Algorithms and Uncertainty, Winter 2025/26	Lecture 7 (5 pages)

Online Bipartite Matching

Thomas Kesselheim	Last Update: November 2, 2025

1 Problem Statement

This time, we will consider online maximum bipartite matching. Consider a bipartite graph $G = (L \cup R, E)$, that is, there are no edges within L or within R. A matching $M \subseteq E$ is a subset of the edges such that for each vertex at most one incident edge is included. Let OPT be a maximum matching, that is, a matching that contains the maximum number of edges. There are algorithms to compute OPT in polynomial time. For example, there is an easy reduction to the maximum-flow problem.

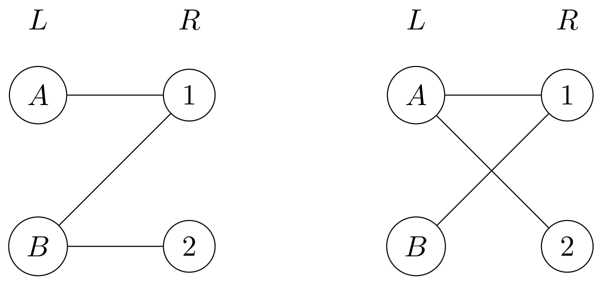
In the online variant of the problem, we do not know the graph from the start but it is revealed to us over time as follows. We know the set L from the start. Now, in each round one vertex r from R is revealed to us including its incident edges. We now have to decide immediately and irrevocably if we want to select one of these edges. If we decide to leave r unmatched, we cannot match it later. If we decide to match it to some $\ell \in L$, then we can neither match any other $r' \in R$ to ℓ nor match r to some other $\ell' \in L$ later on.

Note that this problem is now a maximization problem. Therefore, we will have to flip the sign in the definition of the competitive ratio. We say that a randomized algorithm for a maximization problem with objective function v is strictly α-competitive if

$$\mathbf{E}\left[v(\mathrm{ALG}(\sigma))\right] \geq \alpha \cdot v(\mathrm{OPT}(\sigma)) \quad \text{for any sequence } \sigma \ ,$$

where σ may not depend on the internal randomness of the algorithm.

Example 7.1. *Consider the following example, which already serves as an impossibility result for deterministic algorithms. First, we see only offline vertices A and B and online vertex 1, which we can match to both of them. Then online vertex 2 arrives, which can be matched to only one of A and B.*



2 Greedy Algorithm

Let us consider the following simple greedy algorithm for this problem. Index the vertices in L arbitrarily from 1 to $|L|$. Whenever a vertex $r \in R$ is revealed, if there is still an unmatched neighbor, match it to the unmatched neighbor of smallest index.

Theorem 7.2. *Consider any bipartite graph G and any arrival order of R. Let ALG be the set of edges chosen by the greedy algorithm. Then $|ALG| \geq \frac{1}{2}|OPT|$.*

Proof. We will use a proof approach which is actually using LP duality. However, we state it without explicit reference to duality and rather interpret it as a charging argument. For $\ell \in L$, $r \in R$, we set

$$\alpha_\ell = \begin{cases} 1 & \text{if } \ell \text{ is matched in ALG} \\ 0 & \text{otherwise} \end{cases} \qquad \beta_r = \begin{cases} 1 & \text{if } r \text{ is matched in ALG} \\ 0 & \text{otherwise} \end{cases}$$

Observe that by this definition $\sum_{\ell \in L} \alpha_\ell + \sum_{r \in R} \beta_r = 2|\text{ALG}|$. So if we can show that $|\text{OPT}| \leq \sum_{\ell \in L} \alpha_\ell + \sum_{r \in R} \beta_r$, we are done.

To this end, observe that for any $(\ell, r) \in E$, we have $\alpha_\ell + \beta_r \geq 1$. Suppose otherwise, that is, $\alpha_\ell = 0$ and $\beta_r = 0$. This means that $r \in R$ does not get matched in ALG although $\ell \in L$ would be available. This is a contradiction to the greedy property.

Now we have

$$|\text{OPT}| \leq \sum_{(\ell,r) \in \text{OPT}} (\alpha_\ell + \beta_r) \leq \sum_{\ell \in L} \alpha_\ell + \sum_{r \in R} \beta_r \ ,$$

where in the last step we use that each ℓ and r is matched at most once in OPT. □

One way, we can understand this analysis is as follows: For every edge that our algorithm adds to its matching, we put 2 units of money on a bank account. Afterwards, we can pay 1 unit of money to every edge in the offline optimum.

3 Ranking Algorithm

We will now analyze a slight modification of the Greedy algorithm for which the guarantee will be better. This algorithm is called Ranking. It was introduced by Karp, Vazirani, and Vazirani in 1990. The analysis that we will consider was given by Devanur, Jain, and Kleinberg in 2013.

Ranking:
Index the vertices in L randomly from 1 to $|L|$. That is, draw one of the $n!$ permutations at random and index L accordingly. Whenever a vertex $r \in R$ is revealed, if there is still an unmatched neighbor, match it to the unmatched neighbor of smallest index.

Theorem 7.3. Ranking *is strictly* $1 - \frac{1}{e}$*-competitive.*

Before we get to the proof of this theorem, we start with a useful observation regarding the role of a single offline vertex $\ell \in L$. Namely, we compare the two executions of the algorithm, once on G and once on $G \setminus \{\ell\}$, in which we remove ℓ and all its incident edges. The observation is that without ℓ the situation for our algorithm will only be more difficult as the number of possible matches for a node can not increase.

Lemma 7.4. *Fix any order of the offline vertices and let $\ell \in L$. For each round t, let $U_t \subseteq L$ be the set of unmatched offline vertices in the execution on G after round t and $U'_t \subseteq L$ be the ones on $G \setminus \{\ell\}$. Then, for all t, we have $U_t \supseteq U'_t$.*

Proof. Let $U_0 = L$, $U'_0 = L \setminus \{\ell\}$ be the initial sets of available offline vertices. We show $U_t \supseteq U'_t$ for all $t \geq 0$ by induction. The base case $t = 0$ follows from the definition.

For the induction step, we compare the matches made in the t-th round of the algorithm. Recall that the respective neighbor of smallest index in U_{t-1} or U'_{t-1} respectively will be used for the match.

We now make a case distinction. First consider the case that the algorithm on G matches the newly arrived online vertex to an offline vertex in $U_{t-1} \setminus U'_{t-1}$ or not at all. Then all of U'_{t-1} will still remain unmatched in the execution on G and we have $U_t \supseteq U'_{t-1} \supseteq U'_t$ as by the induction hypothesis $U_{t-1} \supseteq U'_{t-1}$. Otherwise, it uses some vertex ℓ' in U'_{t-1}. The execution on $G \setminus \{\ell\}$ will use the same ℓ' because its rank is smallest among all neighbors in U_{t-1}, so in particular among all neighbors in U'_{t-1}. Now $U_t = U_{t-1} \setminus \{\ell'\}$ and $U'_t = U'_{t-1} \setminus \{\ell'\}$ and therefore $U_t \supseteq U'_t$. □

Now we are ready to prove Theorem 7.3.

Proof of Theorem 7.3. We will use an alternative approach to define the indexing. For each $\ell \in L$, draw Y_ℓ independently uniformly at random from $[0, 1]$ and order L by increasing value of Y_ℓ. Observe that because the random variables $(Y_\ell)_{\ell \in L}$ are independent and identically distributed, each indexing now has the same probability.

Let us first get an overview of the proof approach. The idea is the same as in the proof of Theorem 7.2. Which edges are selected now, of course, is random, depending on $(Y_\ell)_{\ell \in L}$. Therefore also the values α_ℓ and β_r will be random variables now. We define them based on an appropriately chosen function $g\colon [0, 1] \to [0, 1]$ and an appropriate choice of F as follows. Whenever $\ell \in L$ is matched to $r \in R$, we set

$$\alpha_\ell = \frac{g(Y_\ell)}{F} \quad \beta_r = \frac{1 - g(Y_\ell)}{F} \quad .$$

If ℓ or r remain unmatched, set $\alpha_\ell = 0$ or $\beta_r = 0$ respectively.

Observe that this recovers the proof of Theorem 7.2 by setting $g(y) = \frac{1}{2}$ for all y and $F = \frac{1}{2}$. The idea of the proof is to choose g in a smarter way so that F can be set to $1 - \frac{1}{\mathrm{e}}$.

Regardless of the choice of g, observe that for every $(\ell, r) \in \mathrm{ALG}$, we now have $\alpha_\ell + \beta_r = \frac{1}{F}$, that is $\sum_{\ell \in L} \alpha_\ell + \sum_{r \in R} \beta_r = \frac{|\mathrm{ALG}|}{F}$.

This hold *pointwise* for any random outcome. Therefore, we can take the expectation on both sides and get

$$\mathbf{E}\left[\sum_{\ell \in L} \alpha_\ell + \sum_{r \in R} \beta_r\right] = \mathbf{E}\left[\frac{|\mathrm{ALG}|}{F}\right] = \frac{1}{F}\mathbf{E}\left[|\mathrm{ALG}|\right] \quad .$$

Intuitively, this means that we are putting $\frac{1}{F}$ units of money into our bank account for every edge the algorithm adds to its matching.

Due to our choice of g and F, we will be able to show that for any $(\ell, r) \in E$, we have $\mathbf{E}[\alpha_\ell] + \mathbf{E}[\beta_r] \geq 1$. This will then show

$$|\mathrm{OPT}| \leq \sum_{(\ell,r) \in \mathrm{OPT}} (\mathbf{E}[\alpha_\ell] + \mathbf{E}[\beta_r]) \leq \sum_{\ell \in L} \mathbf{E}[\alpha_\ell] + \sum_{r \in R} \mathbf{E}[\beta_r] = \mathbf{E}\left[\sum_{\ell \in L} \alpha_\ell + \sum_{r \in R} \beta_r\right]$$

So *in expectation* the money put into our bank account is enough to pay for the offline optimum. The final balance of our bank account is random but in expectation we break even.

In combination, we get

$$|\mathrm{OPT}| \leq \frac{1}{F}\mathbf{E}\left[|\mathrm{ALG}|\right] \quad ,$$

which shows that our algorithm is F-competitive. So, all we will have to show is that $\mathbf{E}[\alpha_\ell] + \mathbf{E}[\beta_r] \geq 1$ for all $(\ell, r) \in E$. It will turn out that setting $g(y) = \mathrm{e}^{y-1}$ fulfills this property for $F = 1 - \frac{1}{\mathrm{e}}$.

Lemma 7.5. *When setting $g(y) = \mathrm{e}^{y-1}$ and $F = 1 - \frac{1}{\mathrm{e}}$, we have $\mathbf{E}[\alpha_\ell] + \mathbf{E}[\beta_r] \geq 1$ for each $(\ell, r) \in E$.*

Proof. We consider a fixed edge $(\ell, r) \in E$. The values of α_ℓ and β_r are determined by the outcomes of the random variables $(Y_{\ell'})_{\ell' \in L}$. We will keep all $Y_{\ell'}$ for $\ell' \neq \ell$ fixed to arbitrary values and only argue about the value of Y_ℓ. We can do this because the values are drawn independently.

Let us consider the execution of the algorithm on $G \setminus \{\ell\}$, that is, if we remove ℓ from the graph G. We define y^c as follows. If r gets matched in this execution to some ℓ', then set $y^c = Y_{\ell'}$. Otherwise, set $y^c = 1$.

We will show that

$$\mathbf{E}_{Y_\ell}[\alpha_\ell] \geq \frac{1}{F}\left(\mathrm{e}^{y^c-1} - \mathrm{e}^{-1}\right) \qquad \text{and} \qquad \mathbf{E}_{Y_\ell}[\beta_r] \geq \frac{1}{F}(1 - \mathrm{e}^{y^c-1}) \ .$$

Our first observation is that ℓ gets matched whenever $Y_\ell < y^c$. The reason is as follows. If $Y_\ell < y^c$, then by the time r arrives, ℓ could be already matched and we are done. Otherwise, all vertices from R up to this point are matched exactly the same way as if ℓ did not exist. Now, r would be matched to ℓ' if ℓ was not there. But now, as $Y_\ell < y^c = Y_{\ell'}$, r is matched to ℓ instead. So, again ℓ ends up being matched.

Consequently, we get

$$\alpha_\ell \geq \begin{cases} g(Y_\ell)/F & \text{if } Y_\ell < y^c \\ 0 & \text{otherwise} \end{cases}$$

and therefore

$$\mathbf{E}_{Y_\ell}[\alpha_\ell] \geq \int_0^{y^c} \frac{g(y)}{F} dy = \frac{1}{F}\left[\mathrm{e}^{y-1}\right]_{y=0}^{y^c} = \frac{1}{F}\left(\mathrm{e}^{y^c-1} - \mathrm{e}^{-1}\right) \ .$$

Now, let us turn to r. If $y^c = 1$, then $\beta_r \geq \frac{1}{F}(1 - \mathrm{e}^{y^c-1}) = 0$ by definition. So, we can turn to the interesting case that $y^c < 1$. Recall that r is matched to ℓ' and $Y_{\ell'} = y^c$ if ℓ is not present. By Lemma 7.4, r can also be matched to ℓ' when running the algorithm on the graph including ℓ. This means that r is always matched to some ℓ'' with $Y_{\ell''} \leq y^c$, even if vertex ℓ is not excluded. Therefore $\beta_r = (1 - g(Y_{\ell''}))/F \geq (1 - g(y^c))/F$. Consequently, irrespective of Y_ℓ, we have that $\beta_r \geq (1 - g(y^c))/F$

$$\mathbf{E}_{Y_\ell}[\beta_r] \geq \mathbf{E}_{Y_\ell}\left[\frac{1 - g(y^c)}{F}\right] = \frac{1 - g(y^c)}{F} = \frac{1}{F}(1 - \mathrm{e}^{y^c-1}) \ .$$

In combination

$$\mathbf{E}_{Y_\ell}[\alpha_\ell] + \mathbf{E}_{Y_\ell}[\beta_r] \geq \frac{1}{F}\left(\mathrm{e}^{y^c-1} - \mathrm{e}^{-1}\right) + \frac{1}{F}(1 - \mathrm{e}^{y^c-1}) = \frac{1}{F}(1 - \mathrm{e}^{-1}) = 1 \ .$$

Recall that we kept $Y_{\ell'}$ fixed to arbitrary values for $\ell' \neq \ell$. We can now take the expectation over all these random variables on both sides and we are done. □

This concludes the proof of Theorem 7.3. □

In principle, the proof could also be carried out with other functions g. The only property that we require is that for all y^c we need $\int_0^{y^c} \frac{g(y)}{F} dy + \frac{1 - g(y^c)}{F} \geq 1$. Choosing $g(y) = \mathrm{e}^{y-1}$ and $F = 1 - \frac{1}{\mathrm{e}}$ make this inequality tight for every y^c. Therefore, other functions g would lead to a worse F.

In addition, there is also a much stronger impossibility result: Using a generalization of Example 7.1, one can also show that no randomized algorithm is better than $1 - \frac{1}{\mathrm{e}}$-competitive.

References

- R. Karp, U. Vazirani, and V. Vazirani, An Optimal Algorithm for On-line Bipartite Matching, STOC 1990 (original paper)
- N. Devanur, K. Jain, R. Kleinberg, Randomized Primal-Dual analysis of RANKING for Online BiPartite Matching, SODA 2013 (proof structure followed here)