Approximation algorithms for counting the number of perfect matchings in bipartite graphs*

Abbas Mehrabian†

University of Waterloo

April 15, 2010

1 Introduction

The problem of devising an algorithm for counting the number of perfect matchings in bipartite graphs has a long history. Apparently the first algorithm (which works only in planar graphs) was presented in 1961 [8]. Then Valiant in 1979 showed that the problem is #P-complete, meaning that no polynomial-time algorithm exists for the problem, unless $P = NP$ [11]. After that, people focused on developing fast approximation algorithms for the problem. However, even developing a polynomial-time approximation algorithm appeared to be very difficult, as the first fully polynomial-time approximation scheme (see Section 3 for definition) for the problem was given in 2001 [6], and it uses a rather complicated Markov chain. In Table 1, I have collected some of the important results. For each result, the class of graphs in which the algorithm works and the running time are mentioned. Some notations need to be defined to read the table properly: the graph in question is a bipartite graph with two parts of size n, we say the graph is γ-dense if the degree of each vertex is at least γn, and the O^* notation hides the $\log n$ and ϵ^{-1} (where ϵ is the maximum acceptable error of the algorithm) factors.

In this report I will present the algorithm of Jerrum and Sinclair for $\frac{1}{2}$-dense graphs, and then provide a summary of the exponential algorithm of Jerrum and U. Vazirani. Section 2 is an introduction to Markov chains for those who have no previous background. If you have a basic knowledge about Markov chains you can skip Section 2. In Section 3 the terminology is introduced. In Section 4, which is the longest section, the first algorithm

*This is a report for the course CO 754: Approximation Algorithms, Winter 2010, University of Waterloo.

†amehrabi@uwaterloo.ca

author	class of graphs	running time	year	reference
Kasteleyn	planar graphs	$O(n^3)$	1961	[8]
Ryser	all graphs	$2^{O(n)}$	1963	[9]
Jerrum, Sinclair	$\frac{1}{2}$-dense graphs	$O^*(n^9)$	1989	[5]
Jerrum, U. Vazirani	all graphs	$2^{O(\sqrt{n})}$	1996	[7]
Jerrum, Sinclair, Vigoda	all graphs	$O^*(n^{10})$	2001	[6]
Bezáková,Štefankovič,V.Vazirani,Vigoda	all graphs	$O^*(n^7)$	2006	[1]
Huber	d-regular graphs	$O^*(n^{1.5+.5/d})$	2006	[3]
Huber, Law	γ-dense graphs	$O(n^{1.5+1/(4\gamma-2)})$	2008	[4]

Table 1: Summary of the important results

is presented with some detail, and an outline of the second algorithm can be found in Section 5. Finally, in Section 6, an application of the problem is discussed.

2 Markov Chains

Let $H = (V, E)$ be a directed graph with no multiple edges but loops are allowed. For a vertex $v \in V$, let $N^+(v)$ denote the set of outneighbors of v (u is an outneighbor of v if $vu \in E$). Also let $p : E \to (0, 1]$ be some function satisfying $\sum_{u \in N^+(v)} p(v, u) = 1$. A *random walk* on H is the following process, which occurs in a sequence of discrete *steps*: starting at vertex v_0, we proceed at the first step to an outneighbor u of v_0 with probability $p(v_0, u)$. Suppose we go to v_1 in first step. At the second step, we proceed to a randomly chosen outneighbor of v_1 (using the distribution induced by function p) and so on. The choice at each step is independent of all previous choices.

Markov chains is just the formalization of this concept. A *Markov chain* consists of a set of states S (corresponding to the set of vertices in the random walk) and a *transition matrix* P that has one row and one column for each state, and its (i, j)-th element, $P_{i,j}$, is the probability that the next state will be j, given that the current state is i (corresponds to the function p). Thus the entries of P are in $[0, 1]$ since they are probabilities, and for all i, $\sum_j P_{i,j} = 1$. In the following we will always be working with Markov chains with finite state space S, and $p(i, j)$ is just another notation for $P_{i,j}$. An important property of a Markov chain is the *memorylessness* property: the future behavior of the Markov chain depends only on its current state, and not on how it arrived at the present state. I will denote by X_t the state of the Markov chain at time t (corresponding to the vertex we are at, after step t). If we start at state i, the probability that we are at state j in time t is denoted by $p^t(i, j)$. That is, $p^t(i, j) = Pr[X_t = j | X_0 = i]$. From any Markov

chain $\mathcal{MC}$ we can build a directed graph H with vertex set $V(H) = S$ and a function $p : E(H) \to (0, 1]$ such that $\mathcal{MC}$ corresponds to a random walk on H. This graph is called the *underlying graph* of $\mathcal{MC}$.

Let us define the *distribution of the chain at time t* to be the row vector $q^t = (q_1^t, q_2^t, \ldots, q_n^t)$, where q_i^t is the probability that the chain is at state i in time t. Note that q^t depends on the initial distribution q^0. It is not hard to check that $q^{t+1} = q^t P$. A *stationary distribution* for $\mathcal{MC}$ is a probability distribution π such that $\pi = \pi P$. Intuitively, if the Markov chain is in the stationary distribution at time t, then it remains in the stationary distribution at time $t + 1$ (and in future as well).

A Markov chain is said to be *ergodic* if for any pair $u, v \in V(H)$, there is a number t_0 such that for all $t > t_0$ we have $p^t(u, v) > 0$. Ergodicity is an important property because of the following result (a proof may be found in most texts on stochastic processes):

Theorem 2.1. *[Fundamental Theorem of Markov Chains] Any finite and ergodic Markov chain has a unique stationary distribution π and for any initial distribution π_0 we have*

$$\lim_{t\to\infty} \pi_0 P^t = \pi,$$

where the convergence is pointwise.

There are two cases in which ergodicity of a finite Markov chain $\mathcal{MC}$ would fail, and it is evident why in those cases convergence does not happen in general. First, it may be the case that the underlying graph H of $\mathcal{MC}$ is not strongly conneceted. That is, there are two vertices $u, v \in V(H)$ such that there is no directed path from u to v. In this case, it clearly differs if we start from u or v, because if we start from u, we will never reach v (for all $t \geq 0$ we have $p^t(u, v) = 0$) but if we start from some other vertex, it may be possible that we reach v in future. Hence the theorem cannot be true in general for *every initial distribution*. In this case, we say that our chain is *reducible*. If the underlying graph is strongly connected we say that our chain is *irreducible*. Second, it may be the case that there are two vertices u, v and some positive integer T such that $p^t(u, v)$ is positive only for t's that are divisble by T. For example, if the underlying graph is bipartite, then for any two vertices u, v in the same part, $p^t(u, v)$ is zero for all odd t's. Hence the limit will not exist in general, since if we start from u, then $\left(\pi_0 P^{2k}\right)_v = Pr[X_{2k} = v | X_0 = u]$ is positive (and not tending to zero) while $\left(\pi_0 P^{2k+1}\right)_v = Pr[X_{2k+1} = v | X_0 = u]$ is zero. If this second bad case does not happen then we say that our chain is *aperiodic*.

Although the above theorem is very interesting theoretically, it would not be practical if we do not know the stationary distribution. Let us say that a Markov chain is *symmetric* if for any two states i, j we have $p(i, j) = p(j, i)$. Fortunately, if the chain is symmetric (which is the case in most applications) then its stationary distribution is uniform. Let

π^U denote the uniform distribution. That is, $(\pi^U)_i = 1/s$ for all states i, where s is the number of states. Then for each state j we have:

$$(\pi^U P)_j = \sum_{i=1}^{s} (\pi^U)_i P_{j,i} = \sum_{i=1}^{s} P_{i,j}/s = 1/s = (\pi^U)_j$$

3 Terminology

In this report, by *graph* I always mean a bipartite graph whose parts have equal size. Usually n denotes the size of each part. In the following I will assume that the graph in question has at least one perfect matching (this is easy to check in polynomial time). Let us say that G is γ-*dense* if the degree of every vertex is at least γn. And if G is 1/2-dense I just say G is *dense*. Let us denote by $M_k = M_k(G)$ the set of k-matchings (matchings k edges), and let $m_k = m_k(G) = |M_k(G)|$. So $m_1(G) = |E(G)|$ and the objective is to approximate $m_n(G)$. The set of all polynomially bounded functions in variables $x_1, \ldots, x_k$ is denoted by $\mathrm{poly}(x_1, \ldots, x_k)$.

Definition 1. For nonnegative real numbers $a, \hat{a}$ and $\epsilon \in [0, 1]$, I say that $\hat{a}$ *approximates* a *within ratio* $1 + \epsilon$ if

$$a/(1 + \epsilon) \leq \hat{a} \leq a \times (1 + \epsilon).$$

Remark 1. The following are easily derived from the definition:

1. If $\hat{a}$ approximates a within ratio $1 + \epsilon$ then $\hat{a}^{-1}$ approximates a^{-1} within the same ratio.

2. For any nonnegative real number N, if $\hat{a}$ approximates a within ratio $1 + \epsilon$ then $\hat{a}/N$ approximates a/N within the same ratio.

3. If $\hat{a}$ approximates a within ratio $1 + \epsilon$ and $\hat{b}$ approximates b within ratio $1 + \epsilon'$ then $\hat{a}\hat{b}$ approximates ab within ratio $(1 + \epsilon)(1 + \epsilon')$.

4. Let $a_i, \hat{a}_i, b_i, \hat{b}_i, N_i$ be nonnegative real numbers for $i = 1, 2, \ldots, n$, such that for all i, $\hat{a}_i/N_i$ and $\hat{b}_i/N_i$ approximate a_i/N_i and b_i/N_i within ratio $1 + \epsilon$, respectively. Then $\prod_{i=1}^{n} (\hat{a}_i/\hat{b}_i)$ approximates $\prod_{i=1}^{n} (a_i/b_i)$ within ratio $(1 + \epsilon)^{2n}$.

5. If a_3 approximates a_2 within ratio $1 + \epsilon_2$ and a_2 approximates a_1 within ratio $1 + \epsilon_1$ then a_3 approximates a_1 within ratio $(1 + \epsilon_1)(1 + \epsilon_2)$.

Definition 2. A *fully polynomial-time randomized approximation scheme (fpras)* for the problem is a randomized algorithm that given graph G and a parameter $\epsilon \in (0, 1]$ produces in time $poly(n, 1/\epsilon)$ a number $\hat{m}_n(G)$ such that

$$Pr\left[\hat{m}_n(G) \text{ approximates } m_n(G) \text{ within ratio } (1+\epsilon)\right] \geq 3/4.$$

Remark 2. It is easy to verify that given δ, by applying this algorithm $O(\log \delta^{-1})$ times and taking the median of the answers, we can increase this probability to $1 - \delta$.

4 The Algorithm For Dense Graphs

In this section I will present a fully polynomial-time randomized approximation scheme for dense graphs, which is based on Markov chains. The framework was proposed by Broder in 1986 [2] but the theorem proving the polynomial bound on the running time (Theorem 4.2) was proved by Jerrum and Sinclair in 1988 [5]. The monograph by Sinclair [10] contains more details and some further applications.

The idea is to approximate m_n via the product $m_n = \frac{m_n}{m_{n-1}} \times \frac{m_{n-1}}{m_{n-2}} \times \cdots \times \frac{m_2}{m_1} \times m_1$. We approximate each m_k/m_{k-1} using the idea of *sampling*. Suppose that we have a procedure to sample uniformly from the set $M_k \cup M_{k-1}$. If we take a set of samples, say R, from this set, then the ratio of the k-matchings of R to the ratio of $(k-1)$-matchings of R is an approximation for m_k/m_{k-1}. In the first subsection, I will make this idea precise and reduce the problem to *near-uniform sampling* from the set $M_n \cup M_{n-1}$. In the second subsection, I will show how the sampling is done using a Markov chain.

4.1 Sampling means approximate counting

Definition 3. Let S be a finite set. Let us say that algorithm $\mathcal{A}$ is a *near-uniform sampler with tolerance* $\epsilon \in [0, 1]$ *for* S if, in each run, $\mathcal{A}$ returns an element x of S such that for all $U \subseteq S$,

$$\frac{|Pr[x \in U] - |U|/|S||}{|U|/|S|} \leq \epsilon.$$

Lemma 4.1. *Let S be a finite set and $U \subseteq S$. Write $p = |U|/|S|$. Suppose that there exists a near-uniform sampler $\mathcal{A}$ for U with tolerance $\epsilon \in [0, 1]$. A natural method to approximate p is to run $\mathcal{A}$ N times and let X be proportion of the sample that belong to U. Then for any $\delta \in (0, 1]$, if $N \geq (27/p\epsilon^2)\ln(2/\delta)$ then X approximates p within ratio $1 + 5\epsilon$ with probability at least $1 - \delta$.*

Proof. For each sample x we have $\frac{|Pr[x \in U] - p|}{p} \leq \epsilon$. Hence

$$p(1+3\epsilon)^{-1} \leq p(1-\epsilon) \leq Pr[x \in U] \leq p(1+\epsilon) \leq p(1+3\epsilon),$$

and therefore $E[X]$ approximates p within ratio $1+3\epsilon$. If X approximates $E[X]$ within ratio $1+\epsilon/2$ then X approximates p within ratio $(1+\epsilon/2)(1+3\epsilon) \leq 1+5\epsilon$, so

$$\begin{aligned} Pr[X \textit{ approximates } p \textit{ within ratio } & 1+5\epsilon] \\ & \geq Pr[X \textit{ approximates } E[X] \textit{ within ratio } 1+\epsilon/2] \\ & \geq Pr[|X - E[X]| \leq p\epsilon/3] \\ & \geq 1 - 2\exp(-\epsilon^2 pN/27), \end{aligned}$$

where the last inequality follows from Chernoff's bounds. This last quantity is greater than $1-\delta$ if $N \geq (27/p\epsilon^2)\ln(2/\delta)$. □

Note that the number of samples depend on the parameter p. In our problem, this quantity is related to the ratio of k-matchings to $(k-1)$-matchings. The following lemma gives a bound on this.

Lemma 4.2. *For all $2 \leq k \leq n$ we have*

$$1/n^2 \leq \frac{m_k}{m_{k-1}} \leq n^2.$$

Proof. Any k-matching can be built by inserting an edge to some $(k-1)$-matching. Each $(k-1)$-matching has $(n-k+1)$ unmatched vertices in each part, thus at most $(n-k+1)^2$ edges can be inserted to build a k-matching. Therefore, the number of k-matchings is at most $(n-k+1)^2$ times the number of $(k-1)$-matchings, which proves the upper bound.

An *augmenting path* in a matching, is an odd path whose edges are alternately matching and non-matching, and its first and last edges are non-matching. For the lower bound, I first prove that every $(k-1)$-matching $\mathbf{M}$ has an augmenting path of length at most 3. Let V, V' be the bipartation of G, and $v \in V, v' \in V'$ be unmatched in $\mathbf{M}$. Let us denote by $N(u)$ the set of neighbors of a vertex u. If $N(v)$ has an unmatched vertex, then we find an augmenting path of length 1. Suppose this case does not happen. The set $N(v)$ has at least $n/2$ vertices, and all of them are matched with some vertex in V. Since $|N(v')| \geq n/2$, we can find an edge ab of matching, with $a \in N(v), b \in N(v')$. Thus (v, a, b, v') is an augmenting path of length 3 (see Figure 1).

Consider a path P of odd length in a matching $\mathbf{M}$ whose edges are alternately matching and non-matching, and whose first and last edges are matching edges. By *de-augmenting* P I mean building a new matching $\mathbf{M}' = \mathbf{M}\Delta P$, where Δ refers to the symmetric difference operation (see Figure 2). Now, from any k-matching we can build at most $k + 2\binom{k}{2} = k^2$ number of $(k-1)$-matchings by de-augmenting paths of length at most 3, and (by above discussion) we build all $(k-1)$-matchings in this way, hence $m_{k-1} \leq k^2 m_k \leq n^2 m_k$. □

We are in a good shape to state the main lemma of this subsection, which reduces the problem to near-uniform sampling from the set $M_n(G) \cup M_{n-1}(G)$.

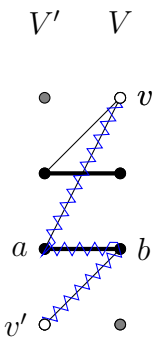

Figure 1: An augmenting path of length 3

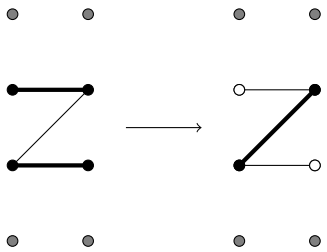

Figure 2: De-augmenting a path of length 3

Lemma 4.3. *Suppose that for any dense G and tolerance $\tau \in (0,1]$, there exists a near-uniform sampler $\mathcal{A}(G,\tau)$ for $M_n(G) \cup M_{n-1}(G)$, with tolerance τ and running time $poly(n, \log \tau^{-1})$. Then there is an fpras for counting the number of perfect matchings in a dense graph.*

Proof. Let G, ϵ be given. First, suppose that for each $2 \le k \le n$, we can approximate $r_k = m_k/m_{k-1}$ within ratio $(1+\epsilon/4n)^2$ with probability $(1-1/8n)^2$ in time $poly(n, 1/\epsilon)$. Let $\hat{r}_k$ be the estimate for r_k. Then we can compute the product $m_1\hat{r}_2\hat{r}_3\ldots\hat{r}_n$ in time $poly(n, 1/\epsilon)$, which approximates m_n within ratio $(1+\epsilon/4n)^{2n} \le 1+\epsilon$ with probability at least $(1-1/8n)^{2n} \ge 3/4$, and this completes the proof.

Next, I describe how to approximate r_k with the desired accuracy. Fix some $0 \le k \le n-2$. To compute the ratio m_{n-k}/m_{n-k-1} we do the following: Let V, V' be the parts of G. Build a new graph G_k by adding k new vertices to each of V, V' and draw all edges between the new vertices and the old vertices in opposite part. Formally, G_k has vertex set $V \cup \{u_1, \ldots, u_k\} \cup V' \cup \{u'_1, \ldots, u'_k\}$ and edge set $E(G) \cup \{u_i v' : 1 \le i \le k, v' \in V'\} \cup \{u'_i v : 1 \le i \le k, v \in V\}$ (see Figure 3 for an example with $n=3$, $k=1$). It is easy to check that G_k is dense.

If you consider a $(n+k)$-matching in G_k and remove the new vertices, then you obtain a $(n-k)$-matching in G. Moreover, any $(n-k)$-matching in G corresponds to exactly

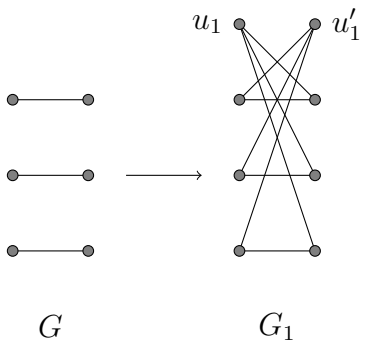

Figure 3: The graph G_1

$(k!)^2$ $(n+k)$-matchings in G_k. Similarly, if you consider a $(n+k-1)$-matching in G_k and remove the new vertices, then you obtain a $(n-k+r)$-matching in G, for some $r \in \{-1, 0, +1\}$. Hence, any matching $\mathbf{M} \in M_{n+k}(G_k) \cup M_{n+k-1}(G_k)$ is of one of the following types (see Figure 4 for an example with $n = 3$, $k = 1$):

1. $\mathbf{M} \in M_{n+k}(G_k)$ and $\mathbf{M}$ corresponds to a $(n-k)$-matching in G: there are $m_{n-k}(G)k!^2$ matchings of this type .

2. $\mathbf{M} \in M_{n+k-1}(G_k)$ and $\mathbf{M}$ corresponds to a $(n-k-1)$-matching in G: there are $m_{n-k-1}(G)k!^2(k+1)^2$ matchings of this type.

3. $\mathbf{M} \in M_{n+k-1}(G_k)$ and $\mathbf{M}$ corresponds to a $(n-k)$-matching in G: there are $m_{n-k}(G)(2k)k!^2$ matchings of this type.

4. $\mathbf{M} \in M_{n+k-1}(G_k)$ and $\mathbf{M}$ corresponds to a $(n-k+1)$-matching in G: there are $m_{n-k+1}(G)k!^2$ matchings of this type.

Let us denote by $T_1, \ldots, T_4$ the set of matchings of type $1, \ldots, 4$, repectively. Hence, if $\mathbf{M}$ is picked uniformly from $M_{n+k}(G_k) \cup M_{n+k-1}(G_k)$, then

$$Pr[\mathbf{M} \in T_1 \cup T_3] = \frac{(2k+1)m_{n-k}(G)}{(k+1)^2 m_{n-k-1}(G) + (2k+1)m_{n-k}(G) + m_{n-k+1}(G)}.$$

Let us denote the denominator of this fraction by N. By Lemma 4.2 (and a little calculation) we find $Pr[\mathbf{M} \in T_1 \cup T_3] \geq 1/3n^3$. Similarly, $Pr[\mathbf{M} \in T_2] = (k+1)^2 m_{n-k-1}(G)/N \geq 1/3n^3$. Recall that G_k is dense so we can sample near-uniformly from the set $M_{n+k}(G_k) \cup M_{n+k-1}(G_k)$. By Lemma 4.1, the number of samples needed to approximate $(2k+1)m_{n-k}(G)/N$ within ratio $1 + \epsilon/4n$ with probability $1 - 1/8n$ is $(1296n^5/\epsilon^2)\ln(16n) \in poly(n, \epsilon^{-1})$. Similarly, one can approximate the ratio $(k+1)^2 m_{n-k-1}(G)/N$ with the same parameters in $poly(n, \epsilon^{-1})$ time. Dividing these estimated values and multiplying

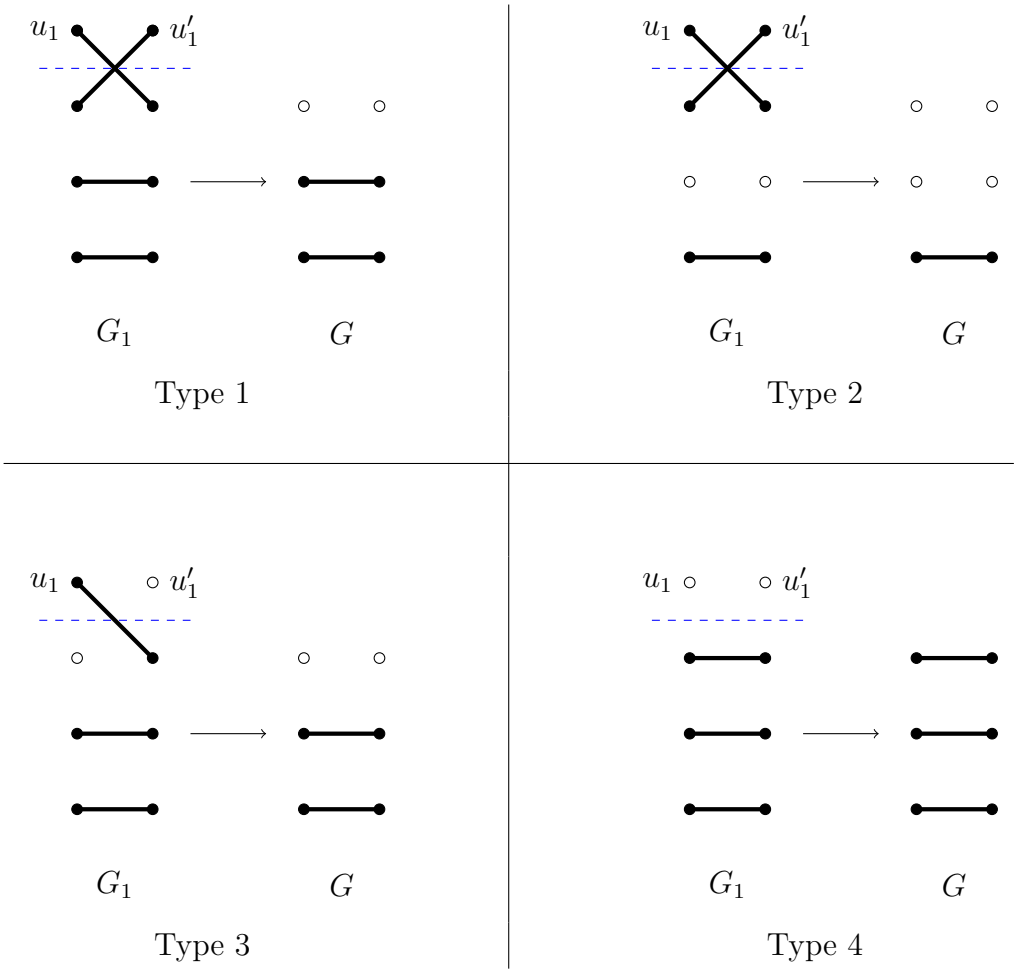

Figure 4: The four types of matchings of $M_4(G_1) \cup M_3(G_1)$

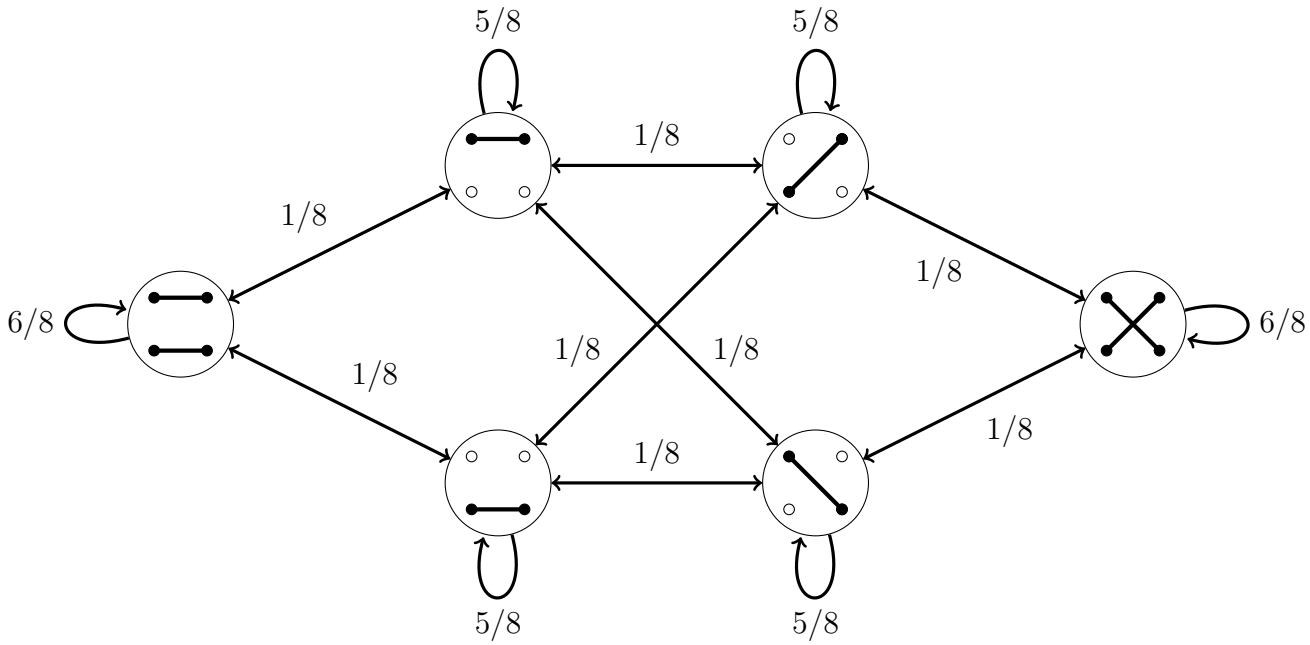

Figure 5: The Markov chain associated with $G = K_{2,2}$

by $(k+1)^2/(2k+1)$ gives with probability $(1-1/8n)^2$ an approximation for m_{n-k}/m_{n-k-1} within ratio $(1+\epsilon/4n)^2$. Note that each sampling is done in time $poly(n, \log \epsilon^{-1})$, so the proof is complete. □

4.2 Sampling via a Markov chain

Definition 4. For a graph G, the Markov chain $\mathcal{MC} = \mathcal{MC}(G)$ has $M_n(G) \cup M_{n-1}(G)$ as its state space, and the transitions are defined as follows: Suppose that we are in state $\mathbf{M}$. Then, with probability 1/2 we remain at $\mathbf{M}$ (this is just to ensure aperiodicity). With probability 1/2 we choose an edge $e = uv$ of G, uniformly at random, and then:

- If $\mathbf{M} \in M_n$ and $e \in \mathbf{M}$, move to $\mathbf{M}' = \mathbf{M} - e$.
- If $\mathbf{M} \in M_{n-1}$ and u, v are unmatched in $\mathbf{M}$, move to $\mathbf{M}' = \mathbf{M} + e$.
- If $\mathbf{M} \in M_{n-1}$, u is matched to w and v is unmatched, move to $\mathbf{M}' = \mathbf{M} + e - uw$.
- Otherwise, remain at $\mathbf{M}$.

We denote by $p(\mathbf{M}, \mathbf{M}')$ the probability of transition from $\mathbf{M}$ to $\mathbf{M}'$.

In Figure 5, the Markov chain $\mathcal{MC}(K_{2,2})$ is illustrated.

Remark 3. The Markov chain $\mathcal{MC}$ is ergodic and its stationary distribution is uniform.

Proof. The self-loops guarantee that the chain is aperiodic. Irreducibility is easy to check. For all $\mathbf{M} \neq \mathbf{M}'$, if we can go from $\mathbf{M}$ to $\mathbf{M}'$ in a single step then $p(\mathbf{M}, \mathbf{M}') = p(\mathbf{M}', \mathbf{M}) = 1/2|E(G)|$, otherwise $p(\mathbf{M}, \mathbf{M}') = p(\mathbf{M}', \mathbf{M}) = 0$. Hence the transition probabilites are symmetric, and the uniform distribution is a stationary distribution. □

Having the above proposition in hand, to near-uniformly sample a matching from $M_n(G) \cup M_{n-1}(G)$, one can start from an arbitrary state and simulate $\mathcal{MC}(G)$ a large number of steps, and then output the final state. By the fundamental theorem of Markov chains (Theorem 2.1) the distribution of final state is close to the stationary distribution, i.e. it is close to the uniform distribution. This method of sampling is called the *Markov Chain Monte Carlo (MCMC)* method. But how fast is the convergence? In other words, how long should we simulate the Markov chain to be sure that the final distribution is sufficiently close to the uniform distribution? The lemma that comes next answers this question, but we need a definition first.

Definition 5. Let π denote the stationary distribution of a Markov chain $\mathcal{MC}$. For a subset A of states of $\mathcal{MC}$, the *conductance* of A is defined as

$$\Phi(A) = \frac{\sum_{ij \in \delta(A)} \pi_i p(i,j)}{\sum_{i \in A} \pi_i},$$

where $\delta(A) = \{ij : i \in A, j \notin A\}$. Note that this is just the conditional probability that the stationary distribution *escapes* from A in a single step, given that it is initially in A. The conductance of $\mathcal{MC}$, $\Phi(\mathcal{MC})$, is defined as the minimum of conductances of subsets A with $0 < \sum_{i \in A} \pi_i \leq 1/2$.

Lemma 4.4. *[10] Let $\mathcal{MC}$ be an ergodic symmetric Markov chain with s states and such that $p(i,i) \geq 1/2$ for all states i, and let $\epsilon \in (0,1]$. Then the minimum t such that the distribution at time t is a near-uniform distribution with tolerance ϵ is at most $\frac{2}{\Phi(\mathcal{MC})^2}(\ln s + \ln \epsilon^{-1})$.*

Hence in order to bound the number of steps required to sampling, we only need to bound the conductance of the Markov chain, which appears to have a strong relation with the structure of the underlying graph. The following theorem is the main result of [6]:

Theorem 4.1. *If G is dense then $\Phi(\mathcal{MC}(G)) \geq 1/12n^6$.*

Proof.(outline) Let s be the number of states of $\mathcal{MC}(G)$ (recall that each state is a matching of G), and m be the number of edges of G. Since the stationary distribution is uniform, we have $\pi(\mathbf{M}) = 1/s$ for all states $\mathbf{M}$. Let H be the underlying graph of

$\mathcal{MC}(G)$. By the way the Markov chain is defined, it is not hard to check that for all $(\mathbf{M}, \mathbf{M}') \in E(H)$ we have $p(\mathbf{M}, \mathbf{M}') = 1/2m$. Thus the formula for the conductance can be simplified as:

$$\Phi\left(\mathcal{MC}(G)\right) = \frac{1}{2m} \min_{0<|A|\leq s/2} \frac{|\delta(A)|}{|A|}.$$

Therefore, we actually need to prove that the underlying graph H has good edge expansion.

The technique used here is called the *canonical paths* technique: suppose that we can introduce, for each pair $\mathbf{M}, \mathbf{M}'$ of vertices of H, a simple $\mathbf{M}, \mathbf{M}'$-path, which is called the canonical $\mathbf{M}, \mathbf{M}'$-path. If we can prove that every edge is contained in at most $3sn^4$ canonical paths, then for any $A \subseteq S$ with $|A| \leq s/2$, the number of canonical paths that cross the cut $\delta(S)$ is $|A|(s - |A|) \geq |A|s/2$. Thus for any such A the number of cut edges must be at least $\frac{|A|s/2}{3sn^4} = |A|/6n^4$, which gives

$$\Phi(A) \geq \frac{1}{2m} \frac{|A|/6n^4}{|A|} \geq 1/12n^6.$$

I will not go into the details of specifying the canonical paths and proving the low containment property here. □

We conclude that for any dense G and tolerance $\epsilon \in (0, 1]$, there exists a near-uniform sampler for $M_n(G) \cup M_{n-1}(G)$ with tolerance ϵ and running time $\text{poly}(n, \log \epsilon^{-1})$. This, together with Lemma 4.3 prove the main result of this section:

Theorem 4.2. *There exists an fpras for counting the number of perfect matchings in a dense graph.*

In their paper [5] Jerrum and Sinclair proved something stronger; they proved that this algorithm is efficient for all graphs in which the ratio m_{n-1}/m_n is polynomially bounded. This seems to be a very weak requirement. Indeed it is much weaker than the denseness requirement. Nevertheless, there exist graphs not satisfying it. The graph in Figure 6 is an example. To see this, note that there is only perfect matching. But if we remove u, v, the resulting graph has an exponential number of perfect matchings, since there are two ways to match the vertices in each hexagon. Therefore, there are an exponential number of matchings of size $n - 1$. Hence the ratio m_{n-1}/m_n is exponentially large.

5 Outline of The Algorithm For General Graphs

In this section I will outline a summary of the algorithm of [7], whose running time is $\exp\left(O(\sqrt{n})\right)$. This algorithm uses the algorithm of Section 4 as a subroutine. Let us introduce an important concept before going into more details.

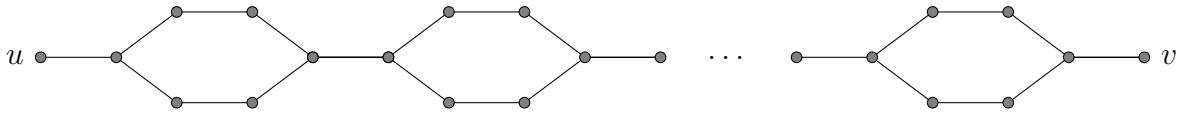

Figure 6: A graph with m_{n-1}/m_n exponentially large

Definition 6. We say that G is *α-vertex-expander* if for any subset A of vertices, which lies completely in one of the parts and has size at most $n/2$, the number of vertices that are adjacent to some vertex of A is at least $(1+\alpha)|A|$.

Let us denote by $N(A)$ the set of vertices that are adjacent to some vertex of A. Also set V, V' be the bipartation of G. If G is α-vertex-expander then we can bound the ratio m_{n-1}/m_n using a very similar technique to the one we used in Lemma 4.2:

Lemma 5.1. *[7] If G is α-vertex-expander then*

$$\frac{m_{n-1}(G)}{m_n(G)} \leq \exp\left(O(\log^2 n/\alpha)\right).$$

Now, if a given G is α-vertex-expander, then the algorithm of Section 4 applied on G has running time $\exp\left(O(\log^2 n/\alpha)\right)$ (a nice thing about exponential functions is that $\mathrm{poly}\left(\exp\left(O\left(f(n)\right)\right)\right)$ and $\exp\left(O\left(f(n)\right)\right)$ are not different). And if G is not an α-vertex-expander, then without loss of generality, there exists a set $A \subseteq V$ with $|N(A)| < (1+\alpha)|A|$. Note that in any perfect matching, vertices of A should be matched to some $|A|$ vertices of $N(A)$. For each subset B of $N(A)$ of size $|A|$, we count the number of perfect matchings in the graphs G_1 induced by $A \cup B$ and G_2 induced by $(V-A) \cup (V'-B)$, multiply these two quantities, and then add up these numbers for all such subsets B. The bound for running time of the algorithm uses the fact that the number of such B's is bounded (is at most $\binom{(1+\alpha)n/2}{n/2}$). There are two difficult steps in the algorithm, which I describe next.

First, how to detect whether G is α-vertex-expander, or find a set A with few neighbors if it is not? The idea is to use *tight sets*. A set $T \subseteq V$ or $T \subseteq V'$ is called tight if $|N(T)| = |T|$. The collection of tight sets is closed under union and intersection; in particular, for each $v \in V \cup V'$ there exists a unique *smallest* tight set containing v, denoted by $\Delta(v)$. It is not hard to check that the sets $\Delta(v)$ are easy to compute in polynomial time (using augmenting paths). Using the idea of tight sets, (and merging them in certain occasions) one can present a procedure that either decides correctly that G is α-vertex-expander or produces a set $A \subseteq V$ or $A \subseteq V'$ such that $|N(A)| < (1+2\alpha)|A|$, and runs in time $\exp\left(O(\alpha n \log n)\right)$ (see Lemma 4 of [7] for details). Notice that the set

A does not necessarily satisfy $|N(A)| < (1+\alpha)|A|$, but the bound $1+2\alpha$ suffices for our needs.

Second, how to run the algorithm of Section 4 on the generated subproblems with a correct parameter ϵ and how to select an appropriate α to obtain a tight approximation with high probability? These details are described in Theorem 5 of [7], where after setting the parameters very carefully and after two pages of calculation, in is shown that the algorithm is both fast and accurate. The total running time is $O(n^3/\epsilon^2)\exp(O(\sqrt{n}\log^2 n))$, which, ignoring the ϵ and polynomial factors, is $\exp(O(\sqrt{n}))$.

6 Application: Approximating the Permanent

Definition 7. The *permanent* of an $n \times n$ matrix $A = [a_{i,j}]$ is defined as

$$\mathrm{per}(A) = \sum_{\sigma} \prod_{i=1}^{n} a_{i,\sigma(i)},$$

where the sum is over all permutations σ of $\{1, 2, \ldots, n\}$.

If A is a 0,1-matrix, then one can build a graph G_A such that the number of perfect matchings of G_A is equal to the permanent of A. The graph G_A has vertex set $V(G_A) = \{u_1, \ldots, u_n, u'_1, \ldots, u'_n\}$ and edge set $E(G_A) = \{u_i u'_j : a_{i,j} = 1\}$. In order to see this, note that the product $\prod_{i=1}^{n} a_{i,\sigma(i)}$ is 1 if and only if the set $\{\{u_1, u'_{\sigma(1)}\}, \{u_2, u'_{\sigma(2)}\}, \ldots, \{u_n, u'_{\sigma(n)}\}\}$ is a perfect matching of G_A, and is 0 otherwise. Therefore, the algorithm of Section 4 gives an fpras for evaluating the permanent of a 0,1-matrix in which the sum of every row and column is at least $n/2$. Moreover, the algorithm of Section 5 gives a $(1+\epsilon)$-approximation algorithm with running time $\exp(O(\sqrt{n}))$ for evaluating the permanent of a 0,1-matrix with high probability.

In 2001, Jerrum et al. gave an fpras for counting the number of perfect matchings in all graphs [6]. This immediately gives an fpras for calculating the permanent of any 0,1-matrix. Moreover, in the same paper, they showed how their algorithm can be turned into an fpras for evaluating the permanent of a matrix with arbitrary nonnegative entries. They further proved that no fpras exists for estimating the permanent of a general matrix, unless $P = NP$.

References

[1] I. Bezáková, D. Štefankovič, V. V. Vazirani and E. Vigoda, Accelerating simulated annealing for the permanent and combinatorial counting problems, SODA'06: Proc. 17th ACM-SIAM Sympos. on Discrete Algorithms (2006), 900–907.

[2] A. Z. Broder, How hard is it to marry at random? (On the approximation of the permanent), STOC'86: Proc. 18th ACM Sympos. on Theory of Computing (1986), 50–58; Erratum, STOC'88: Proc. 20th ACM Sympos. on Theory of Computing (1988), 551.

[3] M. Huber, Exact sampling from perfect matchings of dense regular bipartite graphs, Algorithmica 44 (2006), 183–193.

[4] M. Huber and J. Law, Fast approximation of the permanent for very dense graphs, SODA'08: Proc. 19th ACM-SIAM Sympos. on Discrete Algorithms (2008), 681–689.

[5] M. Jerrum and A. Sinclair, Approximating the permanent, SIAM J. Comput. 18 (1989), 1149–1178.

[6] M. Jerrum, A. Sinclair and E. Vigoda, A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries, J. ACM 51 (2001), 671–697.

[7] M. Jerrum and U. Vazirani, A mildly approximation algorithm for the permanent, Algorithmica 16 (1996), 392–401.

[8] P. W. Kasteleyn, The statistics of dimers on a lattice, I, the number of dimer arrangements on a quadratic lattice, Physica 27 (1961), 1664–1672.

[9] H. J. Ryser, Combinatorial Mathematics, The Carus Mathematical Monographs No. 14, Mathematical Association of America, 1963.

[10] A. Sinclair, Algorithms For Random Generation and Counting: a Markov Chain Approach, Birkhäuser, Boston, 1993.

[11] L. G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1979), 189–201.