CS 388R: Randomized Algorithms	Fall 2015
Lecture 16 — Nov. 2, 2015	
Prof. Eric Price	*Scribe: Jiong Zhang, Chao-Yuan Wu*

1 Overview

In the last lecture we studied bipartite matching problem. In this lecture we extend our analysis to online setting.

2 Introduction

In the original bipartite matching problem we seek to find a maximum matching, i.e. a matching that contains the largest possible number of edges given a graph. On the other hand, in a "online" bipartite matching problem, we observe nodes one by one and assign matchings in an online fashion. Our goal is to find an algorithm that maximizes the competitive ratio $R(A)$.

Definition 1. *(Competitive ratio)*

$$R(A) := \liminf_{I} \frac{\mathbb{E}[\mu_A(I)]}{\mu_*(I)} \tag{1}$$

where $\mu_A(I)$ and $\mu_A(I)$ denote the size of matching for an algorithm A and maximum matching size respectively, given input $I :=$ {graph, arriving order}.

Obviously $R(A) \leq 1$, but can we find a lower bound for $R(A)$?

3 Naive algorithm

Since each edge can block at most two edges, we have $R(A) \geq 0.5$. On the other hand, for any deterministic algorithm A, we can easily find an adversarial input I such that $R(A) \leq 0.5$. See left hand side of Figure 1 for example.

Can we achieve better results with random assignments? Consider the graph on the right hand side of Figure 1, where there is a perfect matching from n nodes on the left to n nodes on right, and the second haft of us are fully connected to the first half of v. Under this setting, the number of correctly matched vertices in the second half of v is at most $n/2$. The expected number of correctly

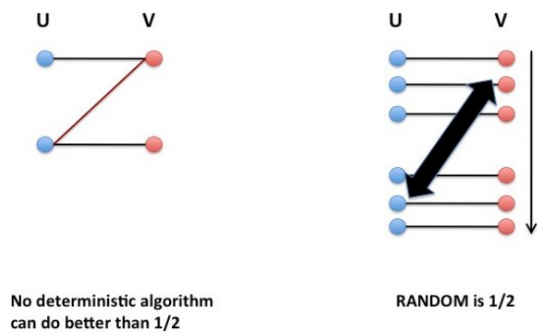

Figure 1: comparison of deterministic and randomized algorithms

matched vertices in the first half is given by:

$$\mathbb{E}[\# \text{ correctly matched vertices}] = \sum_{i=1}^{n/2} \mathbb{P}[i\text{-th vertex is correctly matched}] \tag{2}$$

$$\leq \sum_{i=1}^{n/2} \frac{1}{\frac{n}{2} - i + 2} \tag{3}$$

$$\leq \log(\frac{n}{2} + 1) \tag{4}$$

$$\tag{5}$$

Since $\mu_* = n$, the competitive ratio R:

$$R(A) = \frac{\mathbb{E}[\# \text{ matched}]}{n} \leq \frac{\frac{n}{2} + log(\frac{n}{2} + 1)}{n} \rightarrow \frac{1}{2}$$

We see, unfortunately, this randomized algorithm still does not do better than 1/2.

4 Ranking algorithm

Here we introduce Ranking Algorithm. Consider a graph G with arriving order π. Instead of simply choosing a random edge, we first randomly permute the v's with permutation $\sigma(\cdot)$. We then match u to

$$v := \operatorname*{argmin}_{v' \in \mathcal{N}(u)} \sigma(v')$$

where $\mathcal{N}(u)$ denotes the neighbors of u.

Now we prove that this algorithm achieves a competitive ratio of $1 - 1/e$. We begin by defining our notation. The matching is denoted by Matching(G, π, σ). $M^*(v)$ denotes the vertex matched to v in perfect matching. $G := \{U, V, E\}$, where U, V, E denote left nodes, right nodes and edges respectively.

Lemma 2. *Let $H := G - \{x\}$ with permutation π_H and arriving order σ_H induced by π, σ respectively.* Matching(H, π_H, σ_H) = Matching(G, π, σ) + *augmenting path from x downwards.*

This can be easily seen from the design of the algorithm.

Lemma 3. *Let $u \in U$ and $M^*(u) = v$, if v is not matched under σ, then u is matched to v' with $\sigma(v') \leq \sigma(v)$.*

This again is obvious.

Lemma 4. *Let x_t be the probability that the rank-t vertex is matched. Then*

$$1 - x_t \leq \frac{\sum_{s \leq t} x_s}{n} \tag{6}$$

$$\tag{7}$$

Proof. (Intuitive but incorrect) Let v be the vertex with $\sigma(v) = t$. Note, since σ is uniformly random, v is uniformly random. Let $u := M^*(v)$. Denote by R_t the set of left nodes that are matched to rank $1, 2, \ldots, t$ vertices on the right. We have $\mathbb{E}[\,|R_{t-1}|\,] = \sum_{s \leq t-1} x_s$. If v is not matched, u is matched to some $\tilde{v}$ such that $\sigma(\tilde{v}) < \sigma(v) = t$, or equivalently, $u \in R_{t-1}$. That said,

$$\mathbb{P}(v \text{ not matched}) = 1 - x_t = \mathbb{P}(u \in R_{t-1}) = \mathbb{P}(\frac{\mathbb{E}\left[|R_{t-1}|\right]}{n}) \leq \frac{\sum_{s \leq t} x_s}{n}$$

□

However this proof is not correct since u and R_{t-1} are not independent and thus $\mathbb{P}(u \in R_{t-1}) \neq \mathbb{P}(\frac{\mathbb{E}[|R_{t-1}|]}{n})$. Instead, we use the following lemma to complete the correct proof.

Lemma 5. *Given σ, let $\sigma^{(i)}$ be the permutation that is σ with v moved to the i-th rank. Let $u := M^*(v)$. If v is not matched by σ, for every i, u is matched by $\sigma^{(i)}$ to some $\tilde{v}$ such that $\sigma^{(i)}(\tilde{v}) \leq t$.*

Proof. By Lemma 2, inserting v to i-th rank causes any change to be a move up.

$$\sigma^{(i)}(\tilde{v}) \leq \sigma(\tilde{v}) + 1 \leq t$$

□

Proof. (Correct proof of Lemma 4)

Choose random σ and v, let $\sigma' = \sigma$ with v moved to rank t. $u := M^*(v)$. According to Lemma 5, if v is not matched by σ (with probability x_t), u in σ' is matched to $\tilde{v}$ with $\sigma'(\tilde{v}) \leq t$, or equivalently $u \in R_t$. Note, u and R_t are now independent and $\mathbb{P}(u \in R_t) = |R_t|/n$ holds. The same arguments as in the previous proof complete the proof. □

With Lemma 4, we can finally obtain the final results. Let $s_t := \sum_{s \leq t} x_s$. Lemma 4 is equivalent to $s_t(1 + 1/n) \geq 1 + s_{t-1}$. Solving the recursion, it can also be rewritten as $s_t = \sum_{s \leq t}(1 - 1/(1+n))^s$ for all t. The competitive ratio is thus, $s_n/n \to 1 - 1/e$.

References

[MR] Rajeev Motwani, Prabhakar Raghavan Randomized Algorithms. *Cambridge University Press*, 0-521-47465-5, 1995.