Counting Independent Sets in Graphs with Bounded Bipartite Pathwidth

Martin Dyer[1], Catherine Greenhill[2], and Haiko Müller[1(✉)]

[1] School of Computing, University of Leeds, Leeds LS2 9JT, UK
{M.E.Dyer,H.Muller}@leeds.ac.uk
[2] School of Mathematics and Statistics, UNSW, Sydney 2052, Australia
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 λ, 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

M. Dyer and H. Müller—Research supported by EPSRC grant EP/S016562/1.
C. Greenhill—Research supported by Australian Research Council grant DP19010097.

I. Sau and D. M. Thilikos (Eds.): WG 2019, LNCS 11789, pp. 298–310, 2019.
https://doi.org/10.1007/978-3-030-30786-8_23

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 λ 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 λ 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 $[m] = \{1, 2, \ldots, m\}$ 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*

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

where $N_k = |\mathcal{I}_k(G)|$ for $k = 0, \ldots, \alpha$. Here $\lambda \in \mathbb{C}$ is called the *fugacity*. We consider only real λ 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 $k = 2, \ldots, n$. Thus it follows that for any $\lambda \geq 0$,

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

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

An *almost uniform sampler* for a probability distribution π on a state Ω is a randomised algorithm which takes as input a real number $\delta > 0$ and outputs a sample from a distribution μ such that the *total variation distance* $\frac{1}{2}\sum_{x \in \Omega} |\mu(x) - \pi(x)|$ is at most δ. 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 π is the uniform distribution. We use it in a more general setting.

If $w : \Omega \to \mathbb{R}$ is a *weight function*, then the *Gibbs distribution* π 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 π is uniform. For independent sets with $w(I) = \lambda^{|I|}$, we have

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

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 π, provided the problem is *self-reducible*. Counting independent sets in a graph is a self-reducible problem. (2) can be tightened to

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

2 Markov Chains

2.1 Mixing Time

For general information on Markov chains and approximate counting see [17,18].

Consider a Markov chain on state space Ω with stationary distribution π 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 $\tau_x(\varepsilon) = \min\{n : \mathrm{d_{TV}}(p_n, \pi) \leq \varepsilon\}$, where $\mathrm{d_{TV}}(p_n, \pi)$ is the total variation distance between p_n and π. In the case of the Glauber dynamics for independent sets, the stationary distribution π satisfies (3), and in particular $\pi(\varnothing)^{-1} = P_G(\lambda)$. We will always use $\varnothing$ as our starting state.

Let $\beta_{\max} = \max\{\beta_1, |\beta_{|\Omega|-1}|\}$, where β_1 is the second-largest eigenvalue and $\beta_{|\Omega|-1}$ is the smallest eigenvalue of $\mathbf{P}$. From [9, Proposition 3] follows $\tau_x(\varepsilon) \leq (1-\beta_{\max})^{-1}\left(\ln(\pi(x)^{-1}) + \ln(1/\varepsilon)\right)$, see also [28, Proposition 1(i)]. Hence for $\lambda \geq 1/n$,

$$\tau_{\varnothing}(\varepsilon) \leq (1-\beta_{\max})^{-1}\left(\alpha(G)\ln(n\lambda) + 1 + \ln(1/\varepsilon)\right), \tag{5}$$

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 γ_{XY} from X to Y, namely $X = Z_0 \to Z_2 \to \cdots \to Z_\ell = Y$ 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 γ_{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 ν 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 Ω^* is some set which contains all possible encodings. We usually require ν to be polynomial in n. It is common to refer to the additional information provided by ν 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* ϱ of the chosen set of paths is given by

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

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]:

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

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 $Z' \leftarrow Z \backslash \{v\}$ 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 π is the stationary distribution of the chain. If Z' is obtained from Z by deleting a vertex v then

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

The unweighted version is given by setting $\lambda = 1$, and has uniform stationary distribution. Since the analysis for general λ 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

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

for $\lambda \geq 1/n$. This bound will be dominated by our bound on the relaxation time. We will always use the initial state $Z_0 = \varnothing$, since $\varnothing \in \mathcal{I}(G)$ 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 $X \backslash Y$ black and vertices in $Y \backslash X$ 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 Sect. 3. In Sect. 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 $\mathcal{B} = (B_1, B_2, \ldots, B_r)$ 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* $\mathrm{pw}(G)$ of a given graph G is $\mathrm{pw}(G) = \min_{\mathcal{B}} w(\mathcal{B})$, where the minimum taken over all path decompositions

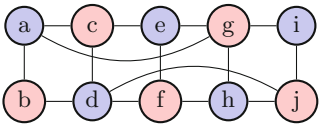


Fig. 1. A bipartite graph

$\mathcal{B}$ 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 $B_i = \varnothing$.

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

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

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

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

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

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

The *bipartite pathwidth* $\mathrm{bpw}(G)$ 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 $\mathrm{bpw}(G) \leq p$. By Lemma 1 $\mathcal{C}_p$ is a hereditary class.

Clearly $\mathrm{bpw}(G) \leq \mathrm{pw}(G)$, 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 Eq. 10. The even more general *interval graphs* do not contain a tripod (depicted in Sect. 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 $\mathrm{bpw}(G) \leq p$. Take $X, Y \in \mathcal{I}(G)$ and let $H_1, \ldots, H_t$ be the connected components of $G[X \oplus Y]$ in lexicographical order. The graph $G[X \oplus Y]$ is bipartite, so every component $H_1, \ldots, H_t$ is connected and bipartite. We will define a canonical path γ_{XY} from X to Y by processing the components $H_1, \ldots, H_t$ in order. Let H_a be the component of $G[X \oplus Y]$ which we are currently processing, and suppose that after processing $H_1, \ldots, H_{a-1}$ we have a partial canonical path $X = Z_0, \ldots, Z_N$. If $a = 0$ then $Z_N = Z_0 = X$. The encoding W_N for Z_N is defined by

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

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

$$Z_N \cap H_s = Y \cap H_s \text{ for } s = 1, \ldots, a-1 \text{ (processed)}, \tag{12}$$
$$Z_N \cap H_s = X \cap H_s \text{ for } s = a, \ldots, t \text{ (not processed)}, \tag{13}$$
$$W_N \cap H_s = X \cap H_s \text{ for } s = 1, \ldots, a-1 \text{ (processed)}, \tag{14}$$
$$W_N \cap H_s = Y \cap H_s \text{ for } s = a, \ldots, t \text{ (not processed)}. \tag{15}$$

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 $Z_N, Z_{N+1}, \ldots, Z_{N+h}$ of independent sets, and a corresponding sequence $W_N, W_{N+1}, \ldots, W_{N+h}$ of encodings, such that $Z_\ell \oplus W_\ell \subseteq X \oplus Y$ and $Z_\ell \cap W_\ell = X \cap Y$ for $j = N, \ldots, N+h$. Define the set of "remembered vertices" $R_\ell = (X \oplus Y) \backslash (Z_\ell \oplus W_\ell)$ for $\ell = N, \ldots, N+h$. By definition, the triple $(Z, W, R) = (Z_\ell, W_\ell, R_\ell)$ satisfies

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

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

Let $\mathcal{B} = (B_1, \ldots, B_r)$ be the lexicographically-least path decomposition of H_a. Here we use the ordering on path decompositions defined at the end of Sect. 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 $B_1, \ldots, B_r$ in order. Initially $R_N = \varnothing$, by (11). If bag B_i is currently being processed and the current independent set is Z and the current encoding is W, then

$$\big(X \cap (B_1 \cup \cdots \cup B_{i-1})\big) \backslash B_i = \big(W \cap (B_1 \cup \cdots \cup B_{i-1})\big) \backslash B_i, \tag{17}$$
$$\big(Y \cap (B_1 \cup \cdots \cup B_{i-1})\big) \backslash B_i = \big(Z \cap (B_1 \cup \cdots \cup B_{i-1})\big) \backslash B_i, \tag{18}$$
$$\big(X \cap (B_{i+1} \cup \cdots \cup B_r)\big) \backslash B_i = \big(Z \cap (B_{i+1} \cup \cdots \cup B_r)\big) \backslash B_i, \tag{19}$$
$$\big(Y \cap (B_{i+1} \cup \cdots \cup B_r)\big) \backslash B_i = \big(W \cap (B_{i+1} \cup \cdots \cup B_r)\big) \backslash B_i. \tag{20}$$

Let Z_ℓ, W_ℓ, R_ℓ 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_ℓ^+ are added to R_ℓ during the preprocessing phase (and must eventually be inserted into the current independent set), and vertices in R_ℓ^- are added to R_ℓ 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, $R_N = R_N^+ = R_N^- = \varnothing$.

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

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

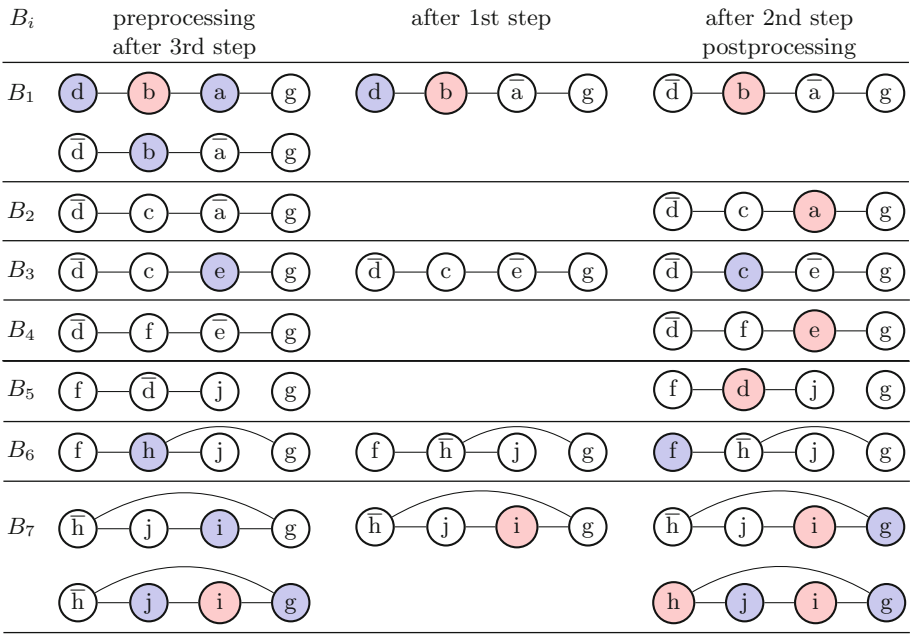
Postprocessing: *All elements of $R_{\ell+1}^-$ which do not belong to B_{i+1} can now be safely added to W_ℓ. This does not change the current independent set or add to the canonical path.*
$W_\ell \leftarrow W_\ell \cup (R_\ell^- \backslash B_{i+1})$; $R_\ell^- \leftarrow R_\ell^- \cap B_{i+1}$;

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

Observe that both Z_ℓ and W_ℓ 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_ℓ, 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_ℓ or R_ℓ^-. A deleted vertex is only added to R_ℓ^- if it belongs to $B_i \cap B_{i+1}$, and so might have a neighbour in W_ℓ. In the insertion steps we add vertices from $\big(B_i \cap (W_\ell \cup R_\ell^+)\big) \backslash B_{i+1}$ to Z_ℓ, 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_ℓ. 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 $X \backslash Y$ is removed from the current independent set at some point, and is never re-inserted, while every vertex of $Y \backslash X$ 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 $\ell_{\max} \leq 2\alpha(G)$.

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



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 $Z \cap B_i = W \cap B_i = \varnothing$. 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 $(Z, Z') \in \gamma_{XY}$.*

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 λ on $\mathcal{I}(G)$ (and initial state $\varnothing$) has mixing time*

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

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

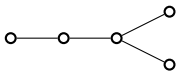
Claw-free graphs exclude the $K_{1,3}$, the *claw*. Claw-free graphs form an important superclass of *line graphs* [4], and independent sets in line graphs are *matchings*.

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 Fig. 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.

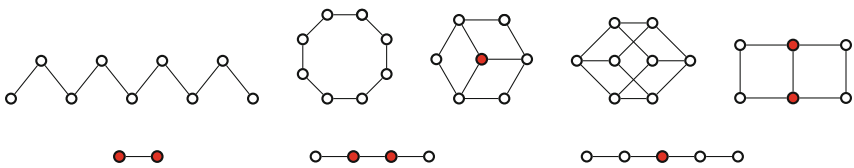


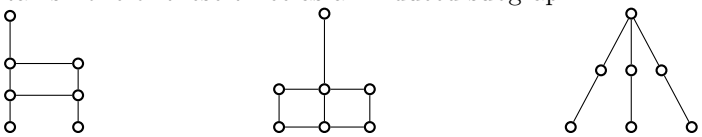
Fig. 2. The path P_9, the cycle C_8, the augmented bipartite wheel BW_3^*, the cube Q_3, an augmented domino, followed by augmented paths P_2^*, P_4^* and P_5^*. (Color figure online)

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 $\mathrm{bpw}(G) \leq d$ for some constant d. The complexity of this question is less clear. Bodlaender [5] has shown that the question $\mathrm{pw}(G) \leq d$, can be answered in $O(2^{d^2} n)$ time. However, this implies nothing about $\mathrm{bpw}(G)$.

In the case of claw-free graphs we can prove stronger sampling results using log-concavity. How far does log-concavity extends 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*, $\mathrm{btw}(G)$. Since $\mathrm{tw}(G) = O(\mathrm{pw}(G) \log n)$ [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. Alekseev, V.E.: Polynomial algorithm for finding the largest independent sets in graphs without forks. Discrete Appl. Math. **135**, 3–16 (2004)
2. Barvinok, A.: Computing the partition function of a polynomial on the Boolean cube. In: Loebl, M., Nešetřil, J., Thomas, R. (eds.) A Journey Through Discrete Mathematics, pp. 135–164. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-44479-6_7

3. Bayati, M., Gamarnik, D., Katz, D., Nair, C., Tetali, P.: Simple deterministic approximation algorithms for counting matchings. In: Proceedings of the STOC, pp. 122–127 (2007)
4. Beineke, L.: Characterizations of derived graphs. J. Comb. Theory **9**, 129–135 (1970)
5. Bodlaender, H.L.: A linear time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**, 1305–1317 (1996)
6. Bodlander, H.L.: A partial k-arboretum of graphs with bounded treewidth. Theor. Comput. Sci. **209**, 1–45 (1998)
7. Chudnovsky, M., Seymour, P.: The roots of the independence polynomial of a clawfree graph. J. Comb. Theory (Ser. B) **97**, 350–357 (2007)
8. Diaconis, P., Saloff-Coste, L.: Comparison theorems for reversible Markov chains. Ann. Appl. Probab. **3**, 696–730 (1993)
9. Diaconis, P., Stroock, D.: Geometric bounds for eigenvalues of Markov chains. Ann. Appl. Probab. **1**, 36–61 (1991)
10. Diestel, R.: Graph Theory. Graduate Texts in Mathematics, vol. 173. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-53622-3
11. Dyer, M., Goldberg, L.A., Greenhill, C., Jerrum, M.: On the relative complexity of approximate counting problems. Algorithmica **38**, 471–500 (2003)
12. Dyer, M., Greenhill, C.: On Markov chains for independent sets. J. Algorithms **35**, 17–49 (2000)
13. Dyer, M., Greenhill, C., Müller, H.: Counting independent sets in graphs with bounded bipartite pathwidth. Preprint: arXiv:1812.03195 (2018)
14. Efthymiou, C., Hayes, T., Stefankovic, D., Vigoda, E., Yin, Y.: Convergence of MCMC and loopy BP in the tree uniqueness region for the hard-core model. In: Proceedings of the FOCS 2016, pp. 704–713. IEEE (2016)
15. Greenhill, C.: The complexity of counting colourings and independent sets in sparse graphs and hypergraphs. Comput. Complex. **9**, 52–72 (2000)
16. Harvey, N.J.A., Srivastava, P., Vondrák, J.: Computing the independence polynomial: from the tree threshold down to the roots. In: Proceedings of the SODA 2018, pp. 1557–1576 (2018)
17. Jerrum, M.: Mathematical foundations of the Markov chain Monte Carlo method. In: Habib, M., McDiarmid, C., Ramirez-Alfonsin, J., Reed, B. (eds.) Probabilistic Methods for Algorithmic Discrete Mathematics. AC, vol. 16, pp. 116–165. Springer, Heidelberg (1998). https://doi.org/10.1007/978-3-662-12788-9_4
18. Jerrum, M.: Counting, Sampling and Integrating: Algorithms and Complexity. Lectures in Mathematics - ETH Zürich. Birkhäuser, Basel (2003)
19. Jerrum, M., Sinclair, A.: Approximating the permanent. SIAM J. Comput. **18**, 1149–1178 (1989)
20. Jerrum, M., Sinclair, A., Vigoda, E.: A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. J. ACM **51**, 671–697 (2004)
21. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theoret. Comput. Sci. **43**, 169–188 (1986)
22. Matthews, J.: Markov chains for sampling matchings, Ph.D. thesis, University of Edinburgh (2008)
23. Luby, M., Vigoda, E.: Approximately counting up to four. In: Proceedings of the STOC 1995, pp. 150–159. ACM (1995)
24. Minty, G.J.: On maximal independent sets of vertices in claw-free graphs. J. Comb. Theory Ser. B **28**, 284–304 (1980)

25. Patel, V., Regts, G.: Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. SIAM J. Comput. **46**, 1893–1919 (2017)
26. Provan, J.S., Ball, M.O.: The complexity of counting cuts and of computing the probability that a graph is connected. SIAM J. Comput. **12**, 777–788 (1983)
27. Robertson, N., Seymour, P.D.: Graph minors I: excluding a forest. J. Comb. Theory Ser. B **35**, 39–61 (1983)
28. Sinclair, A.: Improved bounds for mixing rates of Markov chains and multicommodity flow. Comb. Probab. Comput. **1**, 351–370 (1992)
29. Sly, A.: Computational transition at the uniqueness threshold. In: Proceedings of the FOCS 2010, pp. 287–296. IEEE (2010)
30. Vadhan, S.P.: The complexity of counting in sparse, regular, and planar graphs. SIAM J. Comput. **31**, 398–427 (2001)
31. Weitz, D.: Counting independent sets up to the tree threshold. In: Proceedings of the STOC 2006, pp. 140–149. ACM (2006)