

Maximum Matchings via Glauber Dynamics

Anant Jindal *

Gazal Kochar †

Manjish Pal ‡

Abstract

In this paper we study the classic problem of computing a maximum cardinality matching in general graphs $G = (V, E)$. This problem has been studied extensively more than four decades. The best known algorithm for this problem till date runs in $O(m\sqrt{n})$ time due to Micali and Vazirani [24]. Even for general bipartite graphs this is the best known running time (the algorithm of Karp and Hopcroft [16] also achieves this bound). For regular bipartite graphs one can achieve an $O(m)$ time which, following a series of papers, has been recently improved to $O(n \log n)$ by Goel, Kapralov and Khanna (STOC 2010) [15]. In this paper we present a randomized algorithm based on the Markov Chain Monte Carlo paradigm which runs in $O(m \log^2 n)$ time, thereby obtaining a significant improvement over [24].

We use a Markov chain similar to the *hard-core model* for Glauber Dynamics with *fugacity* parameter λ , which is used to sample independent sets in a graph from the Gibbs Distribution [31], to design a faster algorithm for finding maximum matchings in general graphs. Motivated by results which show that in the hard-core model one can prove fast mixing times (for e.g. it is known that for λ less than a critical threshold the mixing time of the hard-core model is $O(n \log n)$ [27]), we define an analogous Markov chain (depending upon a parameter λ) on the space of all possible partial matchings of a given graph G , for which the probability of a particular matching M in the stationary follows the Gibbs distribution which is:

$$\pi(M) = \frac{\lambda^{|M|}}{\sum_{x \in \Omega} \lambda^{|x|}}$$

where Ω is the set of all possible matchings in G .

We prove upper and lower bounds on the mixing time of this Markov chain. Although our Markov chain is essentially a simple modification of the one used for sampling independent sets from the Gibbs distribution, their properties are quite different. Our result crucially relies on the fact that the mixing time of our Markov Chain is independent of λ , a significant deviation from the recent series of works [11, 26, 28, 29, 30] which achieve computational transition (for estimating the partition function) on a threshold value of λ . As a result we are able to design a randomized algorithm which runs in $O(m \log^2 n)$ time that provides a major improvement over the running time of the algorithm due to Micali and Vazirani. Using the conductance bound, we also prove that mixing takes $\Omega(\frac{m}{k})$ time where k is the size of the maximum matching.

*Laxmi Niwas Mittal Institute of Information Technology, India. anantjindal1@gmail.com

†Laxmi Niwas Mittal Institute of Information Technology, India. gkochar@gmail.com

‡Indian Institute of Technology Gandhinagar, India. manjish_pal@iitgn.ac.in

1 Introduction

Given an unweighted undirected graph $G = (V, E)$ with $|E| = m$ and $|V| = n$, a matching M is a set of edges belonging to E such that no two edges in M are incident on a vertex. If there is a matching of size $n/2$ (for n even), then it is called a *perfect matching*. The Maximum Matching problem is to find the maximum sized matching in a given graph. The computational complexity of this problem has been studied extensively for more than four decades starting with an algorithm of Edmonds.

1.1 General Graphs

Edmonds's celebrated paper 'Paths, Trees and Flowers' [9] was the first to give an efficient algorithm (also called the *blossom shrinking algorithm*) for finding maximum matching in general graphs. This algorithm can be implemented in $O(n^4)$ time. The running time was subsequently improved in a number of papers [10, 20, 22]. All these papers were variants of Edmonds algorithm. Even and Kariv [12] obtained an improvement to $O(n^{2.5})$ which was improved by Micali and Vazirani [24] who gave an $O(m\sqrt{n})$ time algorithm for the problem by a careful handling of blossoms. This is the best known algorithm for finding maximum matchings in general bipartite graphs.

1.2 Bipartite Graphs

For bipartite graphs, the problem can easily be solved using the max-flow algorithm by Ford and Fulkerson, an algorithm usually taught in an undergraduate algorithms course [21], which has a running time of $O(mn)$. The first algorithm for this problem was given by Konig [19]. Hopcroft and Karp [16] gave an algorithm that runs in $O(m\sqrt{n})$ time. This algorithm is an exact and deterministic algorithm. The problem becomes significantly simpler for regular bipartite graphs. In a *d-regular* bipartite graph every vertex has degree d . When d is a power of 2, Gabor and Kariv were able to achieve an $O(m)$ algorithm. After significant efforts, the ideas used there were used by Cole, Ost and Schirra [4] to obtain a get an $O(m)$ algorithm for general d .

In a recent line of attack by Goel, Kapralov and Khanna [13, 14], the authors were able to use sampling based methods to get improved running time. In the most recent paper they were able to achieve a running time of $O(n \log n)$ for d -regular graphs [15]. Their algorithm performs an appropriately truncated random-walk on a modified graph to successively find augmenting path.

2 Our Results

In this paper we give a Markov Chain Monte Carlo algorithm for finding a maximum matching in general bipartite graphs. Our algorithm is in the spirit similar to [15] which also is a 'truncated random walk' based algorithm, however the stationary distribution of the underlying Markov Chains in their case is different from ours. Inspired from the hard-core model with fugacity parameter λ of sampling independent sets from graphs we define a similar Markov Chain over the space of all possible partial matching such that its stationary distribution $\pi(\cdot)$

is the Gibbs distribution, ie. given a matching M its probability π is

$$\pi(M) = \frac{\lambda^{|M|}}{\sum_{\sigma \in \Omega} \lambda^{|\sigma|}}$$

where Ω is the set of all possible matchings in G . Notice that $\pi(M)$ is maximum for maximum matchings and if λ is a significantly large number, $\pi(M)$ tends to 1 for maximum matchings. Our algorithm is extremely simple (being a standard in the MCMC paradigm). Starting from a fixed matching we start a random walk in Ω according to the underlying graph \tilde{G} of the Markov chain. After T_{mix} (mixing time) steps the distribution reached by the algorithm is roughly the same as the Gibbs distribution. More formally, the variation distance of \mathcal{D}^t (the distribution after t steps) from the Gibbs distribution is less than $\frac{1}{2e}$. This just leaves the task of proving an upper-bound on the mixing time of the Markov Chain, for which we resort to the Coupling Method introduced by Bubley and Dyer [2]. We define a metric $\Phi(\cdot, \cdot)$ and apply the bound from [2], to show that the Markov Chain mixes in $O(m \log n)$ time. We also use the conductance method to show that the mixing time will be at least $\Omega(\frac{m}{k})$ where k is the size of the maximum matching. Thus upto logarithmic factors the bounds are same when k is small (at most polylogarithmic). The main result of our paper can be concisely written as follows:

Theorem 1 (Main). *There exists a randomized algorithm which given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$ finds a maximum matching in $O(m \log^2 n)$ time with high probability.*

2.1 Organization

The paper is organized as follows: in Section 3 we give a brief description of the basic idea and technique that are underlying our algorithm. Subsequently in Section 4 we give an overview of the MCMC paradigm which includes basic preliminaries and definitions regarding Markov Chain and Mixing. Section 7 is devoted to the details of the chain being used by us and proving that indeed it has the desired properties. We then prove upper and lower bounds on its mixing time using the coupling method and conductance argument in Section 8 and Section 9 respectively. We end the paper with a conclusion and some open problems.

3 The Idea: Maximum Matchings via Glauber Dynamics

Our ideas are inspired mainly from the results in the Glauber dynamics of the hard-core model of fugacity parameter λ , for sampling independent sets from the Gibbs distribution. According to the Gibbs distribution the probability of an independent set I is given by

$$\mathcal{G}(I) = \frac{\lambda^{|I|}}{\sum_{\rho \in \Omega} \lambda^{|\rho|}}$$

where Ω is the set of all possible independent sets in G and $\mathcal{Z} = \sum_{\rho \in \Omega} \lambda^{|\rho|}$ is also called the *partition function*. Clearly for $\lambda = 1$ the partition function value is same as the number of independent sets in the graph (computing which is a #P-Hard problem).

In the hard-core model given a particular configuration σ of an independent set (σ can be thought of an n -dimensional 0/1 vector which is 1 for all the vertices which are in the independent set and 0 otherwise), we choose a vertex randomly uniformly, if this vertex is already

present in the independent set we keep it with probability $\frac{\lambda}{1+\lambda}$ and discard it with probability $\frac{1}{1+\lambda}$ otherwise the vertex is not in the independent set and if this vertex can be added to the independent set (i.e. none of its neighbors are already present in the independent set) then again it is added with probability $\frac{\lambda}{1+\lambda}$ and rejected with probability $\frac{1}{1+\lambda}$. The beauty of this Markov Chain is that the stationary distribution is the Gibbs distribution. The details of this chain can be found in [31].

The key difference between the mixing time of Glauber Dynamics for independent sets and our case is that one can achieve fast mixing time in the former case only for small values of λ . In fact intuitively one should not be able to obtain fast mixing times for large values of λ because such a result would imply that we can design a randomized polynomial time algorithm for finding maximum independent set in a graph, which is an NP-Hard problem. This intuition has led to a series of papers [11, 26, 28, 29, 30] which ultimately has been successful in proving that there exists a threshold value of $\lambda = \lambda_c$ such that if $\lambda > \lambda_c$ then estimating the partition function is hard (the exact technical condition is that unless $NP = RP$ no FPRAS exists for estimating \mathcal{Z}) and for $\lambda < \lambda_c$ one can obtain an FPTAS for the same problem. Previous to this result, the computational complexity of estimating the partition function (and counting the number of independent sets) was only understood for special graphs [5, 32]. Apart from these results substantial attention has been given to obtain good bounds on the mixing time of this chain for trees [28].

We define a Markov Chain which is tuned to our need. In our case σ is a set of edges which form a partial matching. We make a simple modification to the above chain, wherein instead of picking a random vertex we pick a random edge $e_r \in E$ and perform the same experiment with the parameter λ as in the case of independent sets (notice that the chosen edge won't be added if in the present matching there is an edge sharing an end point with e_r). We then show that this chain is aperiodic and irreducible with stationary as the Gibbs distribution over the space of all possible partial matchings with parameter λ . We then use the techniques of bounding the mixing time to achieve a λ independent upper bound. This remarkable property allows us to exploit the nature of Gibbs distribution (which we obtain for very large values of λ) without getting an overhead on the mixing time.

3.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo algorithms have played a significant role in statistics, econometrics, physics and computing science over the last two decades. For some high-dimensional problems in geometry, such as computing the volume of a convex body in d dimensions, MCMC simulation is the only known general approach for providing a solution within time polynomial in d [6]. For a number of other hard problems like approximating the permanent [18], approximate counting [17], the only known FPRASs (*Fully Polynomial time Randomized Approximation Schemes*) rely on the MCMC paradigm. In this paper, we use this method to obtain a faster algorithm for the classical problem of finding maximum matchings in general graphs, a problem which is known to be solvable in polynomial time.

The Markov Chain Monte Carlo (MCMC) method is a simple and frequently used approach for sampling from the Gibbs distribution of a statistical mechanical system. The idea goes like this, we design a Markov chain whose state space is Ω whose stationary distribution is the desired Gibbs distribution. Starting at an arbitrary state, we simulate the Markov chain on Ω until it is sufficiently close to its stationary distribution. We then output the final state which is a sample from (close to) the desired distribution. The required length of the simulation, in

order to get close to the stationary distribution, is traditionally referred to as the mixing time τ or T_{mix} and the aim is to bound the mixing time to ensure that the simulation is efficient. For a detailed understanding of the theory of Markov Chains we would recommend the recent excellent book by Levin, Peres and Wilmer [23].

4 Preliminaries

4.1 Markov chains

Consider a stochastic process $(X_t)_{t=0}^{\infty}$ on a finite state space Ω . Let P denote a non-negative matrix of size $|\Omega| \times |\Omega|$ which satisfies

$$\sum_{j \in \Omega} P_{ij} = 1 \quad \forall i \in \Omega$$

The process is called a Markov chain if for all times t and $i, j \in \Omega$ probability of going from i^{th} state to j^{th} state is independent of the path by which i^{th} state is reached i.e. if X_t is the state of the process at time t then

$$P[X_{t+1}|X_t = x_t, X_{t-1} = x_{t-1} \dots X_0 = x_0] = P[X_{t+1}|X_t = x_t]$$

A distribution π is called a *stationary distribution* if it satisfies $\pi P = \pi$. A necessary and sufficient condition for a chain to have a unique stationary distribution is that the chain is

1. *Irreducible*: for all $i, j \in \Omega$ there exists a time t such that $P_{ij}^t > 0$; and
2. *Aperiodic*: for all $i \in \Omega$, $\text{GCD} \{t : P_{ii}^t > 0\} = 1$.

A Markov Chain which has both of the above properties is called *ergodic*. For an ergodic Markov chain, if a distribution π satisfies the detailed balance equations

$$\pi_i P_{ij} = \pi_j P_{ji}$$

for all $i, j \in \Omega$ then π is the (unique) stationary distribution and such a chain is called *reversible*.

4.2 Mixing Time

The notion of mixing time is defined as a way to measure the closeness of the distribution after t steps w.r.t. the stationary distribution. The total variation distance between two discrete probability distributions over a finite space Ω is defined as the half of the l_1 norm of the corresponding probability vectors.

$$d_{TV}(\mu, \nu) = \frac{1}{2} \sum_{\omega \in \Omega} |\mu(\omega) - \nu(\omega)|$$

If P_t is the probability distribution after t steps in the random walk then T_{mix} is the minimum t for which,

$$d_{TV}(P_t, \pi) \leq \frac{1}{2e}$$

where π is the stationary distribution. Therefore, if we intend to get close to a stationary distribution we just have to truncate the random walk on the state space after τ steps.

4.3 Conductance

The conductance of a Markov chain is defined as the following quantity,

$$\phi(G) = \min_{S \subset V} \frac{\sum_{i \in S, j \in \bar{S}} \pi_i P_{ij}}{(\sum_{i \in S} \pi_i)(\sum_{i \in \bar{S}} \pi_i)}$$

Another quantity of our interest here is T_{relax} , the *relaxation time* of the Markov chain. T_{relax} is defined as the inverse of conductance i.e. $T_{relax} = \frac{1}{\phi(G)}$. It is known that T_{mix} and T_{relax} obey the following inequality [1].

$$T_{relax} + 1 \leq T_{mix}.$$

Also it is known [28] that

$$T_{relax} = \Omega\left(\frac{1}{\phi}\right),$$

a bound usually used to prove lower bounds on the Mixing time of Markov Chains.

5 The Chain

In this section, we describe the chain considered by our algorithm which is essentially a modification of the hard-core model of Glauber Dynamics. Recall that our objective is to come up with a chain whose stationary distribution ensures that the probability of being at a maximum matching is the largest. Recall that first we need to ensure that our chain is aperiodic and irreducible.

We use the following natural modification of the Markov chain for the hard-core model of Glauber dynamics.

- Choose an edge e_r uniformly at random from E .
- Let

$$\sigma' = \begin{cases} \sigma \cup \{e_r\}, & \text{with probability } \frac{\lambda}{1+\lambda} \\ \sigma \setminus \{e_r\}, & \text{with probability } \frac{1}{1+\lambda} \end{cases}$$

- If σ' is a valid matching, move to state σ' otherwise remain at state σ .

We are now prepared to prove that this is a valid Markov Chain with stationary as the Gibbs distribution, where we define Gibbs distribution as the following distribution over the set of all possible matchings Ω as

$$\mathcal{G}(M) = \frac{\lambda^{|M|}}{\sum_{x \in \Omega} \lambda^{|x|}}$$

In the rest, we will call $Z = \sum_{x \in \Omega} \lambda^{|x|}$.

Lemma 1. X_t is ergodic with stationary distribution as \mathcal{G} .

Proof. Since for every state there is some probability by which the walk can remain in the same state, the chain is aperiodic. Also the underlying graph is connected because given any matching there is a at least one path to reach any other matching (consider the path that first drops all the edges of the initial matching and adds the edges of the new matching one by one).

To show that the stationary of this distribution is \mathcal{G} , we would show that the chain is reversible w.r.t. \mathcal{G} . Consider two distinct states i and j in the Markov Chain. Assume w.l.o.g that there is an edge from i to j (if there is no edge from i to j then the balance equations corresponding to i and j are trivially satisfied). Let the configuration i has t edges in it then according to our construction, j will have either $t+1$ or $t-1$ states (it won't be t because we have assumed that the two states are distinct). We will just look at the case when j has $t+1$ edges, the other case is analogous. Since we are interested in showing reversibility w.r.t. \mathcal{G} , $\pi_i = \frac{\lambda^t}{Z}$ and $\pi_j = \frac{\lambda^{t+1}}{Z}$. Therefore, $\pi_i P_{ij} = \frac{\lambda^t}{Z} \cdot \frac{\lambda}{m(1+\lambda)}$, and $\pi_j P_{ji} = \frac{\lambda^{t+1}}{Z} \cdot \frac{1}{m(1+\lambda)}$. Thus $\pi_i P_{ij} = \pi_j P_{ji} = \frac{\lambda^{t+1}}{mZ(1+\lambda)}$ \square

6 Upper Bound using Coupling

In this section we prove an upper bound on T_{mix} of the chain defined in the previous section. Our bound is based on the coupling argument introduced by Bubley and Dyer [2]. We first give a description of the coupling method.

6.1 Coupling Method

A coupling of a Markov chain on state space is a stochastic process (σ_t, η_t) on $|\Omega| \times |\Omega|$ such that:

- σ_t and η_t are copies of the original Markov chain and
- if $\sigma_t = \eta_t$, then $\sigma_{t+1} = \eta_{t+1}$

Thus the chains follow each other after the first instant when they hit each other. In order to measure the distance between the two copies of the chain, one introduces a distance function Φ on the product state space $\Omega \times \Omega$ so that $\Phi = \Phi(\sigma_t, \eta_t) = 0 \iff \sigma_t = \eta_t$. For two states σ and η let $\rho(\sigma, \eta)$ be the set of all paths from σ to η in the Markov Chain. The following theorem due to to Bubley and Dyer is used to prove mixing time on Markov chain.

Theorem 2. *Let Φ be an integer-valued metric defined on $\Omega \times \Omega$ which takes values in $\{0, 1, \dots, D\}$ such that, for all $\sigma, \eta \in \Omega$ there exists a path $\xi \in \rho(\sigma, \eta)$ with*

$$\Phi(\sigma, \eta) = \sum_i \Phi(\xi^i, \xi^{i+1})$$

Suppose there exists a constant $\beta < 1$ and a coupling (σ_t, η_t) of the Markov chain such that, for all σ_t, η_t ,

$$E[\Phi(\sigma_{t+1}, \eta_{t+1})] \leq \beta \Phi(\sigma_t, \eta_t)$$

Then the mixing time is bounded by

$$\tau \leq \frac{\log(2eD)}{1 - \beta}$$

Proof. Can be found in [2, 31]. \square

In order to bound the mixing time, we will define a coupling so as to minimize the time until both copies of the Markov chain reach the same state, and we will do that by defining the coupling in such a way that on every step both markov chains reach towards same state. The aim is to prove a good upper bound on $E[\Phi(\sigma_{t+1}, \eta_{t+1})]$ in terms of $\Phi(\sigma_t, \eta_t)$. In the following subsection we define the coupling:

6.1.1 Coupling

Consider the following process (σ_t, η_t) on $|\Omega| \times |\Omega|$ where Ω is the space of all possible matchings.

Definition 1. *Choose an edge uniformly at random,*

1. *If insertion is possible in both σ and η then add it probability $\frac{\lambda}{1+\lambda}$ and remove it with probability $\frac{1}{1+\lambda}$.*
2. *If insertion is possible in one and not possible in the other then remove that edge if it is already present in one matching.*

Notice that here we are relying on the fact that one can insert an edge if it is already present in it. It is easy to verify that this indeed is a coupling for the Markov chain defined in the previous section,

We use the following distance function for the aforementioned coupling. Let

$$\sigma \oplus \eta = \left\{ e \in E \mid e \in \left((\sigma \setminus \eta) \cup (\eta \setminus \sigma) \right) \right\} \text{ where } \sigma \text{ and } \eta \in \Omega$$

define the distance function $\Phi(\sigma_t, \eta_t) = |\sigma \oplus \eta|$ which is the number of edges present in one but not in the other. Our objective is to upper-bound $E[\Phi(\sigma_{t+1}, \eta_{t+1})]$ in terms of $d_t = \Phi(\sigma_t, \eta_t)$. Based on the definition of our coupling the following cases may arise once we pick an edge e_r uniformly randomly:

1. **e_r can be added to both σ_t and η_t :** The subevents are (a) e_r was present in both of them, in this case $\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t$, (b) e_r is not present in both of them, in which case again $\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t$ and (c) e_r is present in one but not in other, in which case $\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t - 1$.
2. **e_r can be added in exactly one of σ_t and η_t :** The sub events for this case are (a) e_r is not present in both, which gives $\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t$ (b) e_r is present in one matching and not in other, in which case $\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t - 1$
3. **e_r can't be added to any one of σ_t and η_t :** In this case $\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t$.

Using the above events we can prove the following:

Lemma 2. $E[\Phi(\sigma_{t+1}, \eta_{t+1})] = \Phi(\sigma_t, \eta_t) \left(1 - \frac{1}{m} \right)$.

Proof. Since $\Phi(\sigma_{t+1}, \eta_{t+1})$ can only take two values either d or $d - 1$ we only need to calculate the the probability of the happening of one of these cases. This happens when either the event 1(c) or the event 2(b) takes place (as mentioned above). Thus,

$$\mathbf{Pr}[\Phi(\sigma_{t+1}, \eta_{t+1}) = d_t - 1] = \mathbf{Pr}[1(c) \cup 2(b)]$$

clearly the distance becomes $d_t - 1$ when the chosen edge is one of the edges in $\sigma \oplus \eta$. We can divide $\sigma \oplus \eta$ in to two sets U and \bar{U} . U is the set of edges which can be added to the matching

in which it is not present, and \bar{U} is the set of edge which can't be added to the matching in which it is not present. Using this notation we can write the desired probability as

$$\begin{aligned} \Pr[1(c) \cup 2(b)] &= \sum_{e \in U} \frac{1}{m} \left(\frac{\lambda}{1+\lambda} + \frac{1}{1+\lambda} \right) + \sum_{e \in \bar{U}} \frac{1}{m} \\ &= \frac{|\sigma \oplus \eta|}{m} = \frac{\Phi(\sigma_t, \eta_t)}{m} \end{aligned}$$

Therefore,

$$\begin{aligned} E[\Phi(\sigma_{t+1}, \eta_{t+1})] &= (d_t - 1) \frac{\Phi(\sigma_t, \eta_t)}{m} + d_t \left(1 - \frac{\Phi(\sigma_t, \eta_t)}{m} \right) \\ &= \Phi(\sigma_t, \eta_t) \left(1 - \frac{1}{m} \right) \quad (\text{since } d_t = \Phi(\sigma_t, \eta_t)) \end{aligned}$$

□

We can now prove the following:

Lemma 3. $T_{mix} = O(m \log n)$.

Proof. Given any σ and η we define a $d = \Phi(\sigma, \eta)$ length path as $\sigma = \xi^1, \xi^2 \dots, \xi^d = \eta$ such that ξ^{i+1} is the state obtained by removing exactly one edge from $\xi^i \in \sigma \cap (\sigma \oplus \eta)$ for $i = 1, 2 \dots j$ where ξ^j consists only of edges which do not belong to $\sigma \oplus \eta$ and for all $k \geq j$, ξ^{k+1} is obtained by adding one edge to ξ^k which belongs to $\eta \cap (\sigma \oplus \eta)$. Since $\xi^i, \xi^{i+1} = 1$ for all $i = 1, 2 \dots d-1$, we have

$$\Phi(\sigma, \eta) = \sum_i \Phi(\xi^i, \xi^{i+1})$$

Also we can write

$$E[\Phi(\sigma_{t+1}, \eta_{t+1})] = \beta \Phi(\sigma_t, \eta_t) \quad (\text{with } \beta = \left(1 - \frac{1}{m} \right))$$

this allows us to apply the result from Theorem 2 which gives the following result

$$T_{mix} \leq \frac{\log 2eD}{1 - \left(1 - \frac{1}{m} \right)} \leq m \log(4en) = O(m \log n)$$

where we have used $\beta = (1 - 1/m)$ and $D = 2n$. □

Our algorithm is concisely presented as follows:

Input: A Graph $G = (V, E)$ with $|V| = n$ and $|E| = m$

$\sigma_0 \leftarrow \text{anymatching};$

$\lambda = 2^m;$

for $t = 0$ **to** $10m \log n$ **do**

 choose an edge e_r uniformly randomly ;

$\sigma' = \begin{cases} \sigma_t \cup \{e_r\}, & \text{with probability } \frac{\lambda}{1+\lambda}, \\ \sigma_t \setminus \{e_r\}, & \text{with probability } \frac{1}{1+\lambda}, \end{cases}$

 if σ' is a valid matching;

$\sigma_{t+1} = \sigma';$

 else;

$\sigma_{t+1} = \sigma_t;$

end

return σ_t

Algorithm 1: RandMatching

Theorem 3. *The probability that the random walk in Algorithm 1 ends on a maximum matching is at least $\frac{20}{189}$.*

Proof. Let \mathcal{M}_i be the set all of matchings of size i (and S_i be its cardinality) in the given graph. Also, let k be the size of the maximum matching. We need to find the probability that the random walk lands up on a maximum matching after $T = 10m \log n \geq T_{mix}$ steps. Let X_T be the state after T steps, and π_T be the probability distribution after T steps then by definition of mixing time and triangle inequality,

$$\left| \sum_{\omega \in \mathcal{M}_k} \pi_T(\omega) - \sum_{\omega \in \mathcal{M}_k} \mathcal{G}(\omega) \right| \leq \sum_{\omega \in \mathcal{M}_k} |\pi_T(\omega) - \mathcal{G}(\omega)| \leq \sum_{\omega \in \Omega} |\pi_T(\omega) - \mathcal{G}(\omega)| \leq \frac{1}{e}$$

where $\sum_{\omega \in \mathcal{M}_k} \pi_T(\omega) := \Pr_k(\pi_T)$ is the probability of reaching a maximum matching after T steps (the success probability of the algorithm) and $\sum_{\omega \in \mathcal{M}_k} \mathcal{G}(\omega) := \Pr_k(\mathcal{G})$ is the probability of finding a maximum matching according to the Gibbs distribution. We need to find the probability that X_T is a maximum matching. Thus we have,

$$\Pr_k(\pi_T) \in \left[\Pr_k(\mathcal{G}) - \frac{1}{e}, \Pr_k(\mathcal{G}) + \frac{1}{e} \right]$$

Also,

$$\begin{aligned} \Pr_k(\mathcal{G}) &= \frac{\lambda^k S_k}{\sum_{i=0}^k \lambda^i S_i} \\ &= \frac{\lambda^k S_k}{\lambda^k S_k + \lambda^{k-1} S_{k-1} + \lambda^{k-2} S_{k-2} + \dots + S_0} \end{aligned}$$

dividing by λ^k both numerator and denominator

$$= \frac{S_k}{S_k + \frac{S_{k-1}}{\lambda} + \frac{S_{k-2}}{\lambda^2} + \dots + \frac{S_0}{\lambda^k}}$$

We put $\lambda = S_m$ where $S_m = \max_i^k S_i$

$$= \frac{S_k}{S_k + \frac{S_{k-1}}{S_m} + \frac{S_{k-2}}{S_m^2} + \cdots + \frac{S_0}{S_m^k}} = \frac{1}{1 + \frac{S_{k-1}}{S_k S_m} + \frac{S_{k-2}}{S_k S_m^2} + \cdots + \frac{S_0}{S_k S_m^k}}$$

By definition of S_m , $\frac{S_i}{S_m}$ is always ≤ 1 , hence we have

$$\begin{aligned} \frac{1}{1 + \frac{S_{k-1}}{S_k S_m} + \frac{S_{k-2}}{S_k S_m^2} + \cdots + \frac{S_0}{S_k S_m^k}} &\geq \frac{1}{1 + \frac{1}{S_k} + \frac{1}{S_k S_m} + \frac{1}{S_k S_m^2} \cdots + \frac{1}{S_k S_m^{k-1}}} \\ &\geq \frac{1}{1 + \frac{1}{S_k} \left(1 + \frac{1}{S_m} + \frac{1}{S_m^2} \cdots + \frac{1}{S_m^{k-1}}\right)} \\ &\geq \frac{1}{1 + \frac{1}{S_k} \left(\frac{1 - \left(\frac{1}{S_m}\right)^k}{1 - \frac{1}{S_m}}\right)} \\ &\geq \frac{1}{1 + \frac{1}{S_k} \left(\frac{S_m^k - 1}{S_m^{k-1}(S_m - 1)}\right)} \end{aligned}$$

Since

$$\left(\frac{S_m^k - 1}{S_m^{k-1}(S_m - 1)}\right) = \left(1 + \frac{S_m^{k-1} - 1}{S_m^{k-1}(S_m - 1)}\right) = 1 + \Theta\left(\frac{1}{S_m}\right) \leq \frac{11}{10}$$

$$\Pr_k(\mathcal{G}) \geq \frac{1}{1 + \frac{1}{\frac{11S_k}{10}}} \geq \frac{10}{21}$$

This gives us,

$$\Pr_k(\pi_T) \geq \frac{10}{21} - \frac{1}{e} \geq \frac{20}{189}$$

□

Notice that our proof still goes through even if we choose a λ that is larger than S_m (we can take $\lambda = 2^m$). Therefore λ can be represented using m bits. We can now prove the main theorem,

Theorem 4. *Given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, there exists a randomized algorithm that runs in $O(m \log^2 n)$ time and finds a maximum matching with high probability.*

Proof. Each step of the algorithm runs in $O(1)$ time (we just need to maintain an array which indicates whether the i^{th} vertex is occupied in the matching or not), thus one call of Algorithm 1 runs in $O(m \log n)$ time which by Theorem 3 lands on a maximum matching with probability $\frac{20}{169}$. Thus calling it $10 \log n$ times independently, ensures that we land on a maximum matching in one call is at least $1 - \left(\frac{169}{189}\right)^{10 \log n} = 1 - \frac{1}{n^{O(1)}}$. □

7 Lower Bound via Conductance

The conductance method as defined in the Section is used to obtain lower bounds on the mixing time of the a Markov chain. To get a lower bound on T_{relax} we need an upper-bound on ϕ , and by definition of ϕ , for any cut (S, \bar{S}) .

$$\phi \leq \left(\frac{\sum_{i \in S, j \in \bar{S}} \pi_i P_{ij}}{(\sum_{i \in S} \pi_i)(\sum_{i \in \bar{S}} \pi_i)} \right)$$

This allows us to observe:

Lemma 4. *For any graph G , the conductance of our Markov Chain satisfies*

$$\phi \leq O\left(\frac{k}{m}\right)$$

where k is the size of maximum matching.

Proof. To give an upper bound on conductance we need to construct a cut (S, \bar{S}) for which we can estimate the above quantity. Let S be consisting of exactly one matching which is the maximum matching m_k where k is the size of the maximum matching. Thus the number of edges going out of S is k .

$$\frac{\sum_{i \in S, j \in \bar{S}} \pi_i P_{ij}}{(\sum_{i \in S} \pi_i)(\sum_{i \in \bar{S}} \pi_i)} = \frac{\frac{\lambda^k}{Z} \cdot k \cdot \frac{1}{m(\lambda+1)}}{\frac{\lambda^k}{Z} \left(1 - \frac{\lambda^k}{Z}\right)} = \frac{k}{(1+\lambda)m} \cdot \frac{1}{\left(1 - \frac{\lambda^k}{Z}\right)}$$

Also using the terminology of Theorem 3

$$\frac{\lambda^k}{Z} = \frac{\lambda^k}{\sum_{i=0}^k S_i \lambda^i} = \frac{1}{\sum_{i=0}^k S_{k-i} \frac{1}{\lambda^i}} \leq \frac{1}{\sum_{i=0}^k \frac{1}{\lambda^i}} = \frac{\lambda^k(\lambda-1)}{\lambda^{k+1}-1} = 1 - \Theta\left(\frac{1}{\lambda}\right)$$

Therefore,

$$\frac{\sum_{i \in S, j \in \bar{S}} \pi_i P_{ij}}{(\sum_{i \in S} \pi_i)(\sum_{i \in \bar{S}} \pi_i)} = \frac{k}{(1+\lambda)m} \cdot \frac{1}{\Theta\left(\frac{1}{\lambda}\right)} = \Theta\left(\frac{k}{m}\right)$$

Thus the result follows. \square

As a result of the previous lemma we have the following result.

Lemma 5. *For any graph G , the mixing time of our Markov chain satisfies, $T_{mix} \geq \Omega\left(\frac{m}{k}\right)$*

Proof. Follows from results in Section 4.3. \square

Thus the lower bound is sharp (upto logarithmic factors) if the size of the matching in the given graph is small (say at most $O(\text{poly log } n)$).

A Note on Other methods to prove Lower Bound: There are other methods, apart from conductance, which can be used to prove lower bounds on mixing time. Although powerful and useful in many contexts, it is not clear whether such methods could be applied to our chain. For eg. the Wilson's method [1] expects the knowledge of one eigenvector of the transition vector that is different from the all 1's vector and the corresponding eigenvalue lies in the range $(0, \frac{1}{2})$. Since the matrix P for our case is an exponential sized matrix with apparently no useful pattern in the entries, it is not clear how to come up with such an eigenvector. In fact, we made several educated guesses for coming up with such a vector all of which failed to serve our purpose.

8 Conclusion

In this paper, we gave a new randomized algorithm for finding maximum matchings in general bipartite graphs that runs in $O(m \log^2 n)$ time that improves upon the running time of Micali and Vazirani. Our algorithm was based on the MCMC paradigm which performs a truncated random walk on the Markov Chain defined by the Glauber Dynamics with parameter λ . Apart from the benefit of being very simple (both in analysis and implementation) our algorithm is the first near linear time complexity algorithm for the maximum matching problem for general graphs. Moreover, unlike [15] the running time of our algorithm is not a random variable.

To our knowledge this is for the first time Glauber dynamics and the nature of Gibbs distribution has been exploited to design a faster algorithm for a problem for which efficient solutions are already known, and we hope this idea can be of use in other problems as well. The obvious open problem will be to improve both the upper-bounds and lower bounds on the mixing time. Is it possible to improve upon the present bound to get an $O(m)$ time algorithm? Also one can explore the possibility of proving a tighter lower bound of $\Omega(m)$ by using more refined techniques. More specifically, it would be interesting to see if one can obtain an explicit eigenvector of the transition matrix P (that is different from all 1's vector) and apply Wilson's to get an improved lower bound on the mixing time.

9 Acknowledgements

Manjish Pal would like to thank Eric Vigoda for answering his mails about the Gibbs distribution. Thanks to Purushottam Kar for his valuable comments on the paper.

References

- [1] N. Berestycki, *Eight lectures on Mixing Times*, 2009.
- [2] R. Bubeley and M.E. Dyer, *Path coupling, Dobrushin uniqueness, and approximate counting*, 38th Annual Symposium on Foundations of Computer Science, IEEE, 1997.
- [3] R. Cole and J.E. Hopcroft, *On edge coloring bipartite graphs*, SIAM J. Comput., 11(3), 1982, pp. 540-546.
- [4] R. Cole, K. Ost, and S. Schirra, *Edge-coloring bipartite multigraphs in $O(E \log D)$ time*, Combinatorica, 21(1), 2001, pp. 5-12.
- [5] M. Dyer, A. Frieze and M. Jerrum, *On Counting Independent Sets In Sparse Graphs*, SIAM Journal on Computing 31(5), 2001, pp. 1527-1541.
- [6] M. Dyer, A. Frieze and R. Kannan, *A random polynomial time algorithm for approximating the volume of convex sets*, Journal of the Association for Computing Machinery, 38,1991, pp. 1-17.
- [7] M. Dyer and C. Greenhill, *On Markov chains for independent sets*, Journal of Algorithms 35(1) 2000, pp. 17-49.
- [8] M. Dyer, L. A. Goldberg and M. Jerrum, *Markov chain comparison*, Probability Surveys Vol. 3, 2006, pp. 891-111.
- [9] J. Edmonds, *Paths, Trees and Flowers*, Canadian J. 1965, pp. 449-467

- [10] H. Gabow, *Implementation of Algorithms for Maximum Matching of Graphs*, PhD Thesis, 1973
- [11] A. Galanis, Q. Ge, D. Stefankovic, E. Vigoda and L. Yang., *Improved Inapproximability Results for Counting Independent Sets in the Hard-Core Model*, arXiv:1105.5131v1 , 2011.
- [12] H.N. Gabow and O. Kariv, *Algorithms for edge coloring bipartite graphs and multigraphs*, SIAM J. Comput., 11(1), 1982, pp. 117-129.
- [13] A. Goel, M. Kapralov, and S. Khanna, *Perfect matchings via uniform sampling in regular bipartite graphs*, Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms, 2009, pp. 11-17.
- [14] A. Goel, M. Kapralov, and S. Khanna, *Perfect matchings in $O(n^{1.5})$ time in regular bipartite graphs.*, arXiv:0902.1617v2, 2009.
- [15] A. Goel, M. Kapralov, and S. Khanna, *Perfect Matchings in $O(n \log n)$ Time in Regular Bipartite Graphs*, STOC, 2010.
- [16] J.E. Hopcroft and R.M. Karp, *An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graphs*, SIAM J.Comput., 2(4),1973, pp. 225-231.
- [17] Mark Jerrum and Alistair Sinclair, *The Markov chain Monte Carlo method: an approach to approximate counting and integration* , "Approximation Algorithms for NP-hard Problems," D.S.Hochbaum ed., PWS Publishing, Boston, 1996.
- [18] M.R. Jerrum and A.J. Sinclair, *Approximating the permanent*, SIAM Journal on Computing, 1989 vol. 18, pp.1149-1178.
- [19] D. König, *Über graphen und ihre anwendung auf determinantentheorie und mengenlehre*, Math. Annalen, 77,1916, pp. 453-465.
- [20] T. Kameda and I. Munro, *A $O(|V| \cdot |E|)$ Algorithm for Maximum Matchings of Graphs*, Computing, vol.12, pp. 91-98, 1974.
- [21] R. Kleinberg and Eva Tardos, *Algorithm Design*, Addison Wesley, 2005.
- [22] E.G. Lawler, *Combinatorial Optimization Theory*, Holt, Rinehart and Winston, Chapter 6, pp. 217-239, 1976.
- [23] D. A. Levin, Y. Peres, E. Wilmer, *Markov Chains and Mixing Times*, American Mathematical Society, 2009.
- [24] S. Micali and V. V. Vazirani, *An $O(|E|\sqrt{|V|})$ time Algorithm for Maximum Matchings*, FOCS, 1980, pp.17-27.
- [25] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
- [26] E. Mossel, D. Weitz and N. Wormald, *On the hardness of sampling independent sets beyond the tree threshold*, Probability Theory and Related Fields, 2009 vol. 143, no. 3-4, pp. 401-439.
- [27] F. Martinelli, A. Sinclair and D. Weitz, *Fast mixing for independent sets, colorings and other models on trees*, SODA, 2004, pp. 456-465
- [28] R. Restrepo, D. Stefankovic, J.C. Vera, E. Vigoda and L. Yang, *Phase Transition for Glauber Dynamics for Independent Sets on Regular Trees*, arXiv:1007.2255v1, 2010.
- [29] A. Sly, *Computational Transition at the Uniqueness Threshold*, FOCS, 2010, pp. 287-296.
- [30] D. Weitz, *Counting independent sets up to the tree threshold*, STOC, 2006, pp. 140-149.

- [31] E. J. Vigoda, *Sampling from Gibbs distributions*, 1999, Phd Thesis.
- [32] S. Vadhan, *The Complexity of Counting in Sparse, Regular, and Planar Graphs*, SIAM Journal on Computing 31(2), 2001, pp. 398-427.