On Counting Perfect Matchings in General Graphs

Daniel Štefankovič[1], Eric Vigoda[2], and John Wilmes[2(✉)]

[1] University of Rochester, Rochester, USA
stefanko@cs.rochester.edu
[2] Georgia Institute of Technology, Atlanta, USA
{vigoda,wilmesj}@gatech.edu

Abstract. Counting perfect matchings has played a central role in the theory of counting problems. The permanent, corresponding to bipartite graphs, was shown to be #P-complete to compute exactly by Valiant (1979), and a fully polynomial randomized approximation scheme (FPRAS) was presented by Jerrum, Sinclair, and Vigoda (2004) using a Markov chain Monte Carlo (MCMC) approach. However, it has remained an open question whether there exists an FPRAS for counting perfect matchings in general graphs. In fact, it was unresolved whether the same Markov chain defined by JSV is rapidly mixing in general. In this paper, we show that it is not. We prove torpid mixing for any weighting scheme on hole patterns in the JSV chain. As a first step toward overcoming this obstacle, we introduce a new algorithm for counting matchings based on the Gallai−Edmonds decomposition of a graph, and give an FPRAS for counting matchings in graphs that are sufficiently close to bipartite. In particular, we obtain a fixed-parameter tractable algorithm for counting matchings in general graphs, parameterized by the greatest "order" of a factor-critical subgraph.

1 Introduction

Counting perfect matchings is a fundamental problem in the area of counting/sampling problems. For an undirected graph $G = (V, E)$, let $\mathcal{P}$ denote the set of perfect matchings of G. Can we compute (or estimate) $|\mathcal{P}|$ in time polynomial in $n = |V|$? For which classes of graphs?

A polynomial-time algorithm for the corresponding decision and optimization problems of determining if a given graph contains a perfect matching or finding a matching of maximum size was presented by Edmonds [2]. For the counting problem, a classical algorithm of Kasteleyn [9] gives a polynomial-time algorithm for exactly computing $|\mathcal{P}|$ for planar graphs.

For bipartite graphs, computing $|\mathcal{P}|$ is equivalent to computing the permanent of $n \times n$ $(0,1)$-matrices. Valiant [14] proved that the $(0,1)$-Permanent is #P-complete. Subsequently attention turned to the Markov Chain Monte Carlo (MCMC) approach. A Markov chain where the mixing time is polynomial in n is said to be *rapidly mixing*, and one where the mixing time is exponential in

$\Omega(n)$ is referred to as *torpidly mixing*. A rapidly mixing chain yields an FPRAS (fully polynomial-time randomized approximation scheme) for the corresponding counting problem of estimating $|\mathcal{P}|$ [8].

For dense graphs, defined as those with minimum degree >n/2, Jerrum and Sinclair [6] proved rapid mixing of a Markov chain defined by Broder [1], which yielded an FPRAS for estimating $|\mathcal{P}|$. The Broder chain walks on the collection $\Omega = \mathcal{P} \cup \mathcal{N}$ of perfect matchings $\mathcal{P}$ and near-perfect matchings $\mathcal{N}$; a near-perfect matching is a matching with exactly 2 holes or unmatched vertices. Jerrum and Sinclair [6], more generally, proved rapid mixing when the number of perfect matchings is within a $\mathsf{poly}(n)$ factor of the number of near-perfect matchings, i.e., $|\mathcal{P}|/|\mathcal{N}| \geq 1/\mathsf{poly}(n)$. A simple example, referred to as a "chain of boxes" which is illustrated in Fig. 1, shows that the Broder chain is torpidly mixing. This example was a useful testbed for catalyzing new approaches to solving the general permanent problem.

Jerrum et al. [7] presented a new Markov chain on $\Omega = \mathcal{P} \cup \mathcal{N}$ with a non-trivial weighting scheme on the matchings based on the holes (unmatched vertices). They proved rapid mixing for any bipartite graph with the requisite weights used in the Markov chain, and they presented a polynomial-time algorithm to learn these weights. This yielded an FPRAS for estimating $|\mathcal{P}|$ for all bipartite graphs. That is the current state of the art (at least for polynomial-time, or even sub-exponential-time algorithms).

Could the JSV-Markov chain be rapid mixing on non-bipartite graphs? Previously there was no example for which torpid mixing was established, it was simply the case that the proof in [7] fails. We present a relatively simple example where the JSV-Markov chain fails for the weighting scheme considered in [7]. More generally, the JSV-chain is torpidly mixing on our class of examples for any weighting scheme based on the hole patterns, see Theorem 3 in Sect. 2 for a formal statement following the precise definition of the JSV-chain.

A natural approach for non-bipartite graphs is to consider Markov chains that exploit odd cycles or blossoms in the manner of Edmonds' algorithm. We observe that a Markov chain which considers *all* blossoms for its transitions is intractable since sampling all blossoms is NP-hard, see Theorem 5. On the other hand, a chain restricted to minimum blossoms is not powerful enough to overcome our torpid mixing examples. See Sect. 3 for a discussion.

Finally we utilize the Gallai−Edmonds graph decomposition into factor-critical graphs [2–4,12] to present new algorithmic insights that may overcome the obstacles in our classes of counter-examples. In Sect. 4, we describe how the Gallai−Edmonds decomposition can be used to efficiently estimate $|\mathcal{P}|$, the number of perfect matchings, in graphs whose factor-critical subgraphs have bounded order (Theorem 7), as well as in the torpid mixing example graphs (Theorem 8).

Although all graphs are explicitly defined in the text below, figures depicting these graphs are deferred to the appendix.

1.1 Markov Chains

Consider an ergodic Markov chain with transition matrix P on a finite state space Ω, and let π denote the unique stationary distribution. We will usually

assume the Markov chain is time reversible, i.e., that it satisfies the **detailed balance condition** $\pi(x)P(x,y) = \pi(y)P(x,y)$ for all states $x, y \in \Omega$.

For a pair of distributions μ and ν on Ω we denote their total variation distance as $d_{\mathsf{TV}}(\mu,\nu) = \frac{1}{2}\sum_{x\in\Omega}|\mu(x) - \nu(x)|$. The standard notion of **mixing time** T_{mix} is the number of steps from the worst starting state $X_0 = i$ to reach total variation distance $\leq 1/4$ of the stationary distribution π, i.e., we write $T_{\text{mix}} = \max_{i\in\Omega}\min\{t : d_{\mathsf{TV}}(P^t(i,\cdot),\pi) \leq 1/4\}$.

We use conductance to obtain lower bounds on the mixing time. For a set $S \subset \Omega$ its **conductance** is defined as:

$$\Phi(S) = \frac{\sum_{x\in S, y\notin S}\pi(x)P(x,y)}{\sum_{x\in S}\pi(x)}.$$

Let $\Phi_* = \min_{S\subset\Omega:\pi(S)\leq 1/2}\Phi(S)$. Then (see, e.g., [10,13])

$$T_{\text{mix}} \geq \frac{1}{4\Phi_*}. \tag{1}$$

1.2 Factor-Critical Graphs

A graph $G = (V,E)$ is **factor-critical** if for every vertex $v \in V$, the graph induced on $V \setminus \{v\}$ has a perfect matching. (In particular, $|V|$ is odd.)

Factor-critical graphs are characterized by their "ear" structure. The **quotient** G/H of a graph G by a (not necessarily induced) subgraph H is derived from G by deleting all edges in H and contracting all vertices in H to a single vertex v_H (possibly creating loops or multi-edges). An **ear** of G relative a subgraph H of G is simply a cycle in G/H containing the vertex v_H.

Theorem 1 (Lovász [11]). *A graph G is factor-critical if and only if there is a decomposition $G = C_0 \cup \cdots \cup C_r$ such that C_0 is a single vertex, and C_i is an odd-length ear in G relative to $\bigcup_{j<i} C_j$, for all $0 < i \leq r$.*

Furthermore, if G is factor critical, there exists such a decomposition for every choice of vertex C_0, and the order *r of the decomposition is independent of all choices.*

Since the number of ears in the ear decomposition of a factor-critical graph depends only on the graph, and not on the choice made in the decomposition, we say the **order** of the factor-critical graph G is the number r of ears in any ear decomposition of G.

Factor-critical graphs play a central role in the Gallai−Edmonds structure theorem for graphs. We state an abridged version of the theorem below.

Given a graph G, let $D(G)$ be the set of vertices that remain unmatched in at least one maximum matching of G. Let $A(G)$ be the set of vertices not in $D(G)$ but adjacent to at least one vertex of $D(G)$. And let $C(G)$ denote the remaining vertices of G.

Theorem 2 (Gallai–Edmonds Structure Theorem). *The connected components of $D(G)$ are factor-critical. Furthermore, every maximum matching of G induces a perfect matching on $C(G)$, a near-perfect matching on each connected component of $D(G)$, and matches all vertices in $A(G)$ with vertices from distinct connected components of $D(G)$.*

2 The Jerrum–Sinclair–Vigoda Chain

We recall the definition of the original Markov chain proposed by Broder [1]. The state space of the chain is $\Omega = \mathcal{P} \cup \bigcup_{u,v} \mathcal{N}(u,v)$ where $\mathcal{P}$ is the collection of perfect matchings and $\mathcal{N}(u,v)$ are near-perfect matchings with holes at u and v (i.e., vertices u and v are the only unmatched vertices). The transition rule for a matching $M \in \Omega$ is as follows:

1. If $M \in \mathcal{P}$, randomly choose an edge $e \in M$ and transition to $M \setminus \{e\}$.
2. If $M \in \mathcal{N}(u,v)$, randomly choose a vertex $x \in V$. If $x \in \{u,v\}$ and u is adjacent to v, transition to $M \cup \{(u,v)\}$. Otherwise, let $y \in V$ be the vertex matched with x in M, and randomly choose $w \in \{u,v\}$. If x is adjacent to w, transition to the matching $M \cup \{(x,w)\} \setminus \{(x,y)\}$.

The chain $\mathfrak{X}_{\mathrm{B}}$ is symmetric, so its stationary distribution is uniform. In particular, when $|\mathcal{P}|/|\Omega|$ is at least inverse-polynomial in n, we can efficiently generate uniform samples from $\mathcal{P}$ via rejection sampling, given access to samples from the stationary distribution of $\mathfrak{X}_{\mathrm{B}}$.

In order to sample perfect matchings even when $|\Omega|/|\mathcal{P}|$ is exponentially large, Jerrum et al. [7] introduce a new chain $\mathfrak{X}_{\mathrm{JSV}}$ that changes the stationary distribution of $\mathfrak{X}_{\mathrm{B}}$ by means of a Metropolis filter. The new stationary distribution is uniform across *hole patterns*, and then uniform within each hole pattern, i.e., for every $M \in \Omega$, the stationary probability of M is proportional to $1/|\mathcal{N}(u,v)|$ if $M \in \mathcal{N}(u,v)$, and proportional to $1/|\mathcal{P}|$ if $M \in \mathcal{P}$.

We define $\mathfrak{X}_{\mathrm{JSV}}$ in greater detail. For $M \in \Omega$, define the weight function

$$w(M) = \begin{cases} \frac{1}{|\mathcal{P}|} & \text{if } M \in \mathcal{P} \\ \frac{1}{|\mathcal{N}(u,v)|} & \text{if } M \in \mathcal{N}(u,v) \end{cases} \tag{2}$$

Definition 1. *The chain $\mathfrak{X}_{JSV}$ has the same state space as $\mathfrak{X}_B$. The transition rule for a matching $M \in \Omega$ is as follows:*

1. *First, choose a matching $M' \in \Omega$ to which M may transition, according to the transition rule for $\mathfrak{X}_B$.*
2. *With probability $\min\{1, w(M')/w(M)\}$, transition to M'. Otherwise, stay at M.*

In their paper, Jerrum et al. [7] in fact analyze a more general version of the chain $\mathfrak{X}_{\mathrm{JSV}}$ that allows for arbitrary edge weights in the graph. They show that the chain is rapidly mixing for bipartite graphs G. (They also study the

separate problem of estimating the weight function w, and give a "simulating annealing" algorithm that allows the weight function w to be estimated by gradually adjusting edge weights to obtain an arbitrary bipartite graph G from the complete bipartite graph.) Their analysis of the mixing time uses a canonical paths argument that crucially relies on the bipartite structure of the graph. However, it remained an open question whether a different analysis of the same chain $\mathfrak{X}_{\mathrm{JSV}}$, perhaps using different canonical paths, might generalize to non-bipartite graphs. We rule out this approach.

In fact, we rule out a more general family of Markov chains for sampling perfect matchings. We say a Markov chain is "of $\mathfrak{X}_{\mathrm{JSV}}$ type" if it has the same state space as $\mathfrak{X}_{\mathrm{JSV}}$, with transitions as defined in Definition 1, for *some* weight function $w(M)$ (not necessarily the same as in Eq. (2)) depending only the hole pattern of the matching M.

Theorem 3. *There exists a graph G on n vertices such that for any Markov chain $\mathfrak{X}$ of $\mathfrak{X}_{JSV}$ type on G, either the stationary probability of $\mathcal{P}$ is at most $\exp(-\Omega(n))$, or the mixing time of $\mathfrak{X}$ is at least $\exp(\Omega(n))$.*

The graph G of Theorem 3 is constructed from several copies of a smaller gadget H, which we now define.

Definition 2. *The **chain of boxes gadget** B_k of length k is the graph on $4k$ vertices depicted in Fig. 1. To construct B_k, we start with a path $P_{2k-1} = v_0, v_1, \ldots, v_{2k}$ of length $2k-1$. Then, for every even edge $\{v_{2i}, v_{2i+1}\}$ on the path, we add two additional vertices a_i, b_i, along with edges to form a path $v_{2i}, a_i, b_i, v_{2i+1}$ of length 3.*

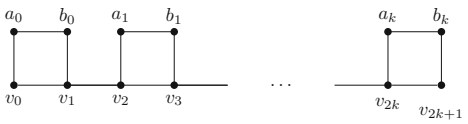

Fig. 1. The "chain of boxes" gadget B_k, which has 2^k perfect matchings, but only a single matching in $\mathcal{N}(v_0, v_{2k+1})$.

Observation 4. *The chain of boxes gadget B_k has 2^k perfect matchings, but only one matching in $\mathcal{N}(v_0, v_{2k+1})$.*

Definition 3. *The **torpid mixing gadget** H_k is the graph depicted in Fig. 2. To construct H, first take a C_{12} and label two antipodal vertices as a and b. Add an edge between a and b, and label the two vertices farthest from a and b as u and v. Label the neighbor of u closest to a as w_1, and the other neighbor of u as w_2. Label the neighbor of v closest to a as z_1 and the other neighbor of v as z_2. Finally, add four chain-of-boxes gadgets B_k, identifying the vertices v_0 and v_{2k} of the gadgets with w_1 and a, with a and z_1, with w_2 and b, and with b and z_2, respectively.*

Note that in Figs. 2 and 3, one "box" from each copy of B_k in the torpid mixing gadget is left undrawn, for visual clarity.

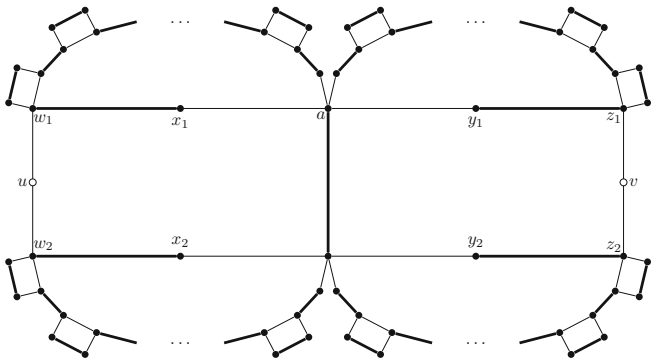

Fig. 2. The torpid mixing gadget H_k. The unique matching $M \in \mathcal{N}(u, v)$ is depicted with thick edges.

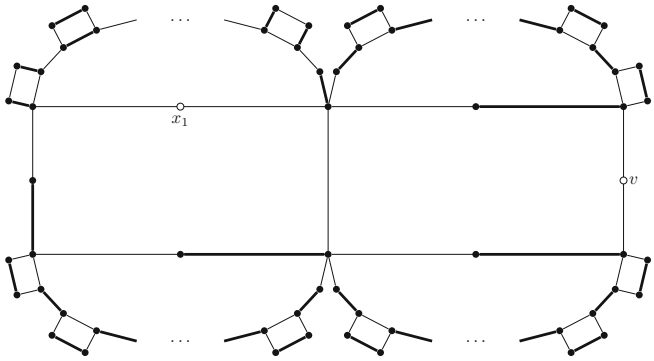

Fig. 3. A matching $M' \in \mathcal{N}(x_1, v)$. There are exponentially many matchings with the same hole pattern, obtained by alternating the 4-cycles above x_1.

Lemma 1. *The torpid mixing gadget $H = H_k$ has $16k + 4$ vertices and exactly 2 perfect matchings. Furthermore, $|\mathcal{N}_H(u, v)| = 1$ and $\mathcal{N}_H(x_1, v) \geq 2^k$.*

The unique matching in $\mathcal{N}_H(u, v)$ is depicted in Fig. 2, and an example of a matching in $\mathcal{N}_H(x_1, v)$, which generalizes easily to the entire family, is depicted in Fig. 3. The details of the proof are deferred to the full version of the paper.

The torpid mixing gadget already suffices on its own to show that the Markov chain $\mathfrak{X}_{\mathfrak{X}_{\text{JSV}}}$ defined in [7] is torpidly mixing. In particular, the conductance out of the set $\mathcal{N}_H(x_1, v) \subseteq \Omega(H)$ is $2^{-\Omega(k)}$. In order to prove the stronger claim of Theorem 3, that every Markov chain of $\mathfrak{X}_{\text{JSV}}$-type fails to efficiently sample perfect matchings, we construct a slightly larger graph from copies of the torpid mixing gadgets.

Definition 4. *The* ***counterexample graph*** G_k *is the graph depicted in Fig. 4. It is defined by replacing every third edge of the twelve-cycle* C_{12} *with the gadget* H_k *defined in Fig. 2. Specifically, let* $\{u_i, v_i\}$ *be the* $3i$*-th edge of* C_{12} *for* $i \in \{1, \ldots, 4\}$*. We delete each edge* $\{u_i, v_i\}$ *and replace it with a copy of* H*, identifying the vertices* u *and* v *of* H *with vertices* u_i *and* v_i *of* C_{12}*. The resulting graph is* G_k*. Thus, of the* 12 *original vertices in* C_{12}*,* 8 *of the corresponding vertices in* G_k *participate in a copy of the gadget* H*, and* 4 *do not. These* 4 *vertices of* G_k *which do not participate in any copy of the gadget* H *are labeled* $t_1, \ldots, t_4$ *in cyclic order, and the copies of the gadget* H *are labeled* $H_1, \ldots H_4$ *in cyclic order, with* H_1 *coming between* t_1 *and* t_2*, and so on. Thus,* t_1 *is adjacent to* u_1 *and* v_4*,* t_i *is adjacent to* u_i *and* v_{i-1} *for* $i \in \{2, \ldots, 4\}$*, and* H_i *contains both* u_i *and* v_i*.*

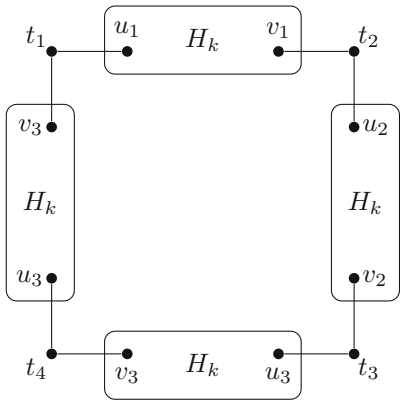

Fig. 4. The "counterexample graph" G_k on which $\mathfrak{X}_{\text{JSV}}$ is torpidly mixing. The boxes labeled H_k represent copies of the torpid mixing gadget of Definition 3.

In particular, G_k has $4|V(H)| + 4 = 64k + 8$ vertices.

The perfect and near-perfect matchings of G_k are naturally divided into four intersecting families. For $i \in \{1, \ldots, 4\}$ we define S_i to be the collection of (perfect and near-perfect) matchings $M \in \Omega(G_k)$ such that the restriction of M to H_i has two holes, at u_i and v_i, i.e., such that the vertices u_i and v_i either have holes in M or are matched outside of H_i.

Lemma 2. *The counterexample graph* G_k *has exactly* 8 *perfect matchings. Of these,* 4 *are in* $S_1 \cap S_3 \setminus (S_2 \cup S_4)$ *and* 4 *are in* $S_2 \cap S_4 \setminus (S_1 \cup S_3)$*.*

The proof of this lemma is deferred to the full version of the paper.

In the proof below, we use the notation $\mathcal{N}(M)$ denote the collection of matchings with the hole pattern as M. That is, $\mathcal{N}(M) = \mathcal{P}$ if $M \in \mathcal{P}$, and $\mathcal{N}(M) = \mathcal{N}(u, v)$ if $M \in \mathcal{N}(u, v)$.

Proof (Proof of Theorem 3). Let G_k be the counterexample graph of Definition 4. We will show that the set $S_1 \cup S_3 \subseteq \Omega(G_k)$ has poor conductance, unless the stationary probability of $\mathcal{P}_{G_k}$ is small. We will write $A = S_1 \cup S_3$ and $\overline{A} = \Omega(G_k) \setminus (S_1 \cup S_3)$.

Let $M \in A$ and $M' \in \overline{A}$ be such that $P(M, M') > 0$. We claim that neither M nor M' are perfect matchings. Assume without loss of generality that $M \in S_1$. If $M \in S_1$ is a perfect matching, then $M \in P_2$ and so $M \in S_3$. The only legal transitions from M to $\Omega \setminus S_1$ are those that introduce additional holes within H_1, but none of these transitions to a matching outside of S_3. Hence, M cannot be perfect. But if M' is perfect, then $M' \in P_1$, and so M' induces a perfect matching on S_1. But then the transition from M to M' must simultaneously affect u_1 and v_1, and no such transition exists.

We denote by $\partial\overline{A}$ the set of matchings $M' \in \overline{A}$ such that there exists a matching $M \in A$ with $P(M, M') > 0$. We claim that for every matching $M' \in \overline{A}$, we have

$$|\mathcal{N}(M') \cap \partial\overline{A}| \leq 2^{k-1}|\mathcal{N}(M')|\,. \tag{3}$$

Let $M' \in \partial\overline{A}$, and let $M \in A$ be such that $P(M, M') > 0$. Suppose first that $M \in S_1$. Label the vertices of H_1 as in Fig. 2, identifying u_1 with u and v_1 with v. Let N be the matching on $H = H_1$ induced by M, and let N' be the matching on H_1 induced by M'. We have $N \in \mathcal{N}_H(u_1, v_1)$. But by Lemma 1, we have $|\mathcal{N}_H(u_1, v_1)| = 1$, i.e., N is exactly the matching depicted in Fig. 2. The only transitions that remove the hole at u are the two that shift the hole to x_1 or x_2, and the only transitions that remove the hole at v are the two that shift the hole to y_1 or y_2. So, without loss of generality, by the symmetry of G_k, we have $N' \in \mathcal{N}_H(x_1, v_1)$. By Lemma 1, $|\mathcal{N}_H(x_1, v_1)| \geq 2^k$, but only one matching in $\mathcal{N}_H(x_1, v_1)$ has a legal transition to N. Therefore, if we replace the restriction of M' to H_1 with any other matching in $\mathcal{N}_H(x_1, v_1)$, we obtain another matching $M'' \in \mathcal{N}(M')$, but M'' has no legal transition to any matching in $\mathcal{N}(M)$. Hence, only a 2^{-k}-fraction of $\mathcal{N}(M')$ has a legal transition to S_1, and similarly only a 2^{-k}-fraction of $\mathcal{N}(M')$ has a legal transition to S_3. In particular, we have proved Eq. (3).

From Eq. (3), it immediately follows that the stationary probability of $\partial\overline{A}$ is

$$\pi(\partial\overline{A}) = \sum_{M' \in \partial\overline{A}} \pi(M') = \sum_{M' \in \overline{A}} \pi(M') \frac{|\mathcal{N}(M') \cap \partial\overline{A}|}{|\mathcal{N}(M')|} = 2^{-k+1}\pi(\overline{A}) \tag{4}$$

We now compute

$$\sum_{\substack{M \in A, M' \in \overline{A} \\ P(M,M')>0}} \pi(M)P(M, M') = \sum_{\substack{M \in A, M' \in \overline{A} \\ P(M,M')>0}} \pi(M')P(M', M) \leq \pi(\partial(\overline{A})) < 2^{-k+1}\pi(\overline{A}),$$

where we first use the detailed balance condition and then Eq. (4).

Now by (1) and the definition of conductance, we have

$$\frac{1}{4\tau_{\mathfrak{X}}} < \Phi(A) < 2^{-k}\frac{\pi(\overline{A})}{\pi(A)}.$$

In particular, if $\tau_{\mathfrak{X}} < 2^{k/2-2}$, then $\pi(\overline{A}) > 2^{k/2+1}\pi(A)$. Suppose this is the case. By Lemma 2, half of the perfect matchings of G_k belong to A. In particular, $\pi(\mathcal{P}_{G_k}) \leq 2\pi(A) < 2^{-k/2+2}$. Hence, either the stationary probability of $\mathcal{P}$ is at most $2^{-k/2+2} = \exp(-\Omega(n))$, or the mixing time of $\mathfrak{X}$ is at least $2^{k/2-2} = \exp(\Omega(n))$. □

We remark that the earlier Markov chain studied by Broder [1] and Jerrum and Sinclair [6] is also torpidly mixing on the counterexample graph of Definition 4, since the ratio of near-perfect matchings to perfect matchings is exponential [6].

3 Chains Based on Edmonds' Algorithm

Given that Edmonds' classical algorithm for *finding* a perfect matching in a bipartite graph requires the careful consideration of odd cycles in the graph, it is reasonable to ask whether a Markov chain for counting perfect matchings should also somehow track odd cycles. In this section, we briefly outline some of the difficulties of such an approach.

A *blossom* of length k in a graph G equipped with a matching M is simply an odd cycle of length $2k+1$ in which k of the edges belong to M. Edmonds' algorithm finds augmenting paths in a graph by exploring the alternating tree rooted at an unmatched vertex, and contracting blossoms to a vertex as they are encountered. Given a blossom B containing an unmatched vertex u, there is an alternating path of even length to every vertex $v \in B$. *Rotating* B to v means shifting the hole at u to v by alternating the u-v path in B.

Adding rotation moves to a Markov chain in the style of $\mathfrak{X}_{\text{JSV}}$ is an attractive possible solution to the obstacles presented in the previous section. Indeed, if it were possible to rotate the 7-cycle containing u and a in the graph in Fig. 2, it might be possible to completely avoid problematic holes at x_1 or x_2.

The difficulty in introducing such an additional move the Markov chain $\mathfrak{X}_{\text{JSV}}$ is in defining the set of *feasible* blossoms that may be rotated, along with a probability distribution over such blossoms. In order to be useful, we must be able to *efficiently sample* from the feasible blossoms at a given near-perfect matching M. Furthermore, the feasible blossoms must respect time reversibility: if B is feasible when the hole is at $u \in B$, then it must also be feasible after rotating the hole to $v \in B$; reversibility of the Markov chain is needed so that we understand its stationary distribution. Finally, the feasible blossoms must be rich enough to avoid the obstacles outlined in the previous section.

The set of "minimum length" blossoms at a given hole vertex u satisfies the first criterion of having an efficient sampling algorithm. But it is easy to see that if only minimum length blossoms are feasible, then the obstacles outlined in the

previous section will still apply (simply by adding a 3-cycle at every vertex). Moreover, families blossoms characterized by minimality may struggle to satisfy the second criterion of time-reversibility. In Fig. 5, there is a unique blossom containing u, but after rotating the hole to v, it is no longer minimal.

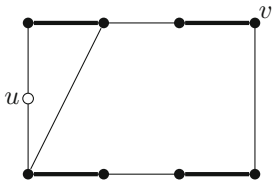

Fig. 5. After rotating the blossom so that the hole is moved from u to v, the blossom is no longer "minimal".

On the other hand, the necessity of having an efficient sampling algorithm for the feasible blossoms already rules out the simplest possibility, namely, the uniform distribution over *all* blossoms containing a given hole vertex u. Indeed, if we could efficiently sample from the uniform distribution over all blossoms containing a given vertex u, then by an entropy argument we could find arbitrarily large odd cycles in the graph, which is NP-hard.

Theorem 5. *Let* Sampling Blossoms *problem be defined as follows. The input is an undirected graph G and a near-perfect matching M with holes at $w, r \in V(G)$. The output is a uniform sample from the uniform distribution of blossoms containing w. Unless NP=RP there is no randomized polynomial-time sampler for* Sampling Blossoms.

The proof is deferred to the full version of the paper.

4 A Recursive Algorithm

We now explore a new recursive algorithm for counting matchings, based on the Gallai−Edmonds decomposition. In the worst case, this algorithm may still require exponential time. However, for graphs that have additional structural properties, for example, those that are "sufficiently close to bipartite" in a sense that will be made precise, our recursive algorithm runs in polynomial time. In particular, it will run efficiently on examples similar to those used to prove torpid mixing of Markov chains in the previous section.

We now state the algorithm. It requires as a subroutine an algorithm for computing the permanent of the bipartite adjacency matrix of a bipartite graph G up to accuracy ε. We denote this subroutine by Permanent(G, ε). The Permanent subroutine requires time polynomial in $|V(G)|$ and $1/\varepsilon$ using the algorithm of Jerrum et al. [7], but we use it as a black-box.

We first argue the correctness of the algorithm.

Algorithm 1. Recursive algorithm for approximately counting the number of perfect matchings in a graph

1: **procedure** RECURSIVE-COUNT(G, ε)
2: If $V(G) = \emptyset$, return 1.
3: Choose $u \in V(G)$.
4: Compute the Gallai–Edmonds decomposition of $G - u$.
5: **for all** $v \in D(G - u)$ **do**
6: $H_v \leftarrow$ the connected component of $G - u$ containing v
7: $m_v \leftarrow$ RECURSIVE-COUNT($H_v - v, \varepsilon/(2n)$)
8: **end for**
9: $m_C \leftarrow$ RECURSIVE-COUNT($C(G - u), \varepsilon/3$)
10: Let $X = A(G - u) \cup \{u\}$, and let Y be the set of connected components in $D(G - u)$. Let G' be the bipartite graph on (X, Y) defined as follows: for every $x \in X$ and $H \in Y$, if x has any neighbors in H in G', add an edge $\{x, H\}$ in G' with weight

$$w(x, H) = \sum_{v \in N(x) \cap H} m_v \,.$$

11: **return** $m_C *$ PERMANENT($G', \varepsilon/3$)
12: **end procedure**

Theorem 6. *Algorithm 1 computes the number of perfect matchings in G to within accuracy ε.*

Proof. We show that the algorithm is correct for graphs on n vertices, assuming it is correct for all graphs on at most $n - 1$ vertices.

We claim that permanent of the incidence matrix of G' defined on line 10 equals the number of perfect matchings in G. Indeed, every perfect matching M of G induces a maximum matching M_u on $G - u$. By the Gallai–Edmonds theorem, M_u matches each element of $A(G')$ with a vertex from a distinct component of $D(G')$, leaving one component of $D(G')$ unmatched. Vertex u must therefore be matched in M with a vertex from the remaining component of $D(G')$. Therefore, M induces a perfect matching M' on G'. Now let $H_x \in Y$ be the vertex of G' matched to x for each $x \in X$. Then the number of distinct matchings of G inducing the same matching M'' on G'' is exactly

$$\prod_{x \in X} \sum_{v \in N(x) \cap H_x} m_v = \prod_{x \in X} w(x, H_x)$$

which is the contribution of M' to the permanent of G'. Similarly, from an arbitrary matching M' of G', with H_x defined as above, we obtain $\prod_{x \in X} w(x, H_x)$ matchings of G, proving the claim.

Hence, it suffices to to compute the permanent of the incidence matrix of G' up to accuracy ε. We know the entries of the incidence matrix up to accuracy $\varepsilon/(2n)$, and $(1+\varepsilon/(2n))^{n/2} < 1+\varepsilon/3$ for ε sufficiently small. Therefore, it suffices to compute the permanent of our approximation of the incidence matrix up to accuracy $\varepsilon/3$ to get overall accuracy better than ε. □

The running time of Algorithm 1 is sensitive to the choice of vertex u on line 3. If u can be chosen so that each component of $D(G-u)$ is small, then the algorithm is an efficient divide-and-conquer strategy. More generally, if u can be chosen so that each component of $D(G-u)$ is in some sense "tractable", then an efficient divide-and-conquer strategy results. In particular, since it is possible to exactly count the number of perfect matchings in a factor-critical graph of bounded order in polynomial time, we obtain an efficient algorithm for approximately counting matchings in graphs whose factor-critical subgraphs have bounded order. This is the sense in which Algorithm 1 is efficient for graphs "sufficiently close" to bipartite.

Theorem 7. *Suppose every factor-critical subgraph of G has order at most k. Then the number of perfect matchings in G can be counted to within accuracy ε in time $2^{O(k)}\mathsf{poly}(n, 1/\varepsilon)$.*

The essential idea of the proof is to first observe that a factor-critical graph can be shrunk to a graph with $O(k)$ edges having the same number of perfect matchings after deleting any vertex. The number of perfect matchings can then be counted by brute force in time $2^{O(k)}\mathsf{poly}(n)$. This procedure replaces the recursive calls on line 6 of the algorithm. The details of the proof are deferred to the full version of the paper.

We note that Theorem 7 is proved by eliminating recursive calls in the algorithm. Although the recursive calls of Algorithm 1 can be difficult to analyze, they can also be useful, as we now demonstrate by showing that Algorithm 1 runs as-is in polynomial time on the counterexample graph of Definition 4, for appropriate choice of the vertex u on the line 3 of the algorithm.

Theorem 8. *Algorithm 1 runs in polynomial time on the counterexample graph of Definition 4, for appropriate choice of the vertex u on the line 3 of the algorithm.*

The proof is given in the full version of the paper.

Acknowledgements. This research was supported in part by NSF grants CCF-1617306, CCF-1563838, CCF-1318374, and CCF-1717349. The authors are grateful to Santosh Vempala for many illuminating conversations about Markov chains and the structure of factor-critical graphs.

References

1. Broder, A.Z.: How hard is it to marry at random? (On the approximation of the permanent). In: Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC), pp. 50–58 (1986). Erratum in Proceedings of the 20th Annual ACM Symposium on Theory of Computing, p. 551 (1988)
2. Edmonds, J.: Paths, trees, and flowers. Can. J. Math. **17**, 449–467 (1965)
3. Gallai, T.: Kritische Graphen II. Magyar Tud. Akad. Mat. Kutató Int. Kőzl. **8**, 273–395 (1963)

4. Gallai, T.: Maximale systeme unabhängiger kanten. Magyar Tud. Akad. Mat. Kutató Int. Kőzl **9**, 401–413 (1964)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
6. Jerrum, M., Sinclair, A.: Approximating the permanent. SIAM J. Comput. **18**(6), 1149–1178 (1989)
7. Jerrum, M., Sinclair, A., Vigoda, E.: A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. J. ACM **51**(4), 671–697 (2004)
8. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theoret. Comput. Sci. **43**(2–3), 169–188 (1986)
9. Kasteleyn, P.W.: Graph theory and crystal physics. In: Graph Theory and Theoretical Physics, pp. 43–110, Academic Press, London (1967)
10. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society, Providence (2009)
11. Lovász, L.: A note on factor-critical graphs. Stud. Sci. Math. Hungar **7**(11), pp. 279–280 (1972)
12. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, Hoboken (1998)
13. Sinclair, A.J.: Algorithms for Random Generation and Counting: A Markov Chain Approach. Birkhäuser, Basel (1988)
14. Valiant, L.G.: The complexity of computing the permanent. Theoret. Comput. Sci. **8**(2), 189–201 (1979)