\tau_{\mathrm{mix}} \geq 1/(4\Phi)$. Thus, for the switch chain on G_n,

$$\tau_{\mathrm{mix}} \;\geq\; \frac{4k(2.3^k - 1)}{4} \;=\; k(2.3^k - 1) \;>\; 3^{k+1} \;>\; 3^{(n+2)/4}\,,$$

for all $k \geq 2$. Thus the mixing time of the switch chain increases exponentially for the graph sequence G_n $(n = 10, 14, 18, \ldots)$.

5. The switch chain in general graphs. We have considered the ergodicity and rapid mixing properties of the switch chain in hereditary classes where all graphs have the relevant property. However, we might ask about recognising the ergodicity, or rapid mixing, of the switch chain for an arbitrary graph. We have seen that there are graphs that are not ergodic, and ergodic graphs that are not rapidly mixing. So we might wish to establish the complexity of recognising ergodicity, or rapid mixing. Since recognising rapid mixing is at least as hard as recognising ergodicity, we will consider only the ergodicity question. Consider the following computational problems, where we measure the complexity of the problem as a function of the graph size n.

Ergodicity
Input: A graph G on n vertices.
Question: Is $\mathcal{G}(G)$ a connected graph?

or the seemingly simpler problem,

Connection
Input: A graph G on n vertices, and two perfect matchings X, Y in G.
Question: Are X, Y in the same component of $\mathcal{G}(G)$?

Connection is easily seen to be in PSPACE. It is the st-connectivity problem on a graph with less than n^n vertices and degrees less than n^2. Then, since st-connectivity is in L (log-space) [30], it follows that

Connection is in PSPACE. Thus **Ergodicity** is also in PSPACE. We simply guess two matchings which are disconnected, and use the **Connection** algorithm to prove disconnection in PSPACE. So **Ergodicity** is in $\mathsf{PSPACE}^{\mathsf{NP}} = \mathsf{PSPACE}$. It is possible that the problem is PSPACE-complete, but we have no evidence for this.

However, it is not clear that either problem is in NP, or even in the polynomial hierarchy, though we suspect that this is the case. We could place **Connection** in NP if we had a polynomial bound on the diameter of $\mathcal{G}(G)$. Then, from the argument above, **Ergodicity** could be solved in co-NP using an oracle for **Connection**, which would place it within the first two levels of the polynomial hierarchy.

Thus we might first ask: what is the maximum diameter of $\mathcal{G}(G)$, over all ergodic graphs G on n vertices? In particular, is this polynomially bounded?

For hereditarily ergodic graphs, we showed, in Lemma 2.2, that $\mathcal{G}(G)$ has diameter $O(n)$. However, this is not true in general. In the following, we show that the diameter of the switch chain can be $\Omega(n^2)$ for a graph on which it is ergodic. Of course, this gives a rather weak lower, rather than an upper, bound on the diameter. But it does show that there is not necessarily a "monotonically improving" path from a matching X to a matching Y in G. And the difficulty of proving even this weak result suggests that establishing a polynomial upper bound will be far from easy.

5.1. The spider's web graph.

Let $\langle j\rangle$ denote j mod 6. The *spider's web* graph W_k is (V_k, E_k),

$$V_0 = \varnothing, \quad U_k = \{u_{kj} : j \in [6]\}, \quad V_k = V_{k-1} \cup U_k \quad (k \geq 1)\,.$$
$$E_1 = \{(u_{11}, u_{14})\} \cup \{(u_{1j}, u_{1,\langle j\rangle+1}) : j \in [6]\}\,,$$
$$E_k = E_{k-1} \cup \{(u_{k-1,j}, u_{kj}), (u_{kj}, u_{k,\langle j\rangle+1}) : j \in [6]\}, \quad (k \geq 1)\,.$$

For example, W_5 is shown in Fig. 5.1. Note that W_k is bipartite, with bipartition $V_{k,0}$, $V_{k,1}$, where $V_{k,p} = \{u_{ij} : i + j = p \bmod 2\}$. We will also define the following subgraphs: the *hexagon* $C_i = W_k[U_i]$ ($i \in [k]$), and the *annulus* $A_i = W_k[U_i \cup U_{i+1}]$ ($i \in [k-1]$). Clearly $C_i \simeq C_1$, for all $i \in [k]$, and $A_i \simeq A_1$, for all $i \in [k-1]$. Also $W_k[V_i] \simeq W_i$ for any $i \in [k]$, so we may refer to this subgraph simply as W_i.

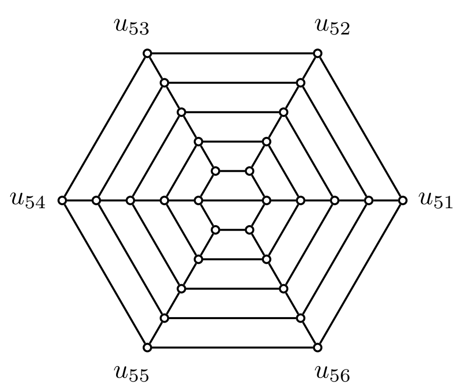


Fig. 5.1: Spider's web W_5

Note that W_k is not hereditarily ergodic for any $k > 1$. This follows from [11, Lem. 2], but note that we have $A_1 \subset W_k$, and the matchings M_1, M_2 in Fig. 5.2 have no switches in A_1.

However, these two matchings are the only obstructions to ergodicity.

Lemma 5.1. *$\mathcal{G}(A_1)$ comprises a connected component, and two isolated vertices, M_1 and M_2.*

Proof. Let M be any matching in A_1. Suppose first that M has a *cross* edge $u_{1j}u_{2j}$ for some $j \in [6]$. The graph $A_1 \setminus \{u_{1j}, u_{2j}\}$, given by deleting u_{1j}, u_{2j}, is hereditarily ergodic, by [11, Lem. 2]. Thus, M is connected to the matching $M_0 = \{u_{1j}u_{2j} : j \in [6]\}$ with all cross edges, as shown in Fig. 5.5. Therefore suppose M has no cross edges, but has a pair of *parallel* edges $\{u_{1j}u_{1,[j]+1},\ u_{2j}u_{2,[j]+1}\}$, for some $j \in [6]$. Switching the quadrangle $(u_{1j}, u_{2j}, u_{2,[j]+1}, u_{1,[j]+1})$ results in a matching M' with a cross edge (u_{1j}, u_{2j}).

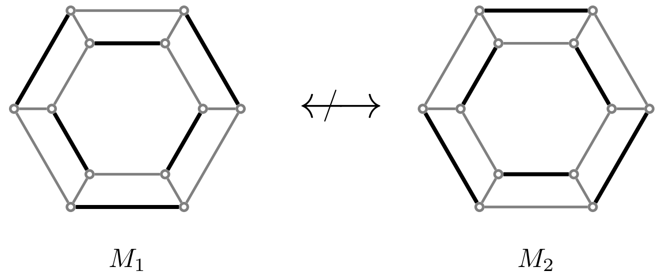


Fig. 5.2: Non-ergodicity of A_1

Hence M is again connected to M_0, via M'. Now any perfect matching in A_1 which has no cross edges or parallel edges is either M_1 or M_2, and these have no available switch. ∎

We will use this to show that W_k is ergodic.

LEMMA 5.2. *$\mathcal{G}(W_k)$ is connected, for all $k \geq 1$.*

Proof. We use induction on k. As basis, W_1 is ergodic: $\mathcal{G}(W_1)$ is shown in Fig. 5.3. For $k > 1$, let X, Y be any two perfect matchings in W_k. From Lemma 5.1, we can exchange $X \cap A_{k-1}$ and $Y \cap A_{k-1}$ to give matchings X_1, Y_1 so that $X_1 \cap A_{k-1}$, $Y_1 \cap A_{k-1}$ have no cross edges. (See Figs. 5.2 and 5.5). By induction, we can exchange $X_1 \cap W_{k-1}$ to $Y_1 \cap W_{k-1}$ to give matchings X_2, Y_2 so that every edge of $X_2 \cap C_{k-1}$ is parallel to an edge of $X_2 \cap C_k$, and every edge of $Y_2 \cap C_{k-1}$ is parallel to an edge of $Y_2 \cap C_k$. Using Lemma 5.1 again, X_2, Y_2 can be transformed to X_3, Y_3, so that $X_3 \cap A_{k-1} = Y_3 \cap A_{k-1}$. Finally, by induction, X_3, Y_3 can be transformed to X_4, Y_4, so that $X_4 \cap W_{k-1} = Y_4 \cap W_{k-1}$ and $X_4 \cap C_k = Y_4 \cap C_k$, and we are done. ∎

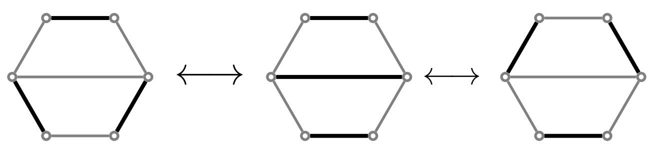


Fig. 5.3: Switching W_1

Let $W_k' = W_k \setminus u_{11}u_{14}$ be the graph W_k after deleting the edge $u_{11}u_{14}$. Then the two perfect matchings M_1, M_2 in W_k' given by

$$M_p = \{w_{i,j}w_{i,\langle j\rangle+1} : i + j = p - 1 \bmod 2,\ i \in [k],\ j \in [6]\} \quad (p = 1, 2),$$

and shown in Fig. 5.4, have no available switch, so are isolated vertices in $\mathcal{G}(W_k')$. Thus W_k' is not ergodic. Given this, we might suppose that M_1 and M_2 are far apart in $\mathcal{G}(W_k)$, and that is the case.

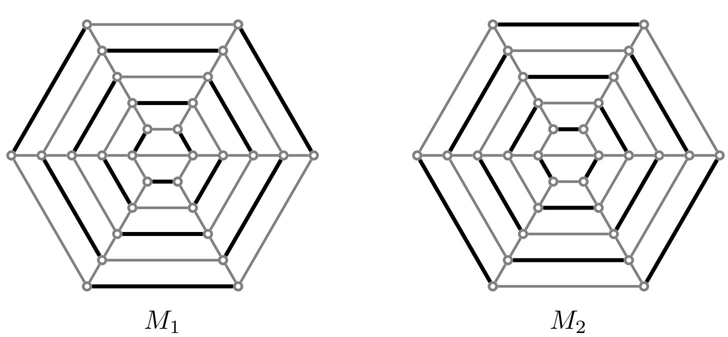


Fig. 5.4: Matchings M_1, M_2 in W_5

LEMMA 5.3. *The distance between M_1, M_2 in $\mathcal{G}(W_k)$ is $k(3k-1)$.*

Proof. Let X_t be the perfect matching at step t on a path P from M_1 to M_2 in $\mathcal{G}(W_k)$, so $X_0 = M_1$ and $X_\ell = M_2$, where ℓ is the path length. For any hexagon C_i ($i \in [k]$), we will write $C_i(t) \supset M_j$ to mean

$C_i \cap X_t = C_i \cap M_j$ $(j = 1, 2)$. Initially $C_i(0) \supset M_1$, for all $i \in [k]$. At step t, we will say C_i has been *exchanged* if $C_i(t) \supset M_2$. Let $s(t) = |\{i \in [k] : C_i(t) \supset M_2\}|$ denote the number of exchanged hexagons, so $s(0) = 0$ and $s(\ell) = k$.

Let t_i be the first step on P at which C_i has been exchanged, and let $t'_i < t_i$ be the last step before any edge of C_i has been switched. Initially, the only switch that can be performed is in W_1, using $u_{11}u_{14}$. After two switches, C_1 can be exchanged (see Fig. 5.3). Since at least two quadrangles must be switched to change the state of six edges, this is clearly the minimum number of switches needed to exchange C_1. Thus $t'_1 = 0$, $t_1 = 2$ and $s(2) = 1$.

For $i > 1$, since M_1, M_2 are edge-disjoint, we can exchange C_i only by switching all six edges. Therefore, since no two edges of C_i share a quadrangle, at least six switches are needed to exchange C_i. Now, an edge of C_i can be switched only if there is a parallel edge in C_{i-1} or C_{i+1}. Since, by assumption $C_i(t'_i), C_{i+1}(t'_i) \supset M_1$, we must have $C_{i-1}(t'_i) \supset M_2$.

Then we have the situation shown in Fig 5.5, and we can perform exactly six switches in A_{i-1} so that $C_i(t_i) \supset M_2$, where $t_i = t'_i + 6$. However, we now have $C_{i-1}(t_i) \supset M_1$, so $s(t_i) = s(t'_i)$. Thus $s(t)$ changes only when C_1 is exchanged. So C_1 must be exchanged at least k times to switch the whole of W_k.

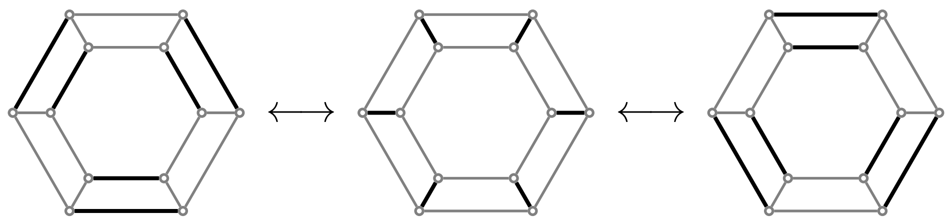


Fig. 5.5: Switching A_i

After switching C_i $(i \in [k-1])$ for the first time, we have $C_i(t_i) \supset M_2$, $C_{i+1}(t_i) \supset M_1$. So we can exchange C_{i+1}, using six switches in A_i. Thus we can propagate the exchanged cycle $C_j(t) \supset M_2$ outwards, starting with $j = 1$, and until $j = i$, leaving $C_j(t_i) \supset M_1$ $(j \in [i-1])$. See Fig. 5.6.

Since C_k must be switched, we continue this outward propagation until $i = k$. Then we have $C_k(t_k) \supset M_2$ and $C_k(t_k) \supset M_1$ $(i \in [k-1])$, after $t_k = 6(k-1)+2 = 6k-4$ switches. This is clearly the minimum number of switches needed to exchange C_k, starting from $X_0 = M_1$.

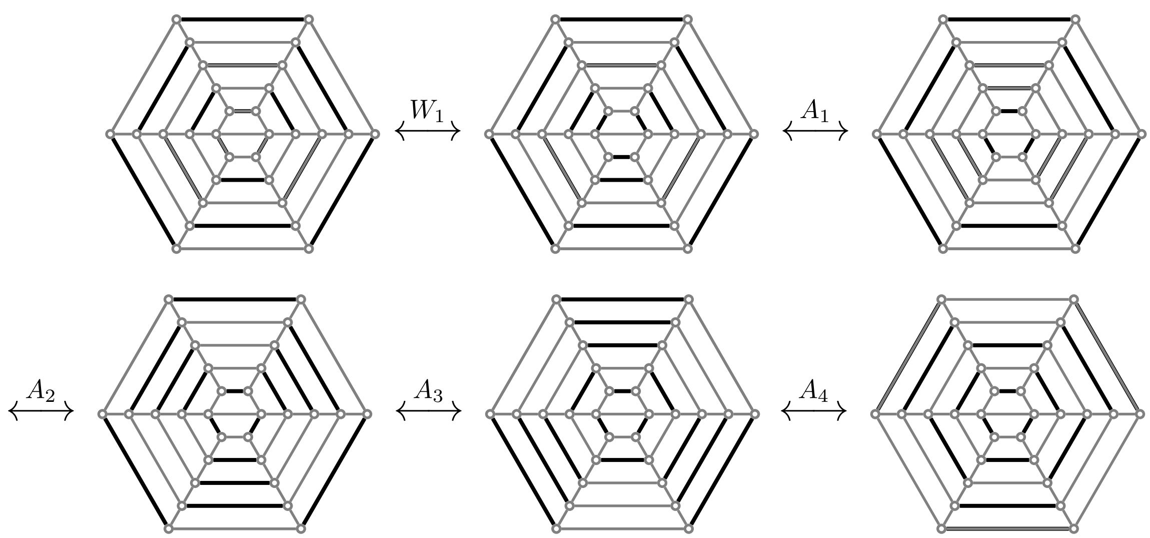


Fig. 5.6: Exchanging C_5

Now, for each $i = k, k-1, \ldots, 2, 1$, suppose we have M_1 in W_i and M_2 in $W_k \setminus W_i$. Then we can exchange C_i in W_i as above, and hence P will terminate with $X_\ell = M_2$. The minimum number of switches needed to exchange C_i in W_i is $t_i = (6i - 4)$. Let $\ell_{\min}$ be the minimum total number of switches required to exchange all C_i ($i \in [k]$). It follows that $\ell_{\min} = \sum_{i=1}^{k} t_i = \sum_{i=1}^{k}(6i-4) = k(3k-1)$. Thus $k(3k-1)$ is the minimum length of any path P in $\mathcal{G}(G)$ from M_1 to M_2. ◻

THEOREM 5.4. *There exists a sequence of graphs G_n, on n vertices, such that the transition graph $\mathcal{G}(G_n)$ is connected, but has diameter $\Omega(n^2)$.*

Proof. For the sequence of graphs W_k ($k = 1, 2, \ldots$), we have $n = 6k$ and, from Lemma 5.3, $\mathcal{G}(W_k)$ has diameter at least $k(3k-1) = n(n-2)/12$. ◻

However, in spite of having no sub-exponential bound on the diameter for general graphs, it not clear that we can even construct graphs for which the diameter is $\Omega(n^3)$.

6. Exact counting. We conclude this paper by considering the problem of exactly counting perfect matchings in some "small" classes of graphs. In fact, we show that all matchings of any fixed size can be counted. For graphs in such classes, there is often an ordering which permits a dynamic programming type of algorithm to be employed. Such an algorithm was given in [11], for example, for monotone graphs of small width. Here we give algorithms for three graph classes defined in the Appendix.

6.1. Cographs. For two graphs $G = (V, E)$ and $H = (W, F)$ with $V \cap W = \varnothing$ we define their *disjoint union* $G \uplus H = (V \cup W, E \cup F)$, and *complete join* $G \bowtie H = (V \cup W, E \cup F \cup \{vw \mid v \in V, w \in W\})$. These two operations are complementary: $\overline{G \bowtie H} = \overline{G} \uplus \overline{H}$ and $\overline{G \uplus H} = \overline{G} \bowtie \overline{H}$.

A graph G is a *cograph* (or *complement-reducible*) if

(a) $G \simeq K_1$, that is, G has one vertex and no edges, or
(b) $G = G_1 \uplus G_2$, where G_1, G_2 are cographs, or
(c) $G = G_1 \bowtie G_2$, where G_1, G_2 are cographs.

The class of cographs was introduced in [7]. In particular, it was shown in [7] that G is a cograph if and only it is P_4-free, where P_4 is the path with four vertices and three edges. Since $P_4 = \overline{P_4}$, this implies that G is a cograph if and only if $\overline{G}$ is a cograph.

The decomposition of a cograph can be represented by a rooted binary tree T, called a *cotree*. The leaves of T are the vertices of G, and its internal nodes are marked $\uplus$ and $\bowtie$, corresponding to constructions (b) and (c) above. Two vertices are adjacent in G if and only if their lowest common ancestor in T is marked $\bowtie$.

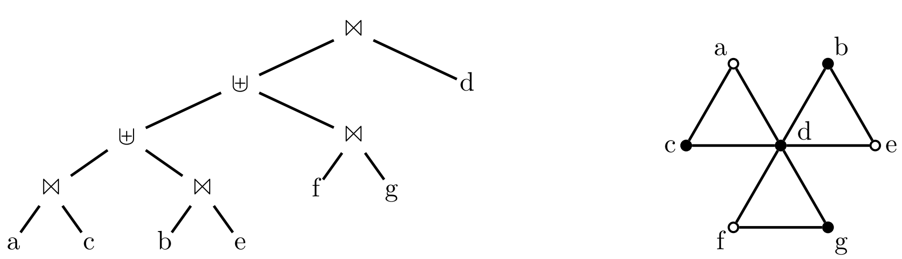


Fig. 6.1: A cotree and the cograph represented with a bipartition into a tripod.

Recurrence equation. For a graph G and an integer s let $m(G, s)$ denote the number of matchings of G that have size exactly s. If G has n vertices then $m(G, s) = 0$ holds for $s < 0$ or $2s > n$. For cographs the values of m can be computed recursively as follows:

leaf For a cograph with one vertex we have $m(K_1, s) = 1$ if $s = 0$ and $m(K_1, s) = 0$ otherwise, because K_1 has only one matching, the empty set.

union For the disjoint union of two graphs G and H we have

$$m(G \uplus H, s) = \sum_{i=0}^{s} m(G, i) \cdot m(H, s-i)$$

because there are no edges between the vertices of G and the vertices of H.

join For $g = |V(G)|$ and $h = |V(H)|$ we have

$$m(G \bowtie H, s) = \sum_{i=0}^{\min(s,\lfloor g/2 \rfloor)} \sum_{j=0}^{\min(s-i,\lfloor h/2 \rfloor)} m(G,i) \cdot m(H,j) \cdot \binom{g-2i}{k} \cdot \binom{h-2j}{k} \cdot k!$$

where $k = s - i - j$. A matching of size s in $G \bowtie H$ partitions into a matching of size i in G, a matching of size j in H and a matching of size k between the vertices of G and H. The subgraph G has $g - 2i$ vertices that are not matched internally, that is, are unsaturated or matched to vertices of H. Similarly, the subgraph H has $h - 2j$ vertices that are not matched internally. From both sets we can choose exactly k vertices to be matched across the join. The partial subgraph isomorphic to $K_{k,k}$ has $k!$ perfect matchings.

Algorithm. Let G be a cograph with n vertices. A cotree T of G can be computed in linear time [17] and has $n - 1$ internal nodes. For every cograph H represented by a rooted subtree of T we can compute $m(H, s)$ for all values of s with $0 \leq s \leq n/2$. This takes time $O(1)$ for leaves, $O(n)$ for union nodes and $O(n^2)$ for join nodes. Hence $m(G, n/2)$, the number of perfect matchings of G, can be computed in time $O(n^4)$.

6.2. Graphs with bounded treewidth. A pair (T, X) is a *tree decomposition* of a graph $G = (V, E)$ if T is a tree with node set I and X maps nodes of T to subsets of V (called *bags*) such that

(a) $\forall v \in V,\ \exists i \in I,\ v \in X(i)$;
(b) $\forall uv \in E,\ \exists i \in I,\ \{u, v\} \subseteq X(i)$;
(c) $\forall v \in V,\ T[\{i \mid v \in X(i)\}]$ is connected.

The *width* of (T, X) is $\max_{i \in I} |X(i)| - 1$ and the *treewidth* of G is the minimum width of a tree decomposition of G. It is denoted as $\text{tw}(G)$. The class of graphs with $\text{tw}(G) \leq w$, for some constant w, is clearly hereditary.

In a *rooted* tree decomposition we choose one node r to become the root of the tree. For all other nodes i, the neighbour of i on the path to r is the *parent* of i, all other neighbours are *children* of i. All neighbours of r are children of the root. For a rooted tree decomposition (T, X) and every $i \in I$ let $Y(i) = X(i) \cup \bigcup_j Y(j)$ where the union is taken over all children of i. Especially we have $Y(i) = X(i)$ for all leaves i of T, and $Y(r) = V$.

A *nice* tree decomposition of $G = (V, E)$ is a rooted tree decomposition (T, X) of G where each node has at most two children, which recursively uses the operations:

start If i is a leaf of T then $X(i) = \varnothing$.
introduce/forget If i has exactly one child j then $X(i)$ and $X(j)$ differ by one vertex. More precisely, i is an *introduce* node if $X(i) \supset X(j)$ and i is a *forget* node if $X(i) \subset X(j)$.
join If i has two children j and k then $X(i) = X(j)$ and $X(i) = X(k)$.
root The root r is a node with $X(r) = \varnothing$, usually a forget node, but a start node if $V = \varnothing$.

Every graph $G = (V, E)$ has a nice tree decomposition of width $\text{tw}(G)$ that contains $O(|V|)$ nodes, see Lemma 13.1.2 on page 149 of [24].

Recurrence equations. Let (T, X) be a nice tree decomposition of a graph $G = (V, E)$. For every node i of T and every set $U \subseteq X(i)$ let $p(i, U)$ denote the number of perfect matchings in the graph $G[Y(i) \setminus U]$ such that every vertex in $X(i) \setminus U$ is matched to a vertex in $Y(i) \setminus X(i)$. That is, a matching containing an edge with both endpoints in $X(i)$ does not contribute to $p(i, U)$ for any U. The numbers $p(i, U)$ can be computed recursively as follows:

start If i is a leaf of T then $p(i, \varnothing) = 1$ since $\varnothing$ is the unique perfect matching of the empty graph.
introduce If i is an introduce node with child j and $v \in X(i) \setminus X(j)$ then $p(i, U) = 0$ and $p(i, U \cup \{v\}) = p(j, U)$ hold for all $U \subseteq X(j)$. By Condition (b) of the definition the new vertex v has no neighbour vertex in $Y(i) \setminus X(i)$.

forget If i is a forget node with child j and $v \in X(j) \setminus X(i)$ then

$$p(i,U) = \sum_{u \in \mathcal{N}(v) \cap X(i) \setminus U} p(j, U \cup \{u,v\})$$

holds for all $U \subseteq X(i)$. The vertex $v \in X(j) \setminus X(i)$ must be matched to a neighbour $u \in X(i)$. We add the edge uv to the matchings of $G[Y(j) \setminus (U \cup \{u,v\})]$. Note that $p(i,U) = 0$ if $\mathcal{N}(v) \cap X(i) \setminus U = \varnothing$.

join If i is a join node with children j and k then

$$p(i,U) = \sum_{\substack{J \cap K = \varnothing \\ J \cup K = X(i) \setminus U}} p(j, U \cup J) \cdot p(k, U \cup K)\,.$$

Condition (c) of the definition implies $(Y(j) \setminus X(j)) \cap (Y(k) \setminus X(k)) = \varnothing$. In $G[Y(i) \setminus U]$ every matching edge with exactly one endpoint in $X(i) \setminus U$ has its other endpoint either in $Y(j) \setminus X(j)$ or in $Y(k) \setminus X(k)$.

Algorithm. The following generalises the algorithm given in [11] for bounded-degree monotone graphs. Let $G = (V,E)$ be a graph with a nice tree decomposition (T,X) rooted at r. By the definition of $p(i,U)$ the graph G has $p(r,\varnothing)$ perfect matchings. This value can be computed recursively by the recurrence equations above. If the width of (T,X) is w then such an algorithm will run in time $O(3^w n)$, where $n = |V|$, by computing "bottom up" from the leaves to the root in the tree T. In the case where (T,X) is a path decomposition, that is, there are no join nodes, the algorithm takes only $O(w 2^w n)$ time.

6.3. Complements of chain graphs. A bipartite graph $G = (V,E)$ with bipartition (X,Y) is a *chain graph* if for every pair of vertices $u,v \in X$ we have $\mathcal{N}(u) \subseteq \mathcal{N}(v)$ or $\mathcal{N}(u) \supseteq \mathcal{N}(v)$. That is, the vertices in X can be linearly ordered such that $\mathcal{N}(x_1) \subseteq \mathcal{N}(x_2) \subseteq \cdots \subseteq \mathcal{N}(x_n)$. It is easy to see that this implies a linear ordering on Y as well such that $\mathcal{N}(y_1) \supseteq \mathcal{N}(y_2) \supseteq \cdots \supseteq \mathcal{N}(y_m)$.

For the sake of completeness, we re-derive a recurrence given in [11] for the number of matchings in a chain graph. For positive integers m and n let $G = (V,E)$ be a chain graph, as defined above, with $V = X_n \cup Y_m$ where $X_n = \{x_i \mid 1 \le i \le n\}$ and $Y_m = \{y_j \mid 1 \le j \le m\}$. Let $M(i,s)$ be the number of matchings of size exactly s in the subgraph G_i of G induced by $X_i \cup Y_m$. We have

$$\begin{aligned}
&M(i,0) = 1 && \text{for } 0 \le i \le n\\
&M(i,s) = 0 && \text{for } 0 \le i < s \le n\\
&M(i,s) = M(i-1,s) + (\deg(x_i) - s + 1) M(i-1,s-1) && \text{for } 1 \le s \le i \le n
\end{aligned}$$

In the last equation, $M(i-1,s)$ counts matchings of size s in G_i with x_i unmatched. The other term counts all matchings of size s in G_i with x_i matched, as follows. Since G_i is a chain graph, each matching of size $(s-1)$ in G_{i-1} can be extended to a matching of size s in G_i, with x_i matched, in exactly $(\deg(x_i) - s + 1)$ ways.

Next we consider complete graphs. Let $p(G)$ denote the number of perfect matchings in G. Then $p(K_{2n+1}) = 0$ and $p(K_{2n}) = (2n)!!$, where $(2n)!! = 2 \cdot 4 \cdots (2n-2)(2n)$, which is $2^n n!$.

Finally let G be the complement of a chain graph with bipartition (X,Y). Then X and Y are cliques of G, and if we remove their edges from G we obtain a chain graph G^{b}. For $|X| = n$ and $|Y| = m$ we have

$$p(G) = \sum_{s=0}^{\min(n,m)} M(n,s) \cdot p(K_{n-s}) \cdot p(K_{m-s})$$

where $M(n,s)$ is the number of matchings of size exactly s in G^{b}. Since we can compute $M(i,s)$ for all values of i and s in $O(n^2)$ time using the recurrence above, $p(G)$ can be computed in this time as well.

[1] G. Bacsó and Z. Tuza, *Dominating cliques in P_5-free graphs*, Periodica Mathematica Hungarica, 21 (1990), pp. 303—308, https://doi.org/10.1007/BF02352694, http://dx.doi.org/10.1007/BF02352694.
[2] O. Blumberg, *Permutations with interval restrictions*, PhD thesis, Stanford University, 2012.
[3] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph classes: a survey*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.
[4] A. Brandstädt and R. Mosca, *On the structure and stability number of P_5- and co-chair-free graphs*, Discrete Applied Mathematics, 132 (2003), p. 47–65.
[5] A. Z. Broder, *How hard is it to marry at random? (On the approximation of the permanent)*, in 18th Annual ACM Symposium on Theory of Computing, Berkeley, California, 28–30 May 1986, pp. 50–58. Erratum in *Proceedings of 20th Annual ACM Symposium on Theory of Computing*, p.551, 1988, Chicago, Illinois, 2–4 May 1988.
[6] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, *The strong perfect graph theorem*, Annals of Mathematics, 164 (2006), pp. 51–229.
[7] D. G. Corneil, H. Lerchs, and L. S. Burlingham, *Complement reducible graphs*, Discrete Applied Mathematics, 3 (1981), pp. 163–174.
[8] P. Dagum and M. Luby, *Approximating the permanent of graphs with large factors*, Theoretical Computer Science, 102 (1992), pp. 283 – 305.
[9] H. N. de Ridder, *Information system on graph classes and their inclusions*, http://www.graphclasses.org/.
[10] 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.
[11] M. Dyer, M. Jerrum, and H. Müller, *On the switch Markov chain for perfect matchings*, Journal of the ACM, 64 (2017), Article 2.
[12] M. Dyer and H. Müller, *Graph classes and the switch Markov chain for matchings*, Annales de la Faculté des sciences de Toulouse, Série 6, 24 (2015), pp. 885–933.
[13] M. Dyer and H. Müller, *Quasimonotone graphs*, http://arxiv.org/abs/1801.06494, 2018.
[14] M. Farber, *Characterizations of strongly chordal graphs*, Discrete Mathematics, 43 (1983), pp. 173–189.
[15] M. U. Gerber, A. Hertz, and D. Schindl, *P_5-free augmenting graphs and the maximum stable set problem*, Discrete Applied Mathematics, 132 (2003), pp. 109–119.
[16] M. C. Golumbic, *Algorithmic graph theory and perfect graphs*, vol. 57 of Annals of Discrete Mathematics, North-Holland, Amsterdam, 2004.
[17] M. Habib and C. Paul, *A simple linear time algorithm for cograph recognition*, Discrete Applied Mathematics, 2005 (145), pp. 183–197.
[18] P. Hell and J. Huang, *Interval bigraphs and circular arc graphs*, Journal of Graph Theory, 46 (2004), pp. 313–327.
[19] M. Jerrum and A. Sinclair, *Approximating the permanent*, SIAM Journal on Computing, 18 (1989), pp. 1149–1178.
[20] M. Jerrum, A. Sinclair, and B. McKay, *When is a graphical sequence stable?*, in Random Graphs, Vol. 2 (Poznán, 1989), Wiley, New York, 1992, p. 101–115.
[21] 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.
[22] M. Jerrum, L. Valiant, and V. Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoretical Computer Science, 43 (1986), pp. 169–188.
[23] P. W. Kasteleyn, *Dimer statistics and phase transitions*, Journal of Mathematical Physics, 4 (1963), pp. 287–293, https://doi.org/http://dx.doi.org/10.1063/1.1703953, http://scitation.aip.org/content/aip/journal/jmp/4/2/10.1063/1.1703953.
[24] T. Kloks, *Treewidth – computations and approximations*, in Lecture Notes in Computer Science, vol. 842, Springer-Verlag, 1994.
[25] E. Köhler, *Graphs without asteroidal triples*, PhD thesis, Technische Universität Berlin, 1999.
[26] E. Lawler, *Combinatorial optimization - networks and matroids*, Holt, Rinehart and Winston, New York, 1976.
[27] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*, American Mathematical Society, 2006.
[28] J. Matthews, *Markov chains for sampling matchings*, PhD thesis, Edinburgh University, 2008.
[29] 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.
[30] O. Reingold, *Undirected connectivity in log-space*, Journal of the ACM, 55 (2008), pp. 17:1–17:24.
[31] E. Scheinerman, *Characterizing intersection classes of graphs*, Discrete Mathematics, 55 (1985), pp.Pages 185-193.
[32] J. Spinrad, A. Brandstädt, and L. Stewart, *Bipartite permutation graphs*, Discrete Applied Mathematics, 18 (1987), pp. 279 – 292, https://doi.org/http://dx.doi.org/10.1016/S0166-218X(87)80003-3, http://www.sciencedirect.com/science/article/pii/S0166218X87800033.
[33] D. Štefankovič., E. Vigoda,and J. Wilmes *On Counting Perfect Matchings in General Graphs*, in Proc. 13th Latin American Theoretical Informatics Symposium (LATIN 2018), Springer LNCS 10807, pp. 873–885, 2018.
[34] L. G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science, 8 (1979), pp. 189–201.
[35] K. Vušković, *Even-hole-free graphs: a survey*, Applicable Analysis and Discrete Mathematics, 4 (2010), pp. 219–240.

Appendix: Containment of graph classes.

The hereditary classes considered above include the following. Some of these graph classes are defined (or can easily be described) by their minimal forbidden induced subgraphs. Thus, for any class of graphs $\mathcal{H}$, a $\mathcal{H}$-*free* graph G is one that has no induced subgraph H' such that $H' \simeq H \in \mathcal{H}$. To that end, we will define the following (non-hereditary) graph classes.

Holes: $\mathbb{H}_n = \{C_i \mid i \in \mathbb{N}, i \geq n\}$, where C_i is a chordless i-cycle.
Antiholes: $\overline{\mathbb{H}}_n = \{\overline{C}_i \mid i \in \mathbb{N}, i \geq n\}$, where $\overline{C}_i$ is the complement of C_i.
Even holes: $\mathbb{E}_n = \{C_{2i} \mid i \in \mathbb{N}, 2i \geq n\}$.
Odd holes: $\mathbb{O}_n = \{C_{2i+1} \mid i \in \mathbb{N}, 2i+1 \geq n\}$.
Suns: $\mathbb{S}_n = \{S_i \mid i \in \mathbb{N}, i \geq n\}$, where S_i is the i-sun, defined as follows. The graph S_i contains a complete graph K_i, together with a new vertex w_e and edges uw_e, vw_e for each edge $e = uv$ of a Hamilton cycle of K_i. The 3-sun is shown in Fig. 3.1.

OddHoleFree *Odd-hole-free graphs* are the $\mathbb{O}_5$-free graphs.

EvenHoleFree *Even-hole-free graphs* are the $\mathbb{E}_6$-free graphs. Note that some papers, *e.g.* [35], define even-hole-free graphs to be $\mathbb{E}_4$-free.

Switchable *Switchable graphs* are defined in Section 2.

WeakChordal *Weakly chordal graphs*, also known as weakly triangulated graphs, are defined by the class of forbidden subgraphs $\mathbb{H}_5 \cup \overline{\mathbb{H}}_5$.

Bipartite *Bipartite graphs* can be coloured by two colours. That is, their vertex set splits into two independent subsets, called colour classes or partite sets. Bipartite graphs are exactly the $\mathbb{O}_3$-free graphs.

Perfect *Perfect graphs* are defined by the absence of graphs in $\mathbb{O}_5 \cup \overline{\mathbb{O}}_5$, from [6].

OddChordal A graph G is *odd-chordal* if every even cycle of length at least six in G has an odd chord.

Chordal A graph G is *chordal* if every cycle of length at least four in G has a chord. That is, chordal graphs are the $\mathbb{H}_4$-free graphs.

ChordalBipartite A bipartite graph G is *chordal bipartite* if every cycle of length at least six in G has a chord. Since every cycle in a bipartite graph is even, and every chord is odd, the class of chordal bipartite graphs is the intersection of the classes of odd chordal and bipartite graphs. This class is characterised by the forbidden set $\mathbb{O}_3 \cup \mathbb{E}_6$. That is, every chordless cycle in a chordal bipartite graph has length four.

TreeWidth w These are the classes of bounded *treewidth*. That is, for every value of k there is a class $\{G \mid \text{tw}(G) \leq k\}$. For $k = 0$ this is all edgeless graphs, for $k = 1$ all forests. For example, the permutation graph in Fig. 2.8 has treewidth 2.

StrongChordal The class of *strongly chordal graphs* is the intersection of the classes of odd chordal and chordal graphs, see [14]. This class is characterised by the forbidden set $\mathbb{H}_4 \cup \mathbb{S}_3$.

Split The vertex set of a *split graph* splits into a clique and an independent set. These are exactly the chordal graphs with chordal complement. The class is characterised by forbidden $2K_2$, C_4 and C_5.

Convex A bipartite graph is *convex* if one of its partite sets can be linearly ordered such that, for each vertex in the other partite set, the neighbours appear consecutively.

Forest An acyclic graph is called *forest*. Each connected component of a forest is a tree. Forests have treewidth at most one. Their minimal forbidden graphs are $\mathbb{H}_3$.

StrongChordalSplit The class of *strongly chordal split* graphs is the intersection of the classes of strongly chordal graphs and split graphs, characterised by the minimal forbidden subgraphs in $\mathbb{S}_3 \cup \{2K_2, C_4, C_5\}$.

Biconvex A bipartite graph is *biconvex* if both its partite sets can be linearly ordered such that all neighbourhoods appear consecutively.

Permutation *Permutation graphs* are the intersection graphs of straight line segments between two parallel lines, where each segment has one endpoint on each line. The ordering of the endpoints defines the characteristic permutation. The intersection model is also called a matching diagram.

Interval *Interval graphs* are the intersection graphs of intervals on the real line.

Monotone The class of *monotone graphs* is the intersection of the classes of bipartite graphs and permutation graphs, see [11].

QMonotone A graph is *quasimonotone* if all its bipartitions are monotone.

ChordalPermutation The class of *chordal permutation graphs* is the intersection of the classes of permutation graphs and interval graphs.

E-Free *E-free* (chordal bipartite) graphs have been characterised in [12]. (“E” is a P_5 with an additional vertex joined (only) to its middle vertex.)

Cograph A graph is complement reducible, or *cograph*, if it has at most one vertex, or is the disjoint union or the complete join of smaller cographs. The class is characterised by the forbidden P_4 [7].

UnitInterval *Unit interval graphs* are the intersection graphs (see, e.g., [31]) of unit-length intervals on the real line. The minimal forbidden graphs for this class are $\mathbb{H}_4 \cup \{K_{1,3}, S_3, \overline{S_3}\}$. (See Fig. 3.1.)

Chain A bipartite graph is a *chain graph* if every pair of vertices in the same partite set has comparable neighbourhoods with respect to set inclusion. This class is characterised by the minimal forbidden subgraphs C_3, $2K_2$ and C_5.

QChains The quasi-class of disjoint unions of chain graphs.

Threshold *Threshold graphs*, characterised by forbidden $2K_2$, C_4 and P_4.

Cochain Complements of chain graphs, characterised by the absence of $3K_1$, C_4 and C_5.

CompleteBipartite Complete bipartite graphs are characterised by the absence of $K_2 + K_1$ (the complement of P_3) and C_3.

Complete Complete graphs are the $2K_1$-free graphs, i.e every pair of vertices is connected by an edge.

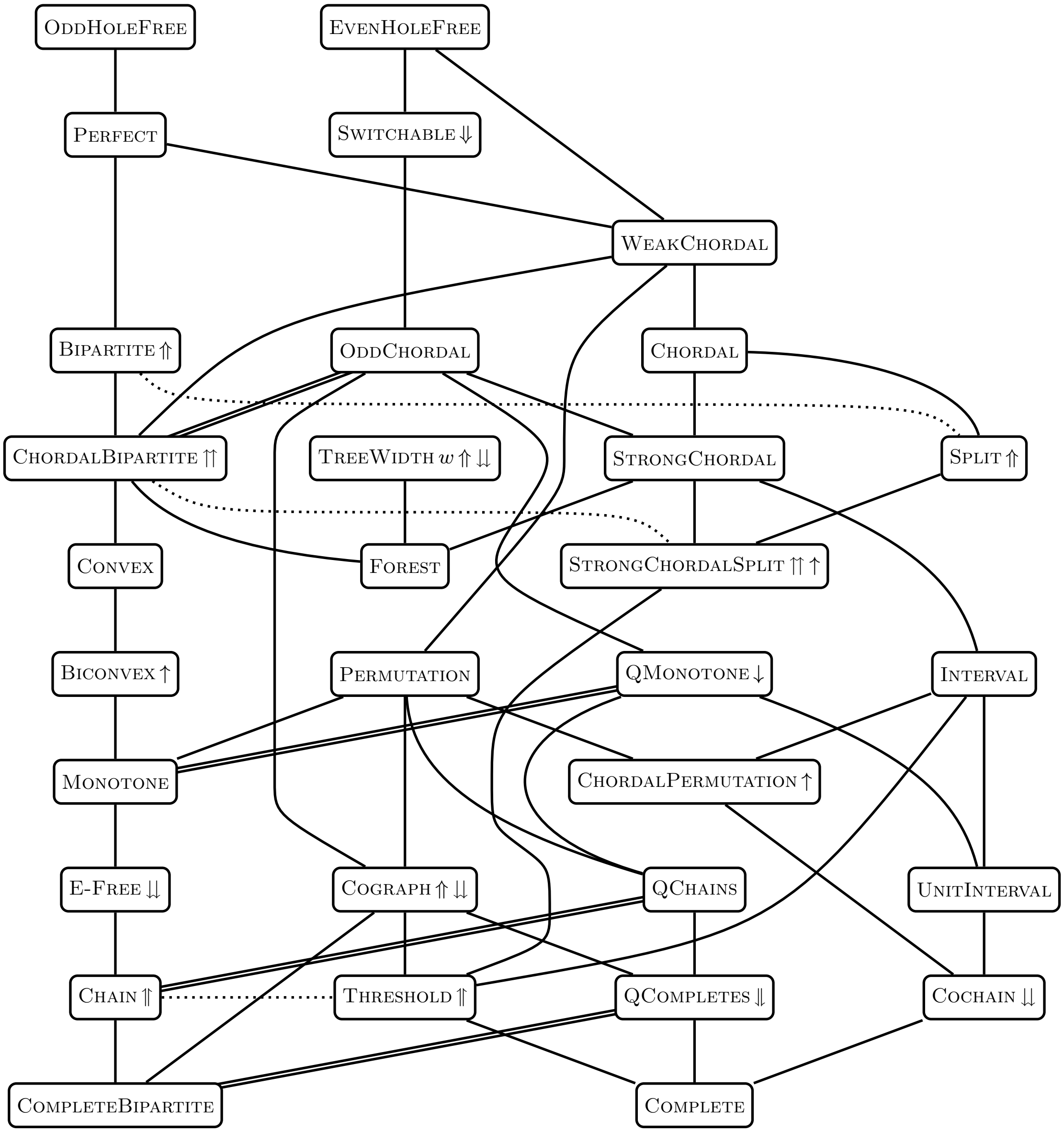
QCompletes Quasi-graphs of disjoint unions of complete bipartite graphs. The class is characterised by the absence of P_4, *paw* (a triangle with pendant edge) and *diamond* (two triangles sharing one edge). Every component of a graph in QCompletes is complete or complete bipartite [13].

The graph classes are partially ordered by inclusion. Fig. A.1 shows a Hasse-diagram of this partial order, restricted to the classes we consider in this paper and some others.

A class $\mathcal{B}$ of bipartite graphs and a class $\mathcal{S}$ of split graphs are *linked* if,

(a) for every G in $\mathcal{B}$, both graphs H obtained from G by completing one of its partite set belongs to $\mathcal{S}$, and
(b) for each graph H in $\mathcal{S}$, the graph G obtained from H by removing all edges between vertices in the clique of the split graph belongs to $\mathcal{B}$.

If the bipartite graph G has partite sets of the same size then the extra edges in H cannot be used by any perfect matching. That is, G and H have exactly the same perfect matchings. If the partite sets of G differ in size then G has no perfect matching. However, H might have a perfect matching if its clique contains more vertices than its independent set. In Fig. A.1 dotted lines indicate linked classes. Double lines indicate the inclusion of a class $\mathcal{C}$ in quasi-$\mathcal{C}^*$ where the graphs in $\mathcal{C}^*$ are disjoint unions of graphs in $\mathcal{C}$. For further information on these classes and references to the original work see [3] or [16].



⇑ This class contains graphs on which the switch chain is not ergodic.
⇓ The switch chain is ergodic on all graphs in this class.
⇈ Counting perfect matchings remains #P-complete when restricted to graphs in this class.
⇊ For all graphs in this class the number of perfect matchings can be computed exactly in polynomial time.
↑ This class contains a sequence of graphs on which the switch chain mixes slowly.
↓ The switch chain mixes rapidly on all graphs in this class.
⥣ This class contains graphs that are not P-stable.
⥥ All graphs in this class are P-stable.

Fig. A.1: Containment of graph classes. Dotted lines indicate linked classes. Double lines indicate the inclusion of a class in the quasi-class of its closure under disjoint union.