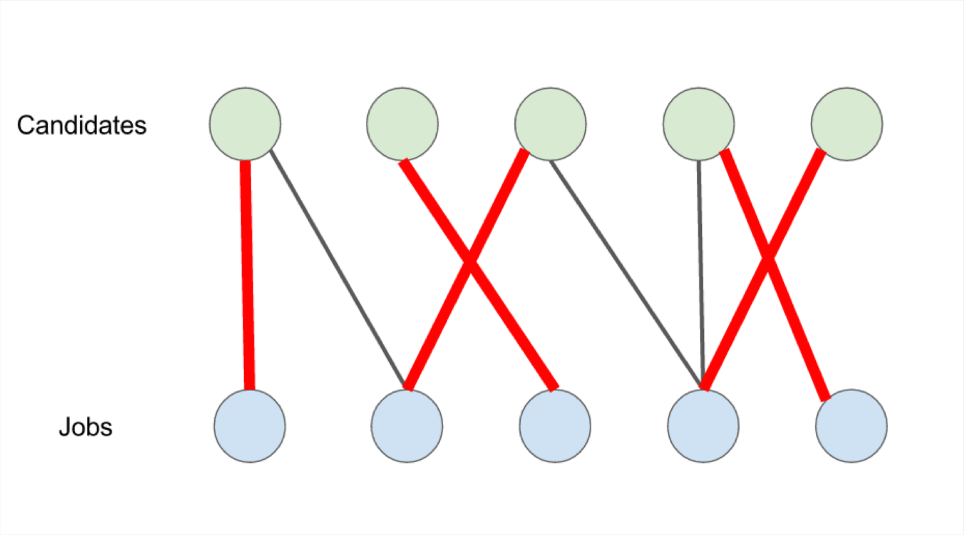
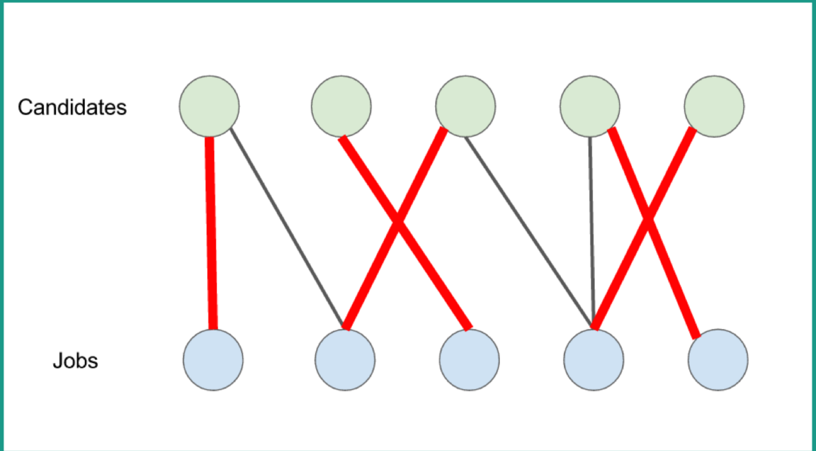
Online Min Cost Matching Overview

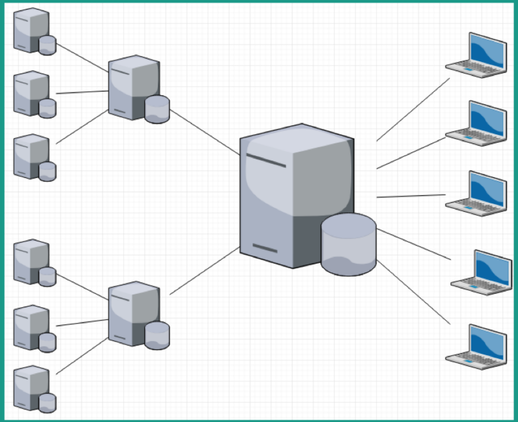


Koosha Jaferian
Winter 2020
Instructor: Prof. Allan Borodin

Outline

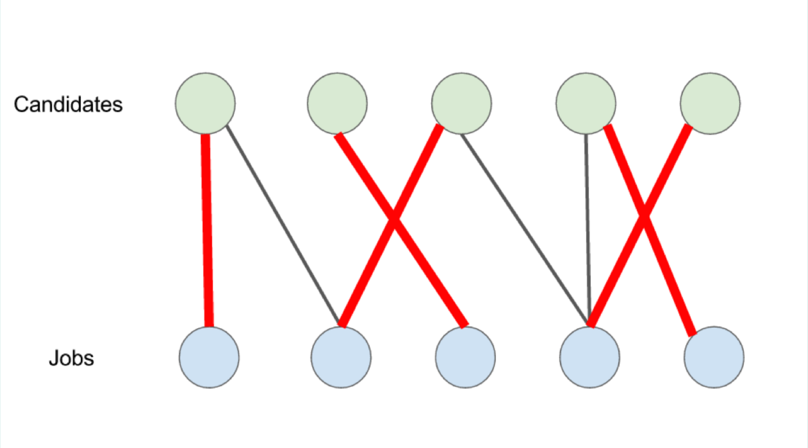
- Reminder from previous presentation
- Models for the problem
- Related Works
- Optimal Algorithm for two models
- Conclusion and Open Problems

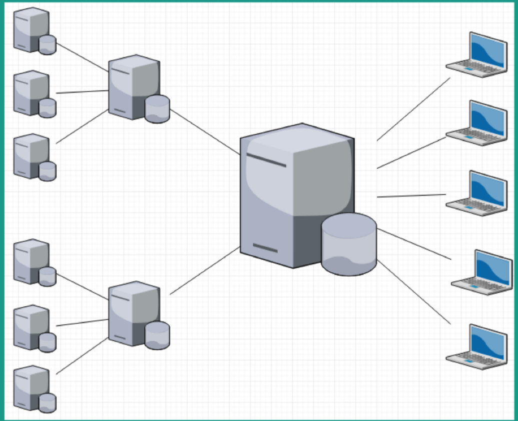




Outline

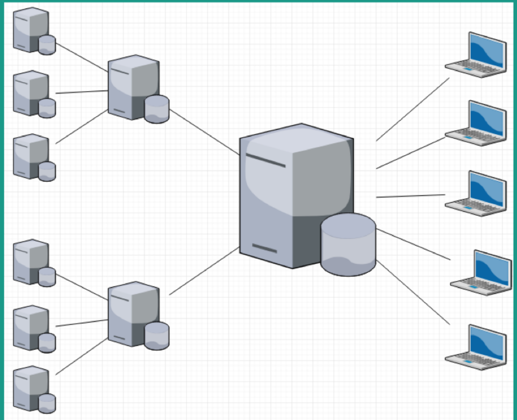
- Reminder from previous presentation
- Models for the problem
- Related Works
- Optimal Algorithm for two models
- Conclusion and Open Problems





Problem Definition

- Complete bipartite graph $G = (R \cup S, E)$
- Positive edge cost c_e for each $e \in E$
- R = requests, and S = servers: $|R| \leq |S|$
- Goal: Match all the requests to a server with the minimum cost.
- $c_M = \sum_{e \in M} c_e$



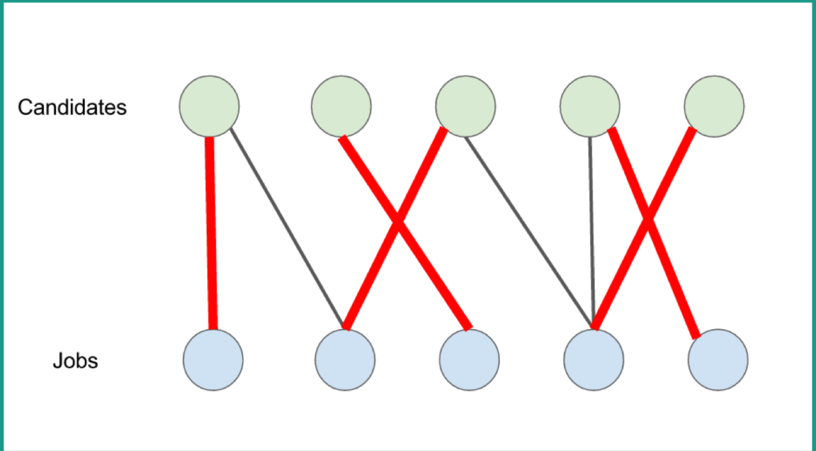
Previous Results

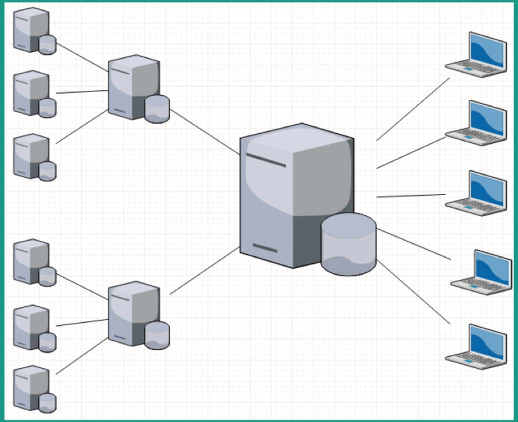
- There is a polynomial-time algorithm for the offline version (Hungarian Method)
- There is no bounded competitive factor for the online version of the problem.

★ Online version with relaxations
  - Metric Graphs → 2n - 1 competitive algorithm
    - Optimal as well
  - Matching on the line → $\theta(log(n))$ competitive algorithm
    - Open problem: Is there any O(1)-competitive algorithm?
  - Other spaces: tree, two dimensional space, etc.

Outline

- ~~Reminder from previous presentation~~
- Models for the problem
- Related Works
- Optimal Algorithm for two models
- Conclusion and Open Problems





How we construct different models?

- Adversary has control over:
  - Location of the requests? (cost of the edges)
  - Random Numbers generated in Randomized Algorithms?
  - Order of arrival for the online nodes (requests)

- Location of the requests
  - Independent?
  - Identically Distributed?
  - Known Distribution?

Adversarial Model

- Adversary has control over:
  - Location of the requests? (cost of the edges) ✅
  - Random Numbers generated in Randomized Algorithms? ✅
  - Order of arrival for the online nodes (requests) ✅

- Location of the requests
  - Independent? ❌
  - Identically Distributed? ❌
  - Known Distribution? ❌

Oblivious Adversary Model

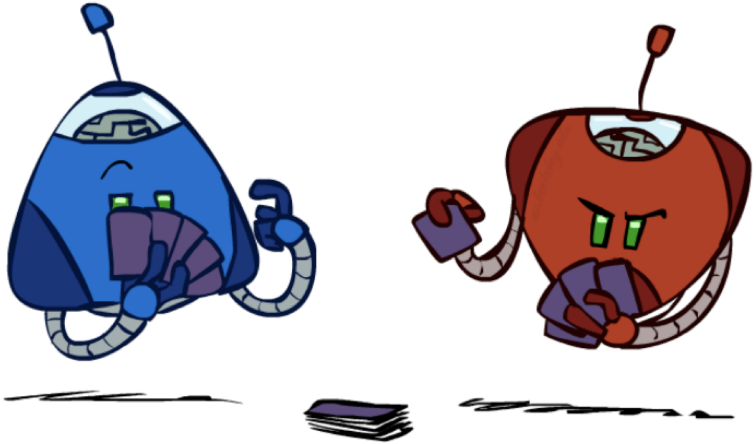
- Adversary has control over:
  - Location of the requests? (cost of the edges) ✓
  - Random Numbers generated in Randomized Algorithms? ✗
  - Order of arrival for the online nodes (requests) ✓

- Location of the requests
  - Independent? ✗
  - Identically Distributed? ✗
  - Known Distribution? ✗

Random Arrival Model

- Adversary has control over:
  - Location of the requests? (cost of the edges) ✓
  - Random Numbers generated in Randomized Algorithms? ✓
  - Order of arrival for the online nodes (requests) ✗

- Location of the requests
  - Independent? ✗
  - Identically Distributed? ✗
  - Known Distribution? ✗



Unknown I.I.D

- Adversary has control over:
  - Location of the requests? (cost of the edges)
  - Random Numbers generated in Randomized Algorithms?
  - Order of arrival for the online nodes (requests)

- Location of the requests
  - Independent?
  - Identically Distributed?
  - Known Distribution?

Known I.I.D

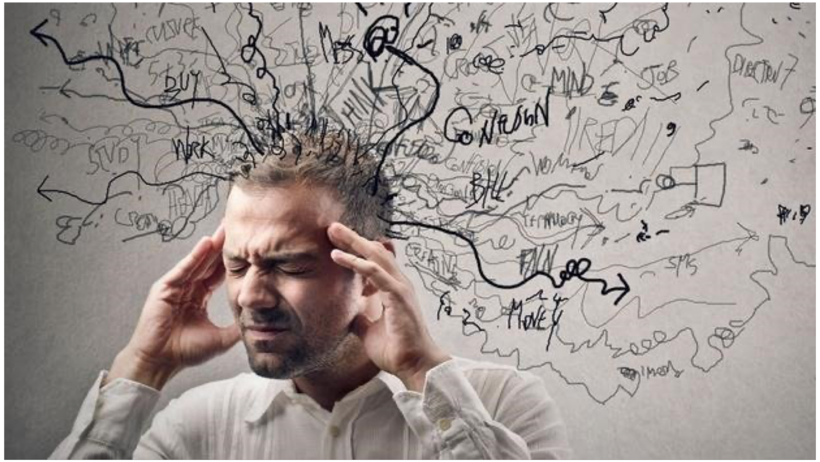
- Adversary has control over:
  - Location of the requests? (cost of the edges)
  - Random Numbers generated in Randomized Algorithms?
  - Order of arrival for the online nodes (requests)

- Location of the requests
  - Independent?
  - Identically Distributed?
  - Known Distribution?

Hardness of these models

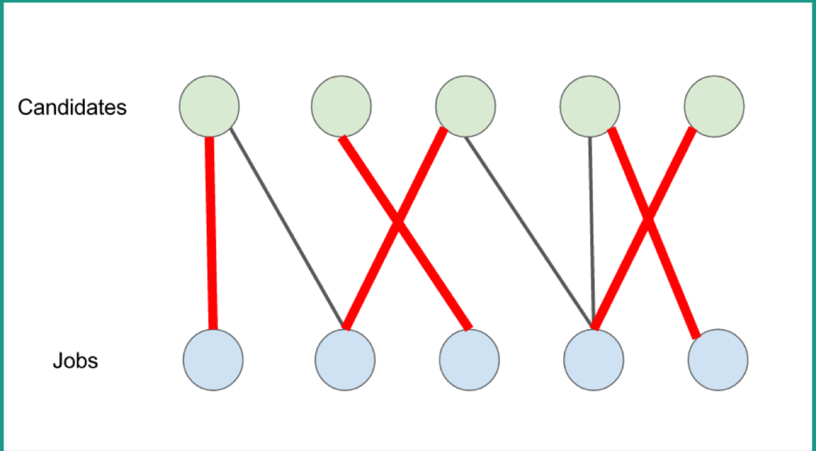
- Raghvendra 2016: The hardness of these five models are as follows:

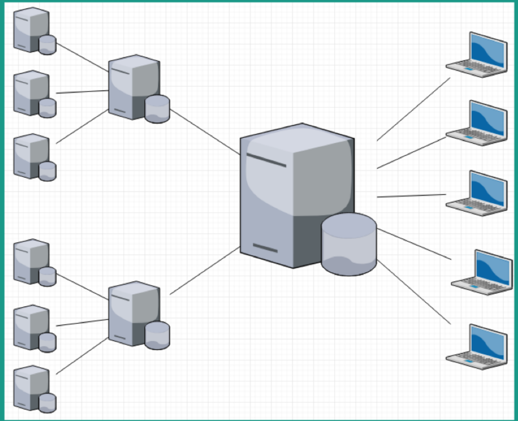
- Adversarial > Oblivious Adversary > Random Arrival > Unknown IID > Known IID



Outline

- ~~Reminder from previous presentation~~
- ~~Models for the problem~~
- Related Works
- Optimal Algorithm for two models
- Conclusion and Open Problems





Works Related to different models (1)

- (Mahdian, Yan 2011): Introducing the random arrival model for maximum matching
- (Karande et al. 2011): Finding a better algorithm for unknown IID for maximum matching.

Model	Adversarial Input	Known Distribution	Unknown Distribution
Lower Bounds (algorithms)	$1-\frac{1}{e}$ [KVV90]	0.67 [FMMM09] 0.699 [BK10] 0.702 [MOGS11]	$1-\frac{1}{e}$ [KVV90] 0.653 [This paper]
Upper Bounds (hardness)	$1-\frac{1}{e}$ [KVV90]	0.998 [FMMM09] 0.902 [BK10] 0.823 [MOGS11]	5/6 [GM08] 0.823 [MOGS11]

- (Bansal et al. 2007): Finding $O(log^2(n))$ algorithm for oblivious adversary model

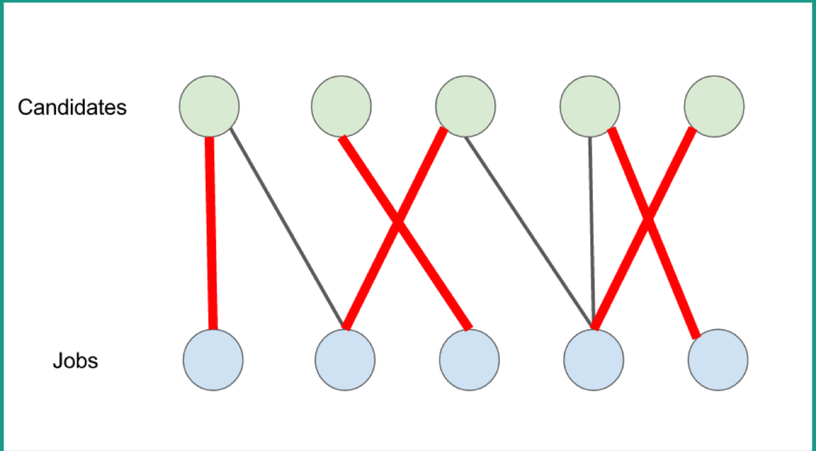
Works Related to different models (2)

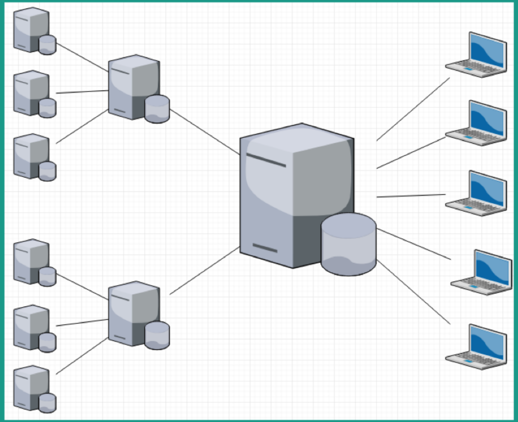
- Kalyanasundaram and Pruhs (1993): 2n-1 competitive ratio for adversarial model
- No factor better than 2n-1

- Can we design an algorithm which is optimal in multiple models?
  - (Raghvendra 2016): Designing an algorithm
    - 2n - 1 competitive in the adversarial model
    - $2H_n - 1 + O(1)$ Competitive in the random arrival model
    - Showing that we can't achieve better than $2H_n - 1 - O(1)$

Outline

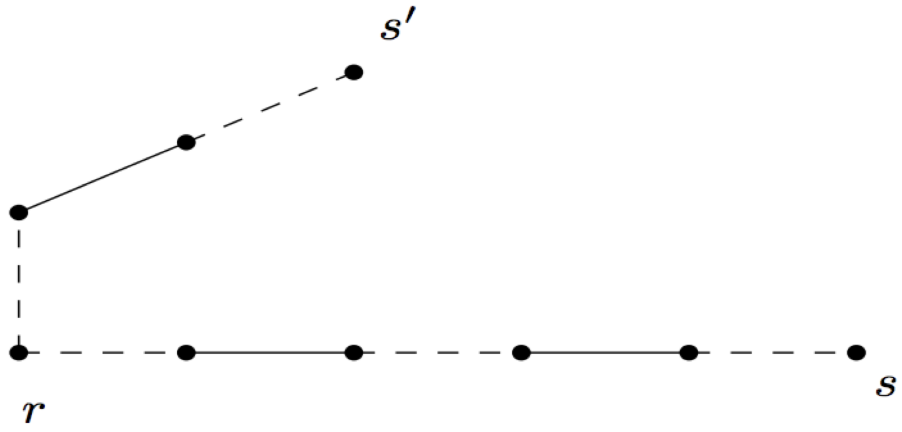
- ~~Reminder from previous presentation~~
- ~~Models for the problem~~
- ~~Related Works~~
- Optimal Algorithm for two models
- Conclusion and Open Problems





Preliminaries (1)

- Matching M* on the graph
- Alternating Path (cycle): a simple path (cycle) whose edges alternate between those in M* and those not in M*
- Free vertex: a vertex which is still unmatched in the graph.
- Augmenting Path: an alternating path between two free vertices.



Preliminaries (2)

- Augment M* along P
  - Remove the edges in P and M* from M*
  - Add the edges in P and not in M* to M*

- t-net cost for the augmenting paths

$$\phi_t(P) = t\left(\sum_{(s,r)\in P\setminus M^*} \mathtt{d}(s,r)\right) - \sum_{(s,r)\in P\cap M^*} \mathtt{d}(s,r)$$

Preliminaries (3)

- t-feasibility concept: a matching M* and a set of dual weights for the vertices shown by y(v) for each vertex

- For every edge between request r and server s, we will have:

$$
\begin{aligned}
y(s) + y(r) &\leq t\mathtt{d}(s, r), \\
y(s) + y(r) &= \mathtt{d}(s, r) \quad \text{for } (s, r) \in M^*.
\end{aligned}
$$

Algorithm (1)

- We maintain a set of dual weights y(v) for each vertex and two matchings M, M*
    - Initialize the weights to 0 and matchings to Ø

Set of free servers in M* (and M) → S_F

- M* with dual weights will remain a t-feasible matching (offline matching)
- M is the online matching

Algorithm. Given a new request r, our algorithm computes the minimum t-net-cost augmenting path P with respect to matching M^*. P starts at r and ends at some free vertex s. The algorithm updates M^* by augmenting it along P, i.e., $M^* \leftarrow M^* \oplus P$. For the online matching M, the algorithm will match the server s to r, i.e., $M \leftarrow M \cup \{(s, r)\}$.

Algorithm (2)

- Invariants of the algorithm:
  - M* and dual weights always form a t-feasible matching
  - All dual weights for the servers are non-positive and for the free servers, they equal to zero.

- How to find minimum t-net cost augmenting path P and update the matchings M, M* and the dual weights in $O(n^2)$ time?

More details

- Construct residual graph G_{M^*} with S U R as vertices, and E_{M^*}
  - Directed graph with s directed to r when $(s,r) \in M^*$
  - Otherwise, r is directed to s

- Assigning costs with the directed edges in E_{M^*}

If $(a,b) \in M^*$, we set the cost of the edge to be the slack $s(a,b) = \mathtt{d}(a,b) - y(a) - y(b)$. From t-feasibility (condition (2)) of M^*, we know the slack of every edge in the matching is $s(a,b) = 0$.

If $(a,b) \notin M^*$, we set the cost of the edge (a,b) to be the slack $s(a,b) = t\mathtt{d}(a,b) - y(a) - y(b)$. From t-feasibility (condition (1)) of M^*, we know $s(a,b) \geq 0$.

Simple observations

- Every edge in E_{M^*} has a non-negative edge cost.
- Set of nodes and edges are identical in G and G_{M^*}
- Directed paths in G_{M^*} correspond to alternating paths in G
- If the two end vertices in directed path are free, it corresponds to an augmenting path

- Use the Dijkstra’s algorithm → minimum cost path from r to any free server
- Directed path $\vec{P}$ is the minimum cost among all the free servers
- P corresponding to $\vec{P}$ in G will be the minimum t-net cost augmenting path.

Dual Weight Updates

- Before augmenting M* along P, we update all the dual weights.
- d is the cost of $\vec{P}$ (d_s)
- For each node v, we will update as follows:

(a) If $d_v \geq d$, then $y(v)$ remains unchanged.
(b) If $d_v < d$, and $v \in R$, then we increase the dual weight $y(v) \leftarrow y(v) + d - d_v$
(c) If $d_v < d$, and $v \in S$, then we decrease the dual weight $y(v) \leftarrow y(v) - d + d_v$.

- Augment M* along P with $M^* \leftarrow M^* \oplus P$
- Update Dual weight for every r' in R ∩ P
  - $y(r') \leftarrow y(r') - (t-1)d(s', r')$

Algorithm Analysis (Brief Idea)

- Assume that the requests are coming by an order $(r_1, r_2, \dots, r_n)$
- (M_i^*, M_i) Are offline and online matchings after the i'th request.
- Let $\ell(P) = \sum_{(s,r)\in P} \mathtt{d}(s, r)$

▶ **Lemma 7.** *Let $t \geq 1$. Let $P_1, \dots, P_n$ be the augmenting paths computed by our algorithm in that order. Then, the t-net-cost of these paths relate to the cost of the online matching as follows:*

(i) $\phi_t(P_i) \leq t\mathtt{d}(s_i, r_i) \leq t\ell(P_i)$.

(ii) $\sum_{i=1}^{n} \phi_t(P_i) \geq ((t-1)/2)w(M) + ((t+1)/2)w(M^*)$.

- Using these inequalities we will prove the competitive ratios.

Intuition behind the Harmonic Number

- Final Competitive ratio for Adversarial Model
  - 2n - 1 + O(1) ✅

$$t = n^2 + 1,$$

- Final Competitive ratio for Random Arrival Model
  - $2H_n - 1 + O(1)$ ✅

In the random arrival model, the input request sequence is a random permutation. Therefore, the i^{th} request can be any one of the remaining $n - i + 1$ requests with the same probability and we have,

$$\mathbb{E}[\phi_t(P_i)] \leq \frac{1}{n-i+1} \sum_{j=1}^{n-i+1} \phi_t(P'_j) \leq \frac{1}{n-i+1} tw(M_{\text{OPT}}).$$

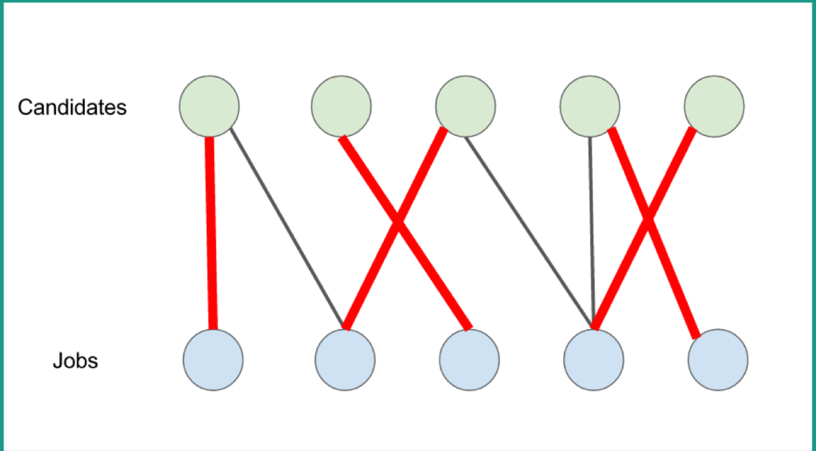
From linearity of expectation,

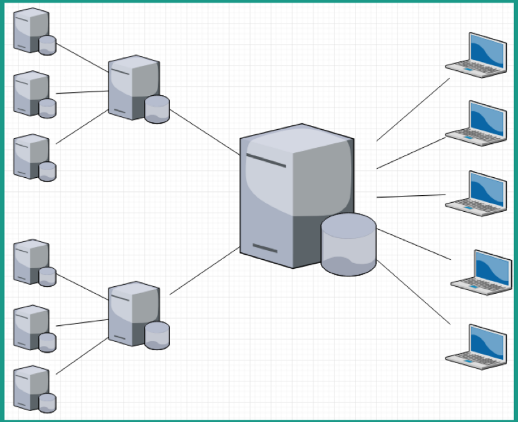
$$\mathbb{E}[\sum_{i=1}^{n} \phi_t(P_i)] \leq \sum_{i=1}^{n} \frac{1}{n-i+1} tw(M_{\text{OPT}}) = t\text{H}_n w(M_{\text{OPT}}).$$

- Where does $H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$ come from?

Outline

- ~~Reminder from previous presentation~~
- ~~Models for the problem~~
- ~~Related Works~~
- ~~Optimal Algorithm for two models~~
- Conclusion and Open Problems

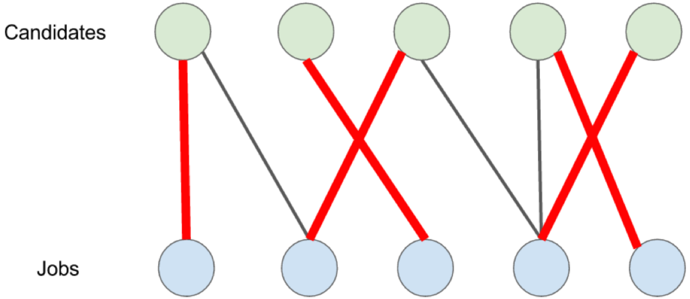




Open Problems?

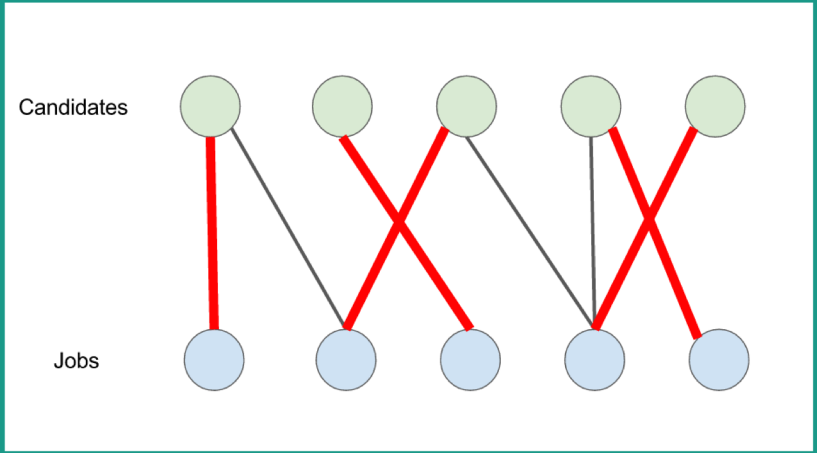
- O(1) competitive algorithm for the matching on the line?
- Extending this approach to k-server problem?
- Extending this approach to the oblivious adversary model?
- Improve the performance in special metrics?
    - Tree?
    - Two dimensional space?
    - Line?
- What happens if we have more servers?

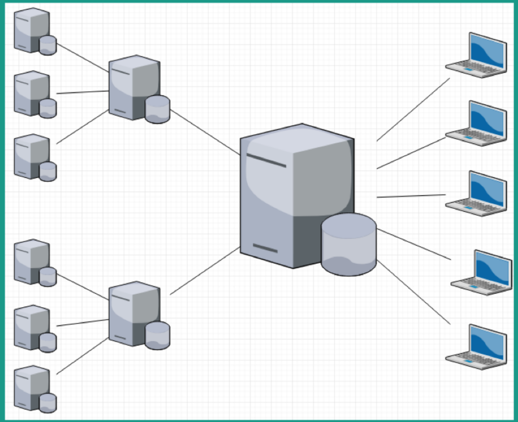
Conclusion



Outline

- ~~Reminder from previous presentation~~
- ~~Models for the problem~~
- ~~Related Works~~
- ~~Optimal Algorithm for two models~~
- ~~Conclusion and Open Problems~~





Any Questions?

Thanks for Your attention!

References

- https://onlinelibrary.wiley.com/doi/epdf/10.1002/nav.3800020109
- https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.2873&rep=rep1&type=pdf
- https://www.sciencedirect.com/science/article/pii/0304397594900426
- https://drops.dagstuhl.de/opus/volltexte/2016/6641/pdf/LIPIcs-APPROX-RANDOM-2016-18.pdf
- http://www.cs.ox.ac.uk/people/elias.koutsoupias/Personal/Papers/paper-kou09.pdf
- https://arxiv.org/pdf/2006.01026.pdf
- https://arxiv.org/pdf/1803.07206.pdf
- http://www.thomas-kesselheim.de/science/onlinematching.pdf
- https://dl.acm.org/doi/pdf/10.1145/1993636.1993716
- https://dl.acm.org/doi/pdf/10.1145/1993636.1993715
- https://www.tau.ac.il/~nivb/download/matching-esa07.pdf

Reference Page