CS787: Advanced Algorithms	**Scribe:** Zelin (Bobby) Lv
Lecture 14: Online Bipartite Matching	**Date:** 03/14/2019

14.1 Online Bipartite Matching

Input: Unweighted Bipartite Graph $G = (L \cup R; E); E \subseteq L \times R$

- L is known ahead of time
- R is revealed one vertex at a time
- When $v \in R$ arrives, we see all edges adjacent to v
- Algorithm needs to commit to a match for v before observing any future arrivals

Goal: Pick a matching of maximum size.

Example:

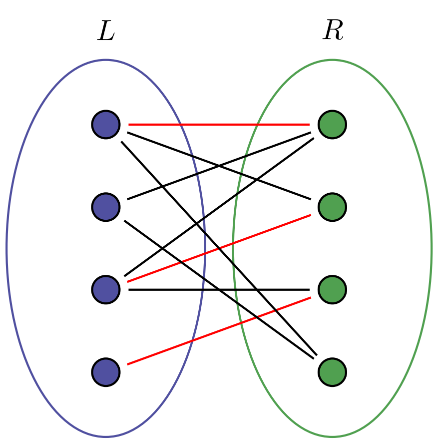


Figure 14.1.1: A result graph where node of R arrives in Online Fashion (The ordering is the top comes first).

Clearly, we found a maximal matching but not maximum matching.

We see that our online Bipartite Matching is not optimal. What would you do for the first vertex arrives? There is not a large space we can do because all the edges seem identical.

14.2 Deterministic Algorithm

Theorem: No deterministic algorithm can get a C.R. (Competitive Ratio) < 2 for solving online bipartite matching.
Proof: Consider the following case with an adversary setting:

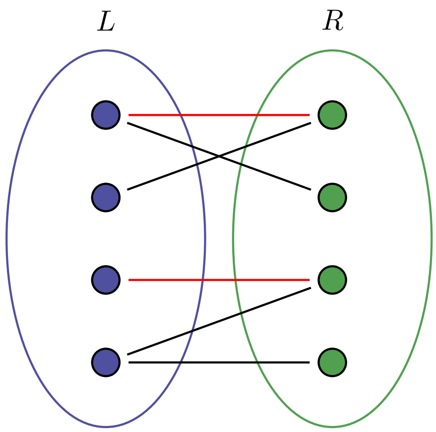


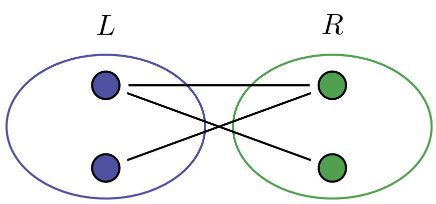
Figure 14.2.2: Adversary setting

In this adversary setting, vertex edge comes with two neighbors and after it's matched with $u \in L$, and a new vertex with online one neighbor as the just matched u arrives. Clearly, for the offline algorithm, we can get a matching of size $|R|$, but here only $\frac{|R|}{2}$ for an online fashion. Thus, the C.R. is 2.

Greedy Algorithm: For every $v \in R$, pick an arbitrary unmatched neighbor. This algorithm gives a maximal matching $\Rightarrow C.R. = 2$ in the case above.

14.3 Randomized Algorithm

Randomized Greedy Algorithm: when $v \in R$ arrives, match it to a random unmatched neighbor.



In the similar setting as we have discussed for Greedy algorithm, the expected number of matching of first vertex $v \in R$ is 1 and since it's randomized, the expected number of matching of second vertex is $\frac{1}{2}$. So the expected number of matching is $\frac{3}{2}$ for this graph. And we have the C.R.$= \frac{2}{3/2} = \frac{4}{3}$.

Here we give a tighter C.R. of the randomized greedy algorithm.
Consider the following case that $|L| = |R| = 2n$:
The maximum matching is $2n$ but our graph has first n vertices of L and the first n vertices of R forming a fully connected graph, as shown in the figure in next page. When $v \in R$ of the first half arrives, if it's matched to the neighbor in the second half of L, the corresponding vertex in the second half can be matched later. Thus we have an expected number of second half matching: $\frac{1}{n+1} + \frac{1}{n} + ... + \frac{1}{2} = H_{n+1} - 1$
The expected number of matching of this greedy algorithm is $\leq n + log_n$ and $OPT = 2n$. Therefore C.R. $\geq 2 - o(1)$

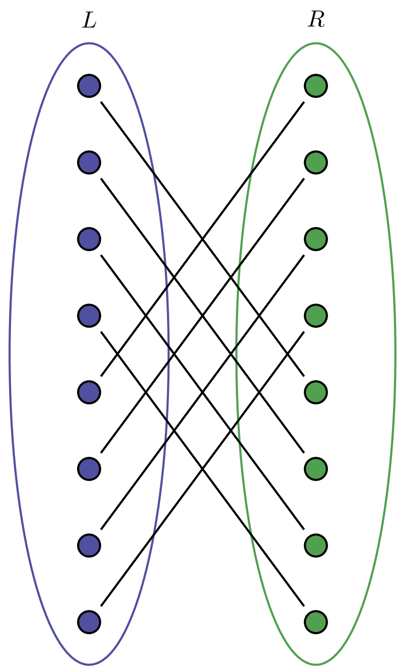


Figure 14.3.3: The intended Matching

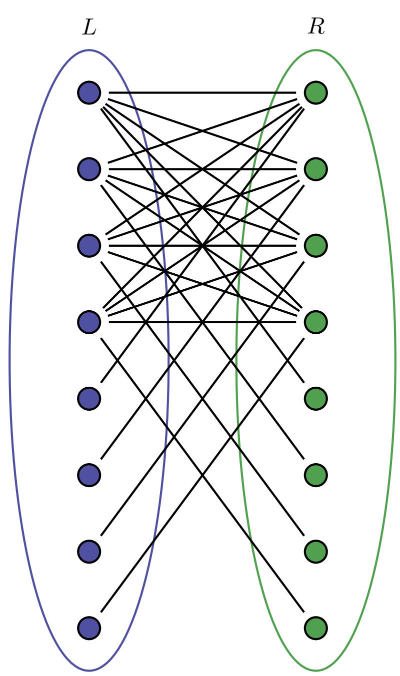


Figure 14.3.4: The graph we get

14.4 The RANKING algorithm

In [1], the authors give an algorithm with an expected competitive ration of $1 - 1/e \approx 0.63$ for solving the online bipartite matching.

RANKING(Karp, Vazirani, Vazirani '90):

1. Pick a uniformly random permutation for nodes
2. When $v \in R$, match it to highest ranked unmatched neighbor.

Theorem 1: The Competitive Ration of RANKING is $1 - 1/e \approx 0.63$.

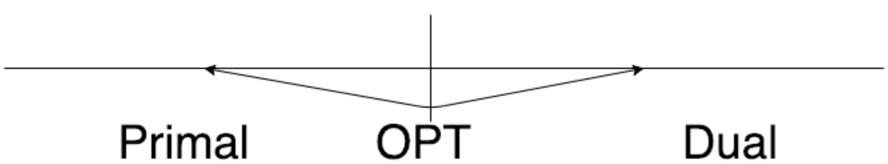
Theorem 2: No randomized algorithm can do better.

14.5 Primal Dual Matching Algorithm

Let X_e be the indicator for e being in a matching. Then we can have the linear programming Primal and its Dual.

$$
\begin{array}{rr}
 & \text{Primal} \\
\max & \Sigma_{(u,v)\in E} X_{uv} \\
\text{subject to} & \Sigma_{v\in R} x_{uv} \leq 1, \quad \forall u \in L \\
 & \Sigma_{u\in L} x_{uv} \leq 1, \quad ,\forall v \in R \\
 & x_{u,v} \geq 0, \quad \forall (u,v) \in E
\end{array}
\qquad
\begin{array}{rr}
 & \text{Dual} \\
\min & \Sigma_{u\in L} \alpha_u + \Sigma_{v\in R} \beta \\
\text{subject to} & \alpha_u \geq 0, \quad \forall u \in L \\
 & \beta_v \geq 0, \quad ,\forall v \in R \\
 & \alpha_u + \beta_v \geq 1, \quad \forall (u,v) \in E
\end{array}
$$

So we have :



And based on the Primal-Dual linear programming, we can modify the RANKING algorithm. This algorithm uses real number but here we don't care about how to store or implement the real numbers.

Modified RANKING:

1. Pick $\forall u \in L$, independently assign value $y_u \in [0, 1]$

2. When $v \in R$ arrives, match it to smallest y_u unmatched neighbor.

3. If u is matched to v, set $\alpha_u := g(y_u)$ and $\beta_v := 1 - g(y_u)$, else set $\beta_v := 0$

Notice that this algorithm can't always guarantee the feasible solution, but we can randomly average the dual solution.

Claim 1: Primal is feasible
Claim 2: Dual cost $\leq$F, which is Primal value
Claim 3: Dual is feasible in expectation $\forall(u,v) \in E$, $\mathbb{E}[\alpha_u] + \mathbb{E}[\beta_v] \geq 1$ $\overline{\beta_v} - \beta$ value of v in the absence of v. Then $\beta_v \geq \overline{\beta} = F(1 - g(y_{u'}))$
If $y_u < y_{u'}$, then $\alpha_u = g(y_u)$
$E[\alpha_u] + E[\beta_v] = E[\alpha_u | y_u < y_{u'}] Pr[y_u < y_{u'}] + F(1 - g(y_{u'})) = F \int_0^{y_{u'}} g(y_u) dy_u + F(1 - g(y_{u'})) \geq$ $(want)$ 1 Regradless of $y_{u'}$.
The proof of that in expectation the dual is feasible[1]:
Proof: Suppose that in the graph $G \setminus \{i\}$ that j gets matched to i' and we reintroduce i. Then

1. $\beta_j \geq \frac{1-g(Y_{i'})}{F}$

2. If $Y_i < Y_{i'}$, then i gets matched and $\alpha_i = \frac{g(Y_i)}{F}$

Now we would like the following inequality to hold for any $\theta \geq 1$.

$$\mathbf{E}[\alpha_i + \beta_j] \geq \int_0^\theta \frac{g(y)}{F} dy + \frac{1 - g(\theta)}{F} (\text{Fix} Y_{i'} = 0)$$

Then we are going to optimize for the largest value of F such that

$$G(\theta) - G(0) + 1 \frac{dG(\theta)}{\theta}$$

By solving this differential equation with boundary term $g(0) = 1$, we get the largest value of

$$F = 1 - \frac{1}{e}, \text{and } g(\theta) \frac{e^\theta}{e}$$

■

14.6 Next Time

We will continue on the RANKING algorithm modified based on Primal/Dual. And we will begin the phone sectary problem.

References

[1] Karp, Richard M., Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, ACM, 1990.

[1] Here I borrow from the previous lecture of CS 880 given by Professor Shuchi Chawla