CS 466/666: Algorithm Design and Analysis **University of Waterloo: Fall 2023**

Lecture 24

December 4, 2023

Instructor: Sepehr Assadi *Scribe: Janani Sundaresan*

Topics of this Lecture

1 A Quick Recap of Random Walks

In the last lecture, we looked at Markov chains and stationary distributions. A Markov chain consists of n states and a probability matrix—called the *transition matrix*—$P \in \mathbb{R}^{n \times n}$, where P_{ij} denotes the probability that the next state of the random walk is j assuming the current state is i. Recall the following standard definitions:

- For any states i and j, the *hitting time* $h(i, j)$ is the expected length of the random walk starting from i and ending in j. We refer to $h_{i,i}$ as the *return time*.
- The *stationary distribution* is a distribution π over the states which remains unchanged after taking another step of the random walk, i.e., $\pi = \pi \cdot P$.
- An *irreducible* Markov chain is a one where the underlying directed graph of transition matrix (having any edge (i, j) whenever $P_{ij} > 0$), is strongly connected.

We will be using the following special case of the fundamental theorem of Markov Chains in this lecture.

Theorem 1. *In any irreducible Markov chain, for any state i, $h(i, i) = 1/\pi_i$; namely, the return time is equal to inverse stationary probability.*

We now use this theorem to design a brilliant and yet extremely simple algorithm for finding perfect matching in *regular* bipartite graphs, due to Goel, Kapralov, and Khanna [GKK13].

2 Perfect Matching in Regular Bipartite Graphs

In this section, we will use a random walk to construct an algorithm which finds a perfect matching in d-regular bipartite graphs in $O(n \log n)$ time. Notice that since the input is of size $\Theta(nd)$, this is even faster than reading the input once! Let us first show that regular bipartite graphs *always* have a perfect matching to begin with.

Proposition 2. *For any integers $n, d \geqslant 1$, any d-regular bipartite graph $G = (L, R, E)$ with $n := |L| = |R|$ has a perfect matching.*

Proof. This can be proven easily using Hall's theorem or alternatively via fractional matchings.

Proof via Hall's theorem. Recall that Hall's (Marriage) Theorem states:

> Let $G = (L \cup R, E)$ be a bipartite graph. G has a matching that matches all vertices in L iff for every subset $A \subseteq L$, we have $|A| \leqslant |N(A)|$ in G.

Now consider our d-regular graph G and fix any set $A \subseteq L$. Let $m(A, N(A))$ denote the number of edges between A and $N(A)$. We have,

$$|A| \cdot d = m(A, N(A)) \leqslant |N(A)| \cdot d;$$

the left equality holds because G is d-regular and all edges in A go to $N(A)$, and the right inequality holds because subset of edges of $N(A)$ are going back to A. This implies $|A| \leqslant |N(A)|$. Thus G has a matching that matches all of L by Hall's theorem. But since $|L| = |R|$, this means G has a perfect matching.

Proof of via fractional matchings. Recall that a fractional matching is an assignment $x \in \mathbb{R}^E$ satisfying

$$\forall v \in V: \quad \sum_{e \ni v} x_e \leqslant 1 \qquad \text{and} \qquad \forall e \in E: \quad x_e \geqslant 0.$$

We proved earlier in the course that any fractional matching in bipartite graphs can be turned into an integral (standard) matching with $\sum_{e \in E} x_e$ many edges. In other words, integrality gap of the linear program for bipartite matching is 1.

Now consider our d-regular graph G and assign $x_e = 1/d$ to every edge. This is clearly a fractional matching as

$$\forall v \in V: \quad \sum_{e \ni v} x_e = \deg(v) \cdot \frac{1}{d} = 1,$$

and we have

$$\sum_{e \in E} x_e = n \cdot d \cdot \frac{1}{d} = n;$$

hence, there is an integral matching of size n also in G, namely a perfect matching. □

Now that we established that there is a perfect matching, let us see how we can find it.

Theorem 3 ([GKK13])**.** *There is a randomized algorithm that for every integer $n, d \geqslant 1$, given a bipartite d-regular graph $G = (L \cup R, E)$ with $n = |L| = |R| = n$, outputs a perfect matching of G in $O(n \log n)$ expected time.*

The algorithm proceeds by finding augmenting paths in the graph using random walks. First, we establish what augmenting paths are and their role in increasing the size of a matching. Then, we will see how we can construct these augmenting paths.

2.1 Role of Augmenting Paths

We define augmenting paths formally.

Definition 4. Given a graph $G = (V, E)$ and a matching $M \subseteq E$, a path $P = (u_1, u_2, \ldots, u_{2k})$ in G is said to be an **augmenting path** for M if,

- Edge (u_{2i-1}, u_{2i}) is not present in M for all $i \in [k]$.
- Edge (u_{2i}, u_{2i+1}) is present in M for all $i \in [k-1]$.

That is, path P starts and ends at unmatched vertices, and alternates between edges from M and $E \setminus M$.

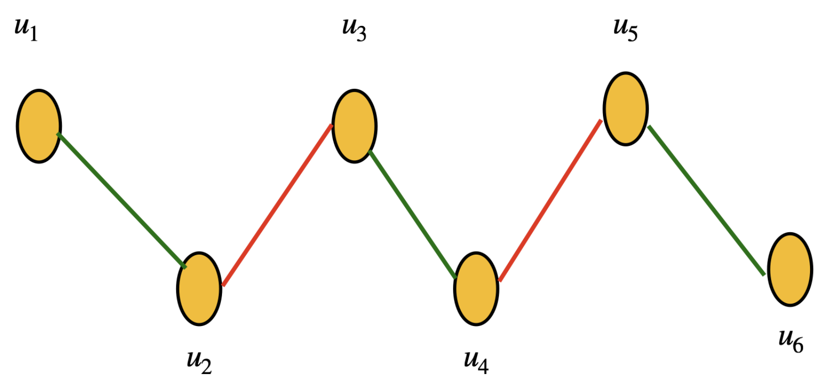

Figure 1: An augmenting path of length 5. The red edges are matching edges, and green edges are non-matching edges. Vertices u_1, u_2 are unmatched. Replacing red edges with green edges in M gives us a larger matching M'.

It is easy to see that whenever we find an augmenting path for a matching M, we can find another matching M', the size of which is larger than that of M by one.

Claim 5. *Given a matching M of graph $G = (L \cup R, E)$ and an augmenting path P of M in graph G, there is a matching M' of G such that $|M'| = |M| + 1$.*

Proof. Let path P be $(u_1, u_2, \ldots, u_k)$ for some even k. Let $E(P)$ be the set of all edges which are present in path P. Then we construct M' as,

$$M' = (M \setminus E(P)) \cup (E(P) \setminus M).$$

In other words, we replace all the edges in path P which are in M by all the edges in path P which are not in M. Refer to Figure 1 for an illustration.

To see that M' remains a matching, note that no edge which is not present in path P is affected, so the status of all vertices which are not in P is unchanged. For vertices in P, both u_1, u_k are unmatched, and we add one edge incident to each of them to M'. For all the other vertices u_i for $1 < i < k$, we remove edge (u_i, u_{i+1}) and add edge (u_{i-1}, u_i) or vice-versa. M' never has two edges incident to one vertex.

Finally, the size of M' is larger than that of M by one, because we remove $(k/2) - 1$ edges from M, and add $(k/2)$ edges. $\square$

The algorithm in Theorem 3 works by repeatedly find augmenting paths to increase the size of a maintained matching. The following lemma states that we can quickly find such paths, for any matching M which is not perfect.

Lemma 6. *For any $k, n, d \geqslant 1$, given a bipartite graph $G = (L \cup R, E)$ which is d-regular with $|L| = |R| = n$, and a matching M which leaves $2k$ vertices unmatched from $L \cup R$, there is a randomized algorithm which finds an augmenting path for M in expected $O(n/k)$ time.*

We will prove this lemma in the next subsection. Let us show that finding these augmenting paths proves Theorem 3.

Proof of Theorem 3. We start with matching $M = \emptyset$. Then we can find an augmenting path in $O(1)$ expected time by Lemma 6, and increase the size of the matching to 1 by Claim 5. We continue to do this until M becomes a perfect matching.

The size of M slowly increases from 0 to n. When the total number of unmatched vertices is $2k$, we know from Lemma 6 that we can find an augmenting path in $O(n/k)$ expected time. Then we use Claim 5 to reduce the number of unmatched vertices by 2. The number of unmatched vertices decreases from $2n$ to 0, so the total expected running time is,

$$2n \cdot (\frac{1}{2n} + \frac{1}{2n-2} + \ldots + \frac{1}{4} + \frac{1}{2}) \leqslant 2n \cdot \sum_{i=1}^{n} \frac{1}{i} = O(n \log n),$$

where in the last step we used that the harmonic sum up to n terms is bounded by $O(\log n)$. ☐

2.2 Finding Augmenting Paths

In this section, we will see how to find augmenting paths by using a random walk.

Notation. We use $V(M) \subseteq L \cup R$ to denote the set of all vertices which is matched by M. For any matching M, and vertex v which is matched by M, we use $M(v)$ to denote the vertex M matches v to. That is, $(v, M(v)) \in M$.

Informally, when we try to find an augmenting path, we want to alternate between matching edges and the non-matching edges. We pick all the non-matching edges at random out of the current vertex. The matching edges are picked deterministically; there is only one choice for the edge in the matching.

Algorithm 1. Algorithm `Aug-Walk` for finding a single augmenting path for a matching M.

(*i*) Pick an unmatched vertex $u_1 \in L$ at random. Start walk W from u_1, and repeat the following.

(*ii*) Let $u \in L$ be the current vertex for walk W. Pick an edge (u, v) with $v \in R$ at random out of vertex u, and add it to W. Stop if v is unmatched.

(*iii*) If v is matched, add edge $(v, M(v))$ to W, and move to vertex $M(v) \in L$. Continue from step (*ii*).

To prove Lemma 6, we need to show that algorithm `Aug-Walk` terminates in $O(n/k)$ time when $2k$ vertices are left unmatched by M. The walk W is not an *undirected random walk* on the underlying graph, as all the matching edges are picked deterministically. We will construct a Markov Chain from graph G and matching M that captures this walk.

Definition 7. Given a bipartite d-regular graph $G = (L \cup R, E)$ and a matching M which leaves $2k$ vertices unmatched, we construct the **matching Markov chain** as follows.

(i) Create a state for each vertex $v \in L \cup R$, and add all edges by orienting them from L to R.

(ii) For each matching edge (u, v) with $u \in L, v \in R$, combine both states u, v into one state. This state has all the edges incoming to v as well as all the edges outgoing from u.

(iii) Create a new state $\star$ which denotes the start and final states. Add edges from $\star$ to all unmatched vertices in L, and add edges from all unmatched vertices in R to $\star$.

(iv) For each state, make all outgoing edges equiprobable.

Refer to Figure 2 for an illustration of Definition 7.

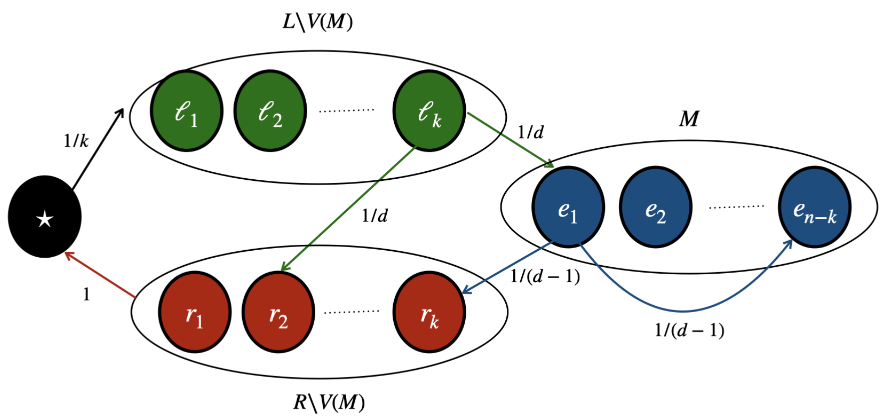

Figure 2: The black state $\star$ has k outgoing equiprobable edges to $L \setminus V(M)$. The green states $L \setminus V(M)$ have d outgoing edges each corresponding to edges in $E \setminus M$. The brown states $R \setminus V(M)$ have one outgoing edge each to $\star$. The blue states M have $d-1$ outgoing edges corresponding to edges in $E \setminus M$.

The key point is that when we want to pick a matching edge from $v \in R$, there is only one choice for such an edge, and the state of the Markov chain does not change. We have *fused* all matching edges into a single state. Let us formally argue why `Aug-Walk` is a random walk on our matching Markov chain.

Claim 8. *Algorithm* `Aug-Walk`*, for graph G and matching M performs a random walk on the Markov chain from Definition 7.*

Proof. We start by picking an unmatched vertex at random, and moving out of state $\star$ picks such a vertex $u_1 \in L$, by step (iii) of the construction. We pick an edge (u_1, u_2) out of u_1 at random, and all edges out of u are equiprobable, by step (v). If $u_2 \in R$ is unmatched, we move out of u_2 into final state $\star$ by step (iii).

If u_2 is matched, u_2 is fused in state $u_2M(u_2)$ by step (ii). We move out of $M(u_2) \in L$ at random into another vertex from R in the graph G, which may be a matching edge state in Markov chain, or a state for an unmatched vertex in R. We repeat till we reach an unmatched vertex in R, and in one more timestep we reach final state $\star$. The walk starts from $\star$, and ends when we reach $\star$ again. $\square$

Remark. Note that `Aug-Walk` returns a *walk* on Markov chain and not a path. However, it can be converted into a path by removing cycles.

Given Claim 8, bounding the runtime of `Aug-Walk` is the same (up to an $O(1)$ factor) as bounding the return time of the state marked by $\star$, i.e., $h(\star, \star)$ in the matching Markov chain. To do so, we only need to compute the stationary distribution of our Markov chain and we can then apply Theorem 1 to conclude the proof.

We will look at the transition matrix of our Markov chain to find the stationary distribution. Let P_{ij} denote the probability of entering state j from state i. There are $n + k + 1$ states in total, divided into four types (see Figure 2). We order them as follows:

1. State $\star$ corresponding to the start and final states. There are k edges outgoing from $\star$ to $L \setminus V(M)$, each with probability $1/k$. There are k edges incoming to $\star$ from $R \setminus V(M)$, each with probability 1.
2. State $\ell_i \in L \setminus V(M)$ for each unmatched vertex $\ell_i \in L$ for $i \in [k]$. Each such state has d outgoing edges, each with probability $1/d$, to either states corresponding to matching edges or unmatched vertices in R. There is one edge incoming to each ℓ_i from $\star$ with probability $1/k$.
3. State $r_i \in R \setminus V(M)$ for each unmatched vertex $r_i \in R$ for $i \in [k]$. They have only one outgoing edge to state $\star$. There are d edges incoming to each r_i, based on edges $(u, r_i) \in E \setminus M$. They may have probability $1/(d-1)$ (if they are from matching edge states), or $1/d$ (if they are from $L \setminus V(M)$).
4. State $e_i = vM(v)$ corresponding to edge $e_i = (v, M(v)) \in M$ incident on matched vertex $v \in R$ for $i \in [n-k]$. Each state has $d-1$ outgoing edges, each with probability $1/(d-1)$, to other matching edge states or unmatched vertices in R. There are $d-1$ incoming edges, based on edges incident to $v \in R$. Again, they have probability $1/(d-1)$ from matching edge states, or $1/d$ from $L \setminus V(M)$.

Let us show what the stationary distribution is.

Claim 9. *Let $t := (n-k)(d-1) + 3kd$. The stationary distribution of matching Markov chain is π where:*

$$\pi_\star = \frac{kd}{t}, \qquad \pi_{\ell_i} = \pi_{r_j} = \frac{d}{t} \text{ for } i, j \in [k], \qquad \pi_{e_i} = \frac{d-1}{t} \text{ for } i \in [n-k].$$

Proof. First, let us show that π is a probability distribution over the states:

$$\begin{aligned}
\sum_{\text{State } s} \pi_s &= \pi_\star + \sum_{\ell \in L \setminus V(M)} \pi_\ell + \sum_{r \in R \setminus V(M)} \pi_r + \sum_{e \in M} \pi_e \\
&= \frac{kd}{t} + k \cdot \frac{d}{t} + k \cdot \frac{d}{t} + (n-k) \cdot \frac{(d-1)}{t} \\
&= \frac{1}{t} \cdot (3kd + (n-k)(d-1)) \\
&= 1.
\end{aligned}$$

We will check what the probability is for transitioning into every state starting from distribution π. It is easy to check for states $\star$ and any $\ell \in L \setminus V(M)$.

$$\Pr(\text{Entering } \star \text{ starting with } \pi) = \sum_{r \in R \setminus V(M)} \pi_r \cdot p_{r,\star} = k \cdot \frac{d}{t} = \frac{kd}{t}.$$

$$\Pr(\text{Entering any } \ell \in L \setminus V(M) \text{ starting with } \pi) = \pi_\star \cdot p_{\star,\ell} = \frac{kd}{t} \cdot \frac{1}{k} = \frac{d}{t}.$$

For any state $r \in R \setminus V(M)$, we have an incoming edge from all $\ell \in L \setminus V(M)$ with $(\ell, r) \in E$, and all edge states $e = (u, v) \in M$ with $(v, r) \in E$. There are totally d incoming edges.

$$\begin{aligned}
\Pr\left(\text{Entering } r \in R \setminus V(M) \text{ starting with } \pi\right) &= \sum_{e \in M} \pi_e \cdot p_{e,r} + \sum_{\ell \in L \setminus V(M)} \pi_\ell \cdot p_{\ell,r} \\
&= \sum_{(u,v) \in M; (v,r) \in E} \frac{d-1}{t} \cdot \frac{1}{d-1} + \sum_{\ell \in L \setminus V(M); (\ell,r) \in E} \frac{d}{t} \cdot \frac{1}{d} \\
&= \sum_{(u,v) \in M; (v,r) \in E} \frac{1}{t} + \sum_{\ell \in L \setminus V(M); (\ell,r) \in E} \frac{1}{t} \\
&= \frac{d}{t}.
\end{aligned}$$

Similarly, for any edge $e = (u, v) \in M$, with $d - 1$ incoming edges,

$$\begin{aligned}
\Pr\left(\text{Entering } e = (u,v) \in M \text{ starting with } \pi\right) &= \sum_{e' \in M} \pi_{e'} \cdot p_{e',e} + \sum_{\ell \in L \setminus V(M)} \pi_\ell \cdot p_{\ell,e} \\
&= \sum_{(u',v') \in M; (v',u) \in E} \frac{d-1}{t} \cdot \frac{1}{d-1} + \sum_{\ell \in L \setminus V(M); (\ell,u) \in E} \frac{d}{t} \cdot \frac{1}{d} \\
&= \sum_{(u',v') \in M; (v',u) \in E} \frac{1}{t} + \sum_{\ell \in L \setminus V(M); (\ell,u) \in E} \frac{1}{t} \\
&= \frac{d-1}{t}.
\end{aligned}$$

This proves that π is indeed the stationary distribution of the matching Markov chain. □

Remark. We could have alternatively changed the matching Markov chain slightly so that it becomes the transition matrix of a random walk on a *directed* graph wherein in-degree and out-degree of all vertices is the same. One can generally show that random walk on any directed graph with this property—often called a balanced directed graph—has stationary distribution wherein probability of each vertex is its out-degree divided by the number of edges (namely, a direct analogue of the stationary distribution in undirected graphs).

We are ready to prove Lemma 6, as we know what the stationary distribution is now.

Proof of Lemma 6. We use algorithm `Aug-Walk` to find an augmenting path in graph G for matching M. By Claim 8, we know that this is a random walk on the matching Markov chain from Definition 7.

We know that any random walk on the matching Markov chain which starts at state $\star$ reaches $\star$ again in expected number of steps $\frac{1}{\pi_\star}$ by Theorem 1. We know what $\pi_\star$ is from Claim 9. Therefore, the expected number of steps of the random walk is,

$$\frac{t}{kd} = \frac{(n-k)(d-1) + 3kd}{kd} = \frac{nd + 2kd - (n-k)}{kd} \leqslant \frac{2n}{k}.$$

□

References

[GKK13] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. Perfect matchings in $O(n \log n)$ time in regular bipartite graphs. *SIAM Journal on Computing*, 42(3):1392–1404, 2013. 1, 2