



UNIVERSITY OF LEEDS

This is a repository copy of *Counting independent sets in graphs with bounded bipartite pathwidth*.

White Rose Research Online URL for this paper:  
<http://eprints.whiterose.ac.uk/148669/>

Version: Accepted Version

---

**Proceedings Paper:**

Dyer, M, Greenhill, C and Müller, H (2019) Counting independent sets in graphs with bounded bipartite pathwidth. In: Sau, I and Thilikos, D, (eds.) Graph-Theoretic concepts in computer science. Lecture Notes in Computer Science 11789. WG 2019, 19-21 Jun 2019, Vall de Nuria, Catalonia, Spain. Springer Verlag . ISBN 9783030307851

[https://doi.org/10.1007/978-3-030-30786-8\\_23](https://doi.org/10.1007/978-3-030-30786-8_23)

---

© Springer Nature Switzerland AG 2019. This is an author produced version of a paper published in Lecture Notes in Computer Science. Uploaded in accordance with the publisher's self-archiving policy.

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

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



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>

# Counting independent sets in graphs with bounded bipartite pathwidth

Martin Dyer<sup>1</sup>, Catherine Greenhill<sup>2</sup>, and Haiko Müller<sup>1</sup>

<sup>1</sup> School of Computing, University of Leeds, Leeds LS2 9JT, UK.

Research supported by EPSRC grant EP/S016562/1.

{M.E.Dyer|H.Muller}@leeds.ac.uk

<sup>2</sup> School of Mathematics and Statistics, UNSW Sydney, NSW 2052, Australia.

Research supported by Australian Research Council grant DP19010097.

C.Greenhill@unsw.edu.au

**Abstract.** The Glauber dynamics can efficiently sample independent sets almost uniformly at random in polynomial time for graphs in a certain class. The class is determined by boundedness of a new graph parameter called bipartite pathwidth. This result, which we prove for the more general hardcore distribution with fugacity  $\lambda$ , can be viewed as a strong generalisation of Jerrum and Sinclair’s work on approximately counting matchings. The class of graphs with bounded bipartite pathwidth includes line graphs and claw-free graphs, which generalise line graphs. We consider two further generalisations of claw-free graphs and prove that these classes have bounded bipartite pathwidth.

**Keywords:** Markov chain Monte Carlo algorithm · Fully polynomial-time randomized approximation scheme · Independent set · Pathwidth

## 1 Introduction

We will show that we can approximate the number of independent sets in graphs for which all bipartite induced subgraphs are well structured, in a sense that we will define precisely. Our approach is to generalise the Markov chain analysis of Jerrum and Sinclair [19] for the corresponding problem of counting matchings. Their canonical path argument relied on the fact that the symmetric difference of two matchings of a given graph  $G$  is a bipartite subgraph of  $G$  consisting of a disjoint union of paths and even-length cycles. We introduce a new graph parameter, which we call bipartite pathwidth, to enable us to give the strongest generalisation of the approach of [19].

### 1.1 Independent set problems

For a given graph  $G$ , let  $\mathcal{I}(G)$  be the set of all independent sets in  $G$ . The *independence number*  $\alpha(G) = \max\{|I| : I \in \mathcal{I}(G)\}$  is the size of the largest independent set in  $G$ . The problem of finding  $\alpha(G)$  is NP-hard in general, even in various restricted cases, such as degree-bounded graphs. However, polynomial

time algorithms have been constructed for finding a maximum independent set, for various graph classes. The most important case has been *matchings*, which are independent sets in the *line graph*  $L(G)$  of  $G$ . This has been generalised to larger classes of graphs, for example *claw-free* graphs [24], which include line graphs [4], and *fork-free* graphs [1], which include claw-free graphs.

Counting independent sets in graphs is known to be #P-complete in general [26], and in various restricted cases [15, 30]. Exact counting is known only for some restricted graph classes. Even approximate counting is NP-hard in general, and is unlikely to be in polynomial time for bipartite graphs [11].

For some classes of graphs, for example line graphs, approximate counting is known to be possible [19, 20]. The most successful Markov chain approach relies on a close correspondence between approximate counting and sampling uniformly at random [21]. It was applied to degree-bounded graphs in [23] and [12]. In his PhD thesis [22], Matthews used a Markov chain for sampling independent sets in *claw-free* graphs. His chain, and its analysis, generalises that of [19].

Several other approaches to approximate counting have been successfully applied to the independent set problem. Weitz [31] used the *correlation decay* approach on degree-bounded graphs, resulting in an FPTAS for counting independent sets in graphs with degree at most 5. Sly [29] gave a matching NP-hardness result. The correlation decay method was also applied to matchings in [3], and was extended to complex values of  $\lambda$  in [16]. Recently, Efthymiou et al. [14] proved that the Markov chain approach can (almost) produce the best results obtainable by other methods.

The *independence polynomial*  $P_G(\lambda)$  of a graph  $G$  is defined in (1) below. The *Taylor series* approach of Barvinok [2] was used by Patel and Regts [25] to give a FPTAS for  $P_G(\lambda)$  in degree-bounded claw-free graphs. The success of the method depends on the location of the roots of the independence polynomial. Chudnovsky and Seymour [7] proved that all these roots are real, and hence they are all negative. Then the algorithm of [25] is valid for all complex  $\lambda$  which are not real and negative. In this extended abstract (for proofs see [13]), we return to the Markov chain approach.

## 1.2 Preliminaries

We write  $[m] = \{1, 2, \dots, m\}$  for any positive integer  $m$ , and let  $A \oplus B$  denote the symmetric difference of sets  $A, B$ . For graph theoretic definitions not given here, see [10]. Throughout this paper, all graphs are simple and undirected.  $G[S]$  denotes the subgraph of  $G$  induced by the set  $S$  and  $N(v)$  denotes the neighbourhood of vertex  $v$ . Given a graph  $G = (V, E)$ , let  $\mathcal{I}_k(G)$  be the set of independent sets of  $G$  of size  $k$ . The *independence polynomial* of  $G$  is the *partition function*

$$P_G(\lambda) = \sum_{I \in \mathcal{I}(G)} \lambda^{|I|} = \sum_{k=0}^{\alpha(G)} N_k \lambda^k, \quad (1)$$

where  $N_k = |\mathcal{I}_k(G)|$  for  $k = 0, \dots, \alpha$ . Here  $\lambda \in \mathbb{C}$  is called the *fugacity*. We consider only real  $\lambda$  and assume  $\lambda \geq 1/n$  to avoid trivialities. We have  $N_0 = 1$ ,  $N_1 = n$  and  $N_k \leq \binom{n}{k}$  for  $k = 2, \dots, n$ . Thus it follows that for any  $\lambda \geq 0$ ,

$$1 + n\lambda \leq P_G(\lambda) \leq \sum_{k=0}^{\alpha(G)} \binom{n}{k} \lambda^k \leq (1 + \lambda)^n. \quad (2)$$

Note also that  $P_G(0) = 1$  and  $P_G(1) = |\mathcal{I}(G)|$ .

An *almost uniform sampler* for a probability distribution  $\pi$  on a state  $\Omega$  is a randomised algorithm which takes as input a real number  $\delta > 0$  and outputs a sample from a distribution  $\mu$  such that the *total variation distance*  $\frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \pi(x)|$  is at most  $\delta$ . The sampler is a *fully polynomial almost uniform sampler (FPAUS)* if its running time is polynomial in the input size  $n$  and  $\log(1/\delta)$ . The word ‘‘uniform’’ here is historical, as it was first used in the case where  $\pi$  is the uniform distribution. We use it in a more general setting.

If  $w : \Omega \rightarrow \mathbb{R}$  is a *weight function*, then the *Gibbs distribution*  $\pi$  satisfies  $\pi(x) = w(x)/W$  for all  $x \in \Omega$ , where  $W = \sum_{x \in \Omega} w(x)$ . If  $w(x) = 1$  for all  $x \in \Omega$  then  $\pi$  is uniform. For independent sets with  $w(I) = \lambda^{|I|}$ , we have

$$\pi(I) = \lambda^{|I|} / P_G(\lambda), \quad (3)$$

and is often called the hardcore distribution. Jerrum, Valiant and Vazirani [21] showed that approximating  $W$  is equivalent to the existence of an FPAUS for  $\pi$ , provided the problem is *self-reducible*. Counting independent sets in a graph is a self-reducible problem. (2) can be tightened to

$$P_G(\lambda) \leq \sum_{k=0}^{\alpha} \binom{n}{k} \lambda^k \leq \sum_{k=0}^{\alpha} \frac{(n\lambda)^k}{k!} \leq (n\lambda)^\alpha \sum_{k=0}^{\alpha} \frac{1}{k!} \leq e(n\lambda)^\alpha. \quad (4)$$

## 2 Markov chains

### 2.1 Mixing time

For general information on Markov chains and approximate counting see [18].

Consider a Markov chain on state space  $\Omega$  with stationary distribution  $\pi$  and transition matrix  $\mathbf{P}$ . Let  $p_n$  be the distribution of the chain after  $n$  steps. We will assume that  $p_0$  is the distribution which assigns probability 1 to a fixed initial state  $x \in \Omega$ . The *mixing time* of the Markov chain, from initial state  $x \in \Omega$ , is  $\tau_x(\varepsilon) = \min\{n : d_{\text{TV}}(p_n, \pi) \leq \varepsilon\}$ , where  $d_{\text{TV}}(p_n, \pi)$  is the total variation distance between  $p_n$  and  $\pi$ . In the case of the Glauber dynamics for independent sets, the stationary distribution  $\pi$  satisfies (3), and in particular  $\pi(\emptyset)^{-1} = P_G(\lambda)$ . We will always use  $\emptyset$  as our starting state.

Let  $\beta_{\max} = \max\{\beta_1, |\beta_{|\Omega|-1}|\}$ , where  $\beta_1$  is the second-largest eigenvalue and  $\beta_{|\Omega|-1}$  is the smallest eigenvalue of  $\mathbf{P}$ . From [9, Proposition 3] follows  $\tau_x(\varepsilon) \leq$

$(1 - \beta_{\max})^{-1} (\ln(\pi(x)^{-1}) + \ln(1/\varepsilon))$ , see also [28, Proposition 1(i)]. Hence for  $\lambda \geq 1/n$ ,

$$\tau_{\emptyset}(\varepsilon) \leq (1 - \beta_{\max})^{-1} (\alpha(G) \ln(n\lambda)) + 1 + \ln(1/\varepsilon), \quad (5)$$

using (4). We can easily prove that  $(1 + \beta_{|\Omega|-1})^{-1}$  is bounded above by  $\min\{\lambda, n\}$ , see (9). It is more difficult to bound the *relaxation time*  $(1 - \beta_1)^{-1}$ .

## 2.2 Canonical paths method

To bound the mixing time of our Markov chain we will apply the *canonical paths* method of Jerrum and Sinclair [19]. This may be summarised as follows. Let the problem size be  $n$  (in our setting,  $n$  is the number of vertices in the graph  $G$ ,  $\Omega = \mathcal{I}(G)$  and hence  $|\Omega| \leq 2^n$ ). For each pair of states  $X, Y \in \Omega$  we define a path  $\gamma_{XY}$  from  $X$  to  $Y$ , namely  $X = Z_0 \rightarrow Z_2 \rightarrow \dots \rightarrow Z_\ell = Y$  such that successive pairs along the path are given by a transition of the Markov chain. Write  $\ell_{XY} = \ell$  for the length of the path  $\gamma_{XY}$ , and let  $\ell_{\max} = \max_{X,Y} \ell_{XY}$ . We require  $\ell_{\max}$  to be at most polynomial in  $n$ . This is usually easy to achieve, but the set of paths  $\{\gamma_{XY}\}$  must also satisfy the following property.

For any transition  $(Z, Z')$  of the chain there must exist an *encoding*  $W$ , such that, given  $(Z, Z')$  and  $W$ , there are at most  $\nu$  distinct possibilities for  $X$  and  $Y$  such that  $(Z, Z') \in \gamma_{XY}$ . That is, each transition of the chain can lie on at most  $\nu |\Omega^*|$  canonical paths, where  $\Omega^*$  is some set which contains all possible encodings. We usually require  $\nu$  to be polynomial in  $n$ . It is common to refer to the additional information provided by  $\nu$  as “guesses”, and we will do so here. In our situation, all encodings will be independent sets, so we may assume that  $\Omega^* = \Omega$ . The *congestion*  $\varrho$  of the chosen set of paths is given by

$$\varrho = \max_{(Z, Z')} \left\{ \frac{1}{\pi(Z) \mathbf{P}(Z, Z')} \sum_{X, Y: \gamma_{XY} \ni (Z, Z')} \pi(X) \pi(Y) \right\}, \quad (6)$$

where the maximum is taken over all pairs  $(Z, Z')$  with  $\mathbf{P}(Z, Z') > 0$  and  $Z' \neq Z$  (that is, over all transitions of the chain), and the sum is over all paths containing the transition  $(Z, Z')$ . A bound on the relaxation time  $(1 - \beta_1)^{-1}$  will follow from a bound on congestion, using Sinclair’s result [28, Cor. 6]:

$$(1 - \beta_1)^{-1} \leq \ell_{\max} \varrho. \quad (7)$$

## 2.3 Glauber dynamics

The Markov chain we employ will be the *Glauber dynamics*. In fact, we will consider a weighted version of this chain, for a given value of the fugacity (also called activity)  $\lambda > 0$ . Define  $\pi(Z) = \lambda^{|Z|} / P_G(\lambda)$  for all  $Z \in \mathcal{I}(G)$ , where  $P_G(\lambda)$  is the independence polynomial defined in (1). A transition from  $Z \in \mathcal{I}(G)$  to  $Z' \in \mathcal{I}(G)$  will be as follows. Choose a vertex  $v$  of  $G$  uniformly at random.

- If  $v \in Z$  then  $Z' \leftarrow Z \setminus \{v\}$  with probability  $1/(1 + \lambda)$ .
- If  $v \notin Z$  and  $Z \cup \{v\} \in \mathcal{I}(G)$  then  $Z' \leftarrow Z \cup \{v\}$  with probability  $\lambda/(1 + \lambda)$ .

– Otherwise  $Z' \leftarrow Z$ .

This Markov chain is irreducible and aperiodic, and satisfies the detailed balance equations  $\pi(Z) \mathbf{P}(Z, Z') = \pi(Z') \mathbf{P}(Z', Z)$  for all  $Z, Z' \in \mathcal{I}(G)$ . Therefore, the Gibbs distribution  $\pi$  is the stationary distribution of the chain. If  $Z'$  is obtained from  $Z$  by deleting a vertex  $v$  then

$$\mathbf{P}(Z, Z') = \frac{1}{n(1 + \lambda)} \quad \text{and} \quad \mathbf{P}(Z', Z) = \frac{\lambda}{n(1 + \lambda)}. \quad (8)$$

The unweighted version is given by setting  $\lambda = 1$ , and has uniform stationary distribution. Since the analysis for general  $\lambda$  is hardly any more complicated than that for  $\lambda = 1$ , we will work with the weighted case.

It follows from the transition procedure that  $\mathbf{P}(Z, Z) \geq \min\{1, \lambda\}/(1 + \lambda)$  for all states  $Z \in \mathcal{I}(G)$ . That is, every state has a self-loop probability of at least this value. Using a result of Diaconis and Saloff-Coste [8, p. 702], we conclude that the smallest eigenvalue  $\beta_{|\mathcal{I}(G)|-1}$  of  $\mathbf{P}$  satisfies

$$(1 + \beta_{|\mathcal{I}(G)|-1})^{-1} \leq \frac{1 + \lambda}{2 \min\{1, \lambda\}} \leq \min\{\lambda, n\} \quad (9)$$

for  $\lambda \geq 1/n$ . This bound will be dominated by our bound on the relaxation time. We will always use the initial state  $Z_0 = \emptyset$ , since  $\emptyset \in \mathcal{I}(G)$  for any graph  $G$ .

In order to bound the relaxation time  $(1 - \beta_1)^{-1}$  we will use the canonical path method. A key observation is that for any  $X, Y \in \mathcal{I}(G)$ , the induced subgraph  $G[X \oplus Y]$  of  $G$  is bipartite. This can easily be seen by colouring vertices in  $X \setminus Y$  black and vertices in  $Y \setminus X$  white, and observing that no edge in  $G$  can connect vertices of the same colour. To exploit this observation, we introduce the *bipartite pathwidth* of a graph in Section 3. In Section 4 we show how to use the bipartite pathwidth to construct canonical paths for independent sets, and analyse the congestion of this set of paths to prove our main result, Theorem 1.

### 3 Pathwidth and bipartite pathwidth

The *pathwidth* of a graph was defined by Robertson and Seymour [27], and has proved a very useful notion in graph theory [6, 10]. A *path decomposition* of a graph  $G = (V, E)$  is a sequence  $\mathcal{B} = (B_1, B_2, \dots, B_r)$  of subsets of  $V$  such that

1. for every  $v \in V$  there is some  $i \in [r]$  such that  $v \in B_i$ ,
2. for every  $e \in E$  there is some  $i \in [r]$  such that  $e \subseteq B_i$ , and
3. for every  $v \in V$  the set  $\{i \in [r] : v \in B_i\}$  forms an interval in  $[r]$ .

The *width* and *length* of this path decomposition  $\mathcal{B}$  are  $w(\mathcal{B}) = \max\{|B_i| : i \in [r]\} - 1$  and  $\ell(\mathcal{B}) = r$  and the *pathwidth*  $\text{pw}(G)$  of a given graph  $G$  is  $\text{pw}(G) = \min_{\mathcal{B}} w(\mathcal{B})$ , where the minimum taken over all path decompositions  $\mathcal{B}$  of  $G$ . Condition 3 is equivalent to  $B_i \cap B_k \subseteq B_j$  for all  $i, j$  and  $k$  with  $1 \leq i \leq j \leq k \leq r$ . If we refer to a bag with index  $i \notin [r]$  then by default  $B_i = \emptyset$ .

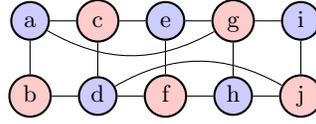


Fig. 1. A bipartite graph

The graph in Fig. 1 has a path decomposition with the following bags:

$$\begin{aligned} B_1 &= \{a, b, d, g\} & B_2 &= \{a, c, d, g\} & B_3 &= \{c, d, g, e\} & B_4 &= \{d, e, f, g\} \\ B_5 &= \{d, f, g, j\} & B_6 &= \{f, g, h, j\} & B_7 &= \{g, h, i, j\} \end{aligned}$$

This path decomposition has length 7 and width 3. If  $P$  is a path,  $C$  is a cycle,  $K_n$  is a complete graph and  $K_{a,b}$  is a complete bipartite graph then

$$\text{pw}(P) = 1, \quad \text{pw}(C) = 2, \quad \text{pw}(K_n) = n - 1, \quad \text{pw}(K_{a,b}) = \min\{a, b\}. \quad (10)$$

The following result will be useful for bounding the pathwidth. The first statement is [5, Lemma 11], while the second appears in [27, equation (1.5)].

**Lemma 1.** *For every subgraph  $H$  of  $G$ ,  $\text{pw}(H) \leq \text{pw}(G)$  holds. If  $W \subseteq V(G)$  then  $\text{pw}(G) \leq \text{pw}(G - W) + |W|$ .*

The *bipartite pathwidth*  $\text{bpw}(G)$  of a graph  $G$  is the maximum pathwidth of an induced subgraph of  $G$  that is bipartite. For any integer  $p \geq 2$ , let  $\mathcal{C}_p$  be the class of graphs  $G$  with  $\text{bpw}(G) \leq p$ . By Lemma 1  $\mathcal{C}_p$  is a hereditary class.

Clearly  $\text{bpw}(G) \leq \text{pw}(G)$ , but the bipartite pathwidth of  $G$  may be much smaller than its pathwidth. A more general example is the class of *unit interval graphs*. These may have cliques of arbitrary size, and hence arbitrary pathwidth. However they are claw-free, so their induced bipartite subgraphs are linear forests, and hence they have bipartite pathwidth at most 1 from Equation 10. The even more general *interval graphs* do not contain a tripod (depicted in Section 5.3), so their bipartite subgraphs are forests of caterpillars, and hence they have bipartite pathwidth at most 2.

**Lemma 2.** *Let  $p$  be a positive integer.*

- (i) *Every graph with at most  $2p + 1$  vertices belongs to  $\mathcal{C}_p$ .*
- (ii) *No element of  $\mathcal{C}_p$  can contain  $K_{p+1, p+1}$  as an induced subgraph.*

A fixed linear order  $<$  on the vertex set  $V$  of a graph  $G$ , extends to subsets of  $V$  as follows: if  $A, B \subseteq V$  then  $A < B$  if and only if (a)  $|A| < |B|$ ; or (b)  $|A| = |B|$  and the smallest element of  $A \oplus B$  belongs to  $A$ . Next, given two path decompositions  $\mathcal{A} = (A_j)_{j=1}^r$  and  $\mathcal{B} = (B_j)_{j=1}^s$  of  $G$ , we say that  $\mathcal{A} < \mathcal{B}$  if and only if (a)  $r < s$ ; or (b)  $r = s$  and  $A_j < B_j$ , where  $j = \min\{i : A_i \neq B_i\}$ .

## 4 Canonical paths for independent sets

Suppose that  $G \in \mathcal{C}_p$ , so that  $\text{bpw}(G) \leq p$ . Take  $X, Y \in \mathcal{I}(G)$  and let  $H_1, \dots, H_t$  be the connected components of  $G[X \oplus Y]$  in lexicographical order. The graph

$G[X \oplus Y]$  is bipartite, so every component  $H_1, \dots, H_t$  is connected and bipartite. We will define a canonical path  $\gamma_{XY}$  from  $X$  to  $Y$  by processing the components  $H_1, \dots, H_t$  in order. Let  $H_a$  be the component of  $G[X \oplus Y]$  which we are currently processing, and suppose that after processing  $H_1, \dots, H_{a-1}$  we have a partial canonical path  $X = Z_0, \dots, Z_N$ . If  $a = 0$  then  $Z_N = Z_0 = X$ . The encoding  $W_N$  for  $Z_N$  is defined by

$$Z_N \oplus W_N = X \oplus Y \quad \text{and} \quad Z_N \cap W_N = X \cap Y. \quad (11)$$

In particular, when  $a = 0$  we have  $W_0 = Y$ . We remark that (11) will not hold during the processing of a component, but always holds immediately after the processing of a component is complete. Because we process components one-by-one, in order, and due to the definition of the encoding  $W_N$ , we have

$$Z_N \cap H_s = Y \cap H_s \text{ for } s = 1, \dots, a-1 \text{ (processed),} \quad (12)$$

$$Z_N \cap H_s = X \cap H_s \text{ for } s = a, \dots, t \text{ (not processed),} \quad (13)$$

$$W_N \cap H_s = X \cap H_s \text{ for } s = 1, \dots, a-1 \text{ (processed),} \quad (14)$$

$$W_N \cap H_s = Y \cap H_s \text{ for } s = a, \dots, t \text{ (not processed).} \quad (15)$$

We now describe how to extend this partial canonical path by processing the component  $H_a$ . Let  $h = |H_a|$ . We will define a sequence  $Z_N, Z_{N+1}, \dots, Z_{N+h}$  of independent sets, and a corresponding sequence  $W_N, W_{N+1}, \dots, W_{N+h}$  of encodings, such that  $Z_\ell \oplus W_\ell \subseteq X \oplus Y$  and  $Z_\ell \cap W_\ell = X \cap Y$  for  $j = N, \dots, N+h$ . Define the set of “remembered vertices”  $R_\ell = (X \oplus Y) \setminus (Z_\ell \oplus W_\ell)$  for  $\ell = N, \dots, N+h$ . By definition, the triple  $(Z, W, R) = (Z_\ell, W_\ell, R_\ell)$  satisfies

$$(Z \oplus W) \cap R = \emptyset \quad \text{and} \quad (Z \oplus W) \cup R = X \oplus Y. \quad (16)$$

This immediately implies that  $|Z_\ell| + |W_\ell| + |R_\ell| = |X| + |Y|$  for  $\ell = N, \dots, N+h$ .

Let  $\mathcal{B} = (B_1, \dots, B_r)$  be the lexicographically-least path decomposition of  $H_a$ . Here we use the ordering on path decompositions defined at the end of Section 3. Since  $G \in \mathcal{C}_p$ , the maximum bag size in  $\mathcal{B}$  is  $d \leq p+1$ .

We process  $H_a$  by processing the bags  $B_1, \dots, B_r$  in order. Initially  $R_N = \emptyset$ , by (11). If bag  $B_i$  is currently being processed and the current independent set is  $Z$  and the current encoding is  $W$ , then

$$(X \cap (B_1 \cup \dots \cup B_{i-1})) \setminus B_i = (W \cap (B_1 \cup \dots \cup B_{i-1})) \setminus B_i, \quad (17)$$

$$(Y \cap (B_1 \cup \dots \cup B_{i-1})) \setminus B_i = (Z \cap (B_1 \cup \dots \cup B_{i-1})) \setminus B_i, \quad (18)$$

$$(X \cap (B_{i+1} \cup \dots \cup B_r)) \setminus B_i = (Z \cap (B_{i+1} \cup \dots \cup B_r)) \setminus B_i, \quad (19)$$

$$(Y \cap (B_{i+1} \cup \dots \cup B_r)) \setminus B_i = (W \cap (B_{i+1} \cup \dots \cup B_r)) \setminus B_i. \quad (20)$$

Let  $Z_\ell, W_\ell, R_\ell$  denote the current independent set, encoding and set of remembered vertices, immediately after the processing of bag  $B_{i-1}$ . We will write  $R_\ell = R_\ell^+ \cup R_\ell^-$  where vertices in  $R_\ell^+$  are added to  $R_\ell$  during the preprocessing phase (and must eventually be inserted into the current independent set), and vertices in  $R_\ell^-$  are added to  $R_\ell$  due to a deletion step (and will go into the encoding during the postprocessing phase). When  $i = 0$  we have  $\ell = N$  and in particular,  $R_N = R_N^+ = R_N^- = \emptyset$ .

**Preprocessing:** We “forget” the vertices of  $B_i \cap B_{i+1} \cap W_\ell$  and add them to  $R_\ell^+$ . This does not change the independent set or add to the canonical path.  
 $R_\ell^+ \leftarrow R_\ell^+ \cup (B_i \cap B_{i+1} \cap W_\ell)$ ;  $W_\ell \leftarrow W_\ell \setminus (B_i \cap B_{i+1})$ ;

**Deletion steps:** for each  $u \in B_i \cap Z_\ell$ , in lexicographical order, **do**  
 $Z_{\ell+1} \leftarrow Z_\ell \setminus \{u\}$ ;  
**if**  $u \notin B_{i+1}$  **then**  $W_{\ell+1} \leftarrow W_\ell \cup \{u\}$ ;  $R_{\ell+1}^- \leftarrow R_\ell^-$   
**else**  $W_{\ell+1} \leftarrow W_\ell$ ;  $R_{\ell+1}^- \leftarrow R_\ell^- \cup \{u\}$ ;  
 $\ell \leftarrow \ell + 1$ ;

**Insertion steps:** for each  $u \in (B_i \cap (W_\ell \cup R_\ell^+)) \setminus B_{i+1}$ , in lexicogr. order, **do**  
 $Z_{\ell+1} \leftarrow Z_\ell \cup \{u\}$ ;  
**if**  $u \in W_\ell$  **then**  $W_{\ell+1} \leftarrow W_\ell \setminus \{u\}$ ;  $R_{\ell+1}^+ \leftarrow R_\ell^+$ ;  
**else**  $W_{\ell+1} \leftarrow W_\ell$ ;  $R_{\ell+1}^+ \leftarrow R_\ell^+ \cup \{u\}$ ;  
 $\ell \leftarrow \ell + 1$ ;

**Postprocessing:** All elements of  $R_{\ell+1}^-$  which do not belong to  $B_{i+1}$  can now be safely added to  $W_\ell$ . This does not change the current independent set or add to the canonical path.  
 $W_\ell \leftarrow W_\ell \cup (R_{\ell+1}^- \setminus B_{i+1})$ ;  $R_\ell^- \leftarrow R_{\ell+1}^- \cap B_{i+1}$ ;

By construction, vertices added to  $R_\ell^+$  are removed from  $W_\ell$ , so the “otherwise” case for insertion is precisely  $u \in R_\ell^+$ .

Observe that both  $Z_\ell$  and  $W_\ell$  are independent sets at every step. This is true initially (when  $\ell = N$ ) and remains true. The preprocessing phases removes all vertices of  $B_i \cap B_{i+1}$  from  $W_\ell$ , which makes room for other vertices to be inserted into the encoding later. A deletion step shrinks the current independent set and adds the removed vertex into  $W_\ell$  or  $R_\ell^-$ . A deleted vertex is only added to  $R_\ell^-$  if it belongs to  $B_i \cap B_{i+1}$ , and so might have a neighbour in  $W_\ell$ . In the insertion steps we add vertices from  $(B_i \cap (W_\ell \cup R_\ell^+)) \setminus B_{i+1}$  to  $Z_\ell$ , now that we have made room. Here  $B_i$  is the last bag which contains the vertex being inserted into the independent set, so any neighbour of this vertex in  $X$  has already been deleted from the current independent set. This phase can only shrink the encoding  $W_\ell$ . Also observe that (16) holds for  $(Z, W, R) = (Z_\ell, W_\ell, R_\ell)$  at every point. Finally, by construction we have  $R_\ell \subseteq B_i$  at all times. Table 1 illustrates this construction for the graph in Fig. 1.

Each step of the canonical path alters the current independent set  $Z_i$  by exactly one element of  $X \oplus Y$ . Every vertex of  $X \setminus Y$  is removed from the current independent set at some point, and is never re-inserted, while every vertex of  $Y \setminus X$  is inserted into the current independent set once, and is never removed. Vertices outside  $X \oplus Y$  are never altered and belong to all or none of the independent sets in the canonical path. Therefore  $\ell_{\max} \leq 2\alpha(G)$ .

**Lemma 3.** *At any transition  $(Z, Z')$  which occurs during the processing of bag  $B_i$ , the set  $R$  of remembered vertices satisfies  $R \subseteq B_i$ , with  $|R| \leq p$  unless  $Z \cap B_i = W \cap B_i = \emptyset$ . In this case  $R = B_i$ , which gives  $|R| = p + 1$ , and  $Z' = Z \cup \{u\}$  for some  $u \in B_i$ .*

**Lemma 4.** *Given a transition  $(Z, Z')$ , the encoding  $W$  of  $Z$  and the set  $R$  of remembered vertices, we can uniquely reconstruct  $(X, Y)$  with  $(Z, Z') \in \gamma_{XY}$ .*

$B_i$	preprocessing after 3rd step	after 1st step	after 2nd step postprocessing
$B_1$			
$B_2$			
$B_3$			
$B_4$			
$B_5$			
$B_6$			
$B_7$			

**Table 1.** The steps of the canonical path, processing each bag in order.

**Theorem 1.** *Let  $G \in \mathcal{C}_p$  be a graph with  $n$  vertices and let  $\lambda \geq 1/n$ , where  $p \geq 2$  is an integer. Then the Glauber dynamics with fugacity  $\lambda$  on  $\mathcal{I}(G)$  (and initial state  $\emptyset$ ) has mixing time*

$$\tau_{\emptyset}(\varepsilon) \leq 2e\alpha(G) n^{p+1} \lambda^p \left(1 + \max(\lambda, 1/\lambda)\right) \left(\alpha(G) \ln(n\lambda) + 1 + \ln(1/\varepsilon)\right).$$

When  $p$  is constant, this upper bound is polynomial in  $n$  and  $\max(\lambda, 1/\lambda)$ .

## 5 Recognisable subclasses of $\mathcal{C}_p$

Theorem 1 shows that the Glauber dynamics for independent sets is rapidly mixing for any graph  $G$  in the class  $\mathcal{C}_p$ , where  $p$  is a fixed positive integer. However, the complexity of recognising membership in the class  $\mathcal{C}_p$  is unknown. Therefore, we consider here three classes of graphs determined by small excluded subgraphs. These classes have polynomial time recognition algorithms. Note that we must always exclude large complete bipartite subgraphs. The three classes are nested. We will obtain better bounds for pathwidth in the smaller classes, and hence better mixing time bounds in Theorem 1.

### 5.1 Claw-free graphs

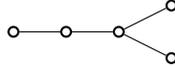
Claw-free graphs exclude the  $K_{1,3}$ , the *claw*. Claw-free graphs form an important superclass of *line graphs* [4], and independent sets in line graphs are *matchings*.

**Lemma 5.** *Let  $G$  be a claw-free graph with independent sets  $X, Y \in \mathcal{I}(G)$ . Then  $G[X \oplus Y]$  is a disjoint union of paths and cycles.*

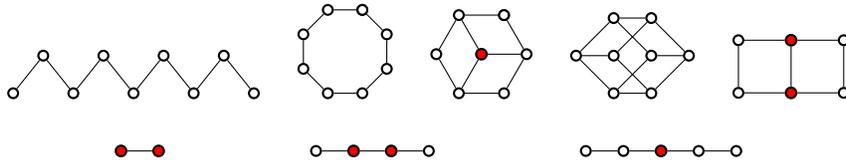
**Lemma 6.** *Claw-free graphs are a proper subclass of  $\mathcal{C}_2$ .*

### 5.2 Graphs with no fork or complete bipartite subgraph

Fork-free graphs exclude the following induced subgraph, the fork:



Two vertices  $u$  and  $v$  are *false twins* if  $N(u) = N(v)$ . In Figure 2, vertices to which false twins can be added are indicated by red colour. Hence each graph containing a red vertex represents an infinite family of augmented graphs.



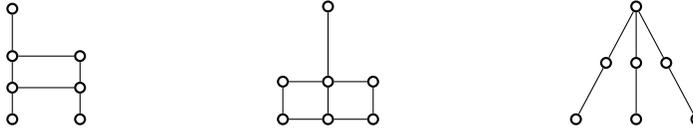
**Fig. 2.** The path  $P_9$ , the cycle  $C_8$ , the augmented bipartite wheel  $BW_3^*$ , the cube  $Q_3$ , an augmented domino, followed by augmented paths  $P_2^*, P_4^*$  and  $P_5^*$ .

**Lemma 7.** *A bipartite graph is fork-free if and only if every connected component is a path, a cycle of even length, a  $BW_3^*$ , a cube  $Q_3$ , or can be obtained from a complete bipartite graph by removing at most two edges that form a matching.*

**Lemma 8.** *For all integers  $d \geq 1$  the fork-free graphs without induced  $K_{d+1, d+1}$  have bipartite pathwidth at most  $\max(4, d + 2)$ .*

### 5.3 Graphs free of armchairs, stirrers and tripods

The graphs depicted below are called *armchair*, *stirrer* and *tripod*. A *fast graph* is a graph that contains none of these three as an induced subgraph.



**Theorem 2.** *For every integer  $d \geq 1$ , a fast bipartite graph that does not contain  $K_{d+1, d+1}$  as an induced subgraph has pathwidth at most  $4d - 1$ .*

## 6 Conclusions and further work

It is clearly NP-hard in general to determine the bipartite pathwidth of a graph, since it is NP-complete to determine the pathwidth of a bipartite graph. However, we need only determine whether  $\text{bpw}(G) \leq d$  for some constant  $d$ . The complexity of this question is less clear. Bodlaender [5] has shown that the question  $\text{pw}(G) \leq d$ , can be answered in  $O(2^{d^2}n)$  time. However, this implies nothing about  $\text{bpw}(G)$ .

In the case of claw-free graphs we can prove stronger sampling results using log-concavity. How far does log-concavity extend in this setting? Does it hold for fork-free graphs? Does some generalisation of log-concavity hold for graphs of bounded bipartite pathwidth? Where log-concavity holds, it allows us to approximate the number of independent sets of a given size. However, there is still the requirement of “amenability” [19]. Jerrum, Sinclair and Vigoda [20] have shown that this can be dispensed with in the case of matchings. Can this be done for claw-free graphs? More ambitiously, can the result of [20] be extended to fork-free graphs and larger classes of graphs of bounded bipartite pathwidth?

An extension would be to consider *bipartite treewidth*,  $\text{btw}(G)$ . Since  $\text{tw}(G) = O(\text{pw}(G) \log n)$  [6, Thm. 66], our results here immediately imply that bounded bipartite treewidth implies *quasipolynomial* mixing time for the Glauber dynamics. Can this be improved to polynomial time?

Finally, can approaches to approximate counting be employed for the independent set problem in particular graph classes? Patel and Regts [25] have used the Taylor expansion approach for claw-free graphs. Could this be extended?

## References

1. V. E. Alekseev, Polynomial algorithm for finding the largest independent sets in graphs without forks, *Discrete Applied Mathematics* **135** (2004), 3–16.
2. A. Barvinok, Computing the partition function of a polynomial on the Boolean cube, in: *A Journey Through Discrete Mathematics*, Springer, 2017, pp. 135–164.
3. M. Bayati, D. Gamarnik, D. Katz, C. Nair and P. Tetali, Simple deterministic approximation algorithms for counting matchings, *Proc. STOC 2007*, 122–127.
4. L. Beineke, Characterizations of derived graphs, *Journal of Combinatorial Theory* **9** (1970), 129–135.
5. H. L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, *SIAM Journal on Computing* **25** (1996), 1305–1317.
6. H. L. Bodlaender, A partial  $k$ -arboretum of graphs with bounded treewidth, *Theoretical Computer Science* **209** (1998), 1–45.
7. M. Chudnovsky and P. Seymour, The roots of the independence polynomial of a clawfree graph, *Journal of Combinatorial Theory (Series B)* **97** (2007), 350–357.
8. P. Diaconis and L. Saloff-Coste, Comparison theorems for reversible Markov chains. *Annals of Applied Probability* 3:696–730, 1993.
9. P. Diaconis and D. Stroock, Geometric bounds for eigenvalues of Markov chains. *Annals of Applied Probability*, 1:36–61, 1991.
10. R. Diestel, *Graph Theory*, Graduate Texts in Mathematics **173**, Springer, 2012.

11. M. Dyer, L. A. Goldberg, C. Greenhill and M. Jerrum, On the relative complexity of approximate counting problems, *Algorithmica* **38** (2003) 471–500.
12. M. Dyer and C. Greenhill, On Markov chains for independent sets, *Journal of Algorithms* **35** (2000), 17–49.
13. M. Dyer, C. Greenhill and H. Müller, Counting independent sets in graphs with bounded bipartite pathwidth. Preprint: [arXiv:1812.03195](https://arxiv.org/abs/1812.03195), 2018.
14. C. Efthymiou, T. Hayes, D. Stefankovic, E. Vigoda and Y. Yin, Convergence of MCMC and loopy BP in the tree uniqueness region for the hard-core model, in *Proc. FOCS 2016*, IEEE, 2016, 704–713.
15. C. Greenhill, The complexity of counting colourings and independent sets in sparse graphs and hypergraphs, *Computational Complexity* **9** (2000), 52–72.
16. N. J. A. Harvey, P. Srivastava and J. Vondrák, Computing the independence polynomial: from the tree threshold down to the roots, in *Proc. SODA 2018*, 1557–1576
17. M. Jerrum, Mathematical foundations of the Markov chain Monte Carlo method, in *Probabilistic Methods for Algorithmic Discrete Math.*, Springer, 1998, 116–165.
18. M. Jerrum, *Counting, Sampling and Integrating: Algorithms and Complexity*, Lectures in Mathematics – ETH Zürich, Birkhäuser, Basel, 2003.
19. M. Jerrum and A. Sinclair, Approximating the permanent, *SIAM Journal on Computing* **18** (1989), 1149–1178.
20. 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), 671–697.
21. M. R. Jerrum, L. G. Valiant and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, *Theoretical Computer Science* **43** (1986), 169–188.
22. J. Matthews, Markov chains for sampling matchings, PhD Thesis, University of Edinburgh, 2008.
23. M. Luby and E. Vigoda, Approximately counting up to four, in *Proc. STOC 1995*, ACM, 1995, 150–159.
24. G. J. Minty, On maximal independent sets of vertices in claw-free graphs, *Journal of Combinatorial Theory, Series B*, **28** (1980), 284–304,
25. V. Patel and G. Regts, Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials, *SIAM Journal on Computing* **46** (2017), 1893–1919.
26. J. S. Provan and M. O. Ball, The complexity of counting cuts and of computing the probability that a graph is connected, *SIAM Journal on Comp.* **12** (1983) 777–788.
27. N. Robertson and P. D. Seymour, Graph minors I: Excluding a forest, *Journal of Combinatorial Theory, Series B*, **35** (1983), 39–61.
28. A. Sinclair, Improved bounds for mixing rates of Markov chains and multicommodity flow, *Combinatorics, Probability and Computing* **1** (1992), 351–370.
29. A. Sly, Computational transition at the uniqueness threshold, in *Proc. FOCS 2010*, IEEE, 2010, 287–296.
30. S. P. Vadhan, The complexity of counting in sparse, regular, and planar graphs, *SIAM Journal on Computing* **31** (2001), 398–427.
31. D. Weitz, Counting independent sets up to the tree threshold, in *Proc. STOC 2006*, ACM, 2006, 140–149.