Min-Cost Matching with Delays

Itai Ashlagi[1], Yossi Azar[2], Moses Charikar[1], Ashish Chiplunkar[2], Ofir Geri[1], Haim Kaplan[2], Rahul Makhijani[1], Yuyi Wang[3], and Roger Wattenhofer[3]

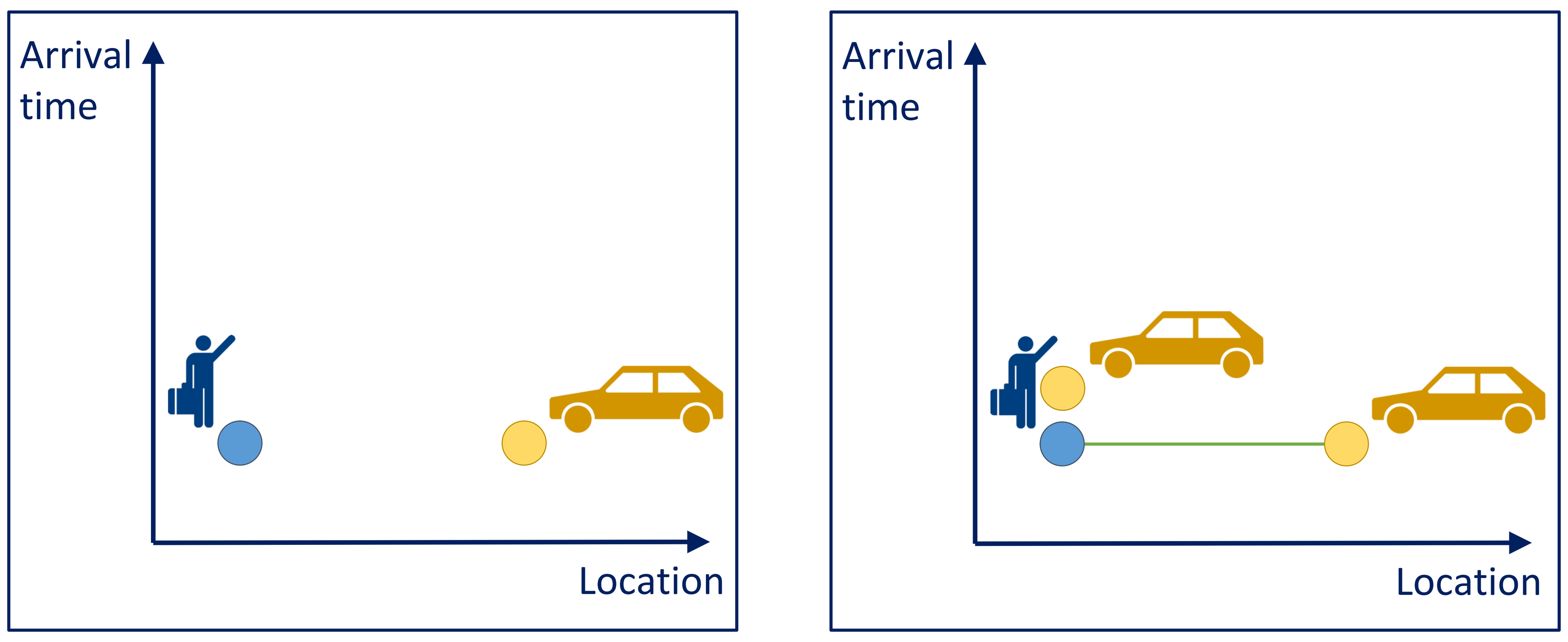
[1]Stanford University, [2]Tel Aviv University, [3]ETH Zürich

Motivation

- Example: matching passengers to drivers in ridesharing platforms (e.g., Lyft)
- How much would you wait before matching?

Would you match immediately?

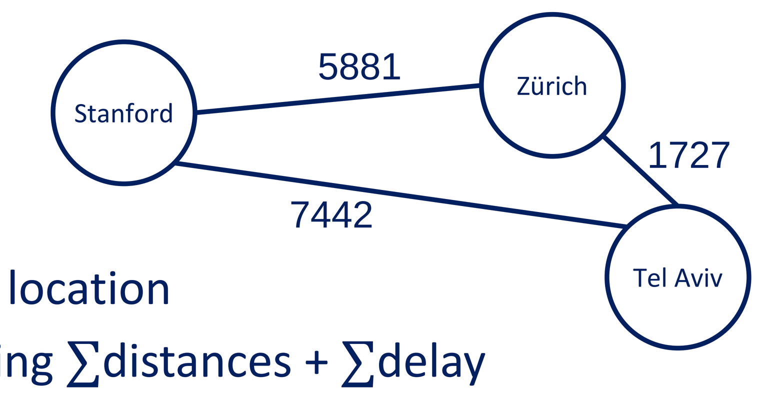
What if another driver arrives shortly after?



Inherent tradeoff:
Cost of distance vs. cost of delay

- In traditional online algorithms, requests are served *upon arrival*.
- Here, service can be delayed, but incurs cost for delay [EKW16].
- Many applications: players in online gaming platforms, kidney exchange, labor markets…

The MPMD Problem



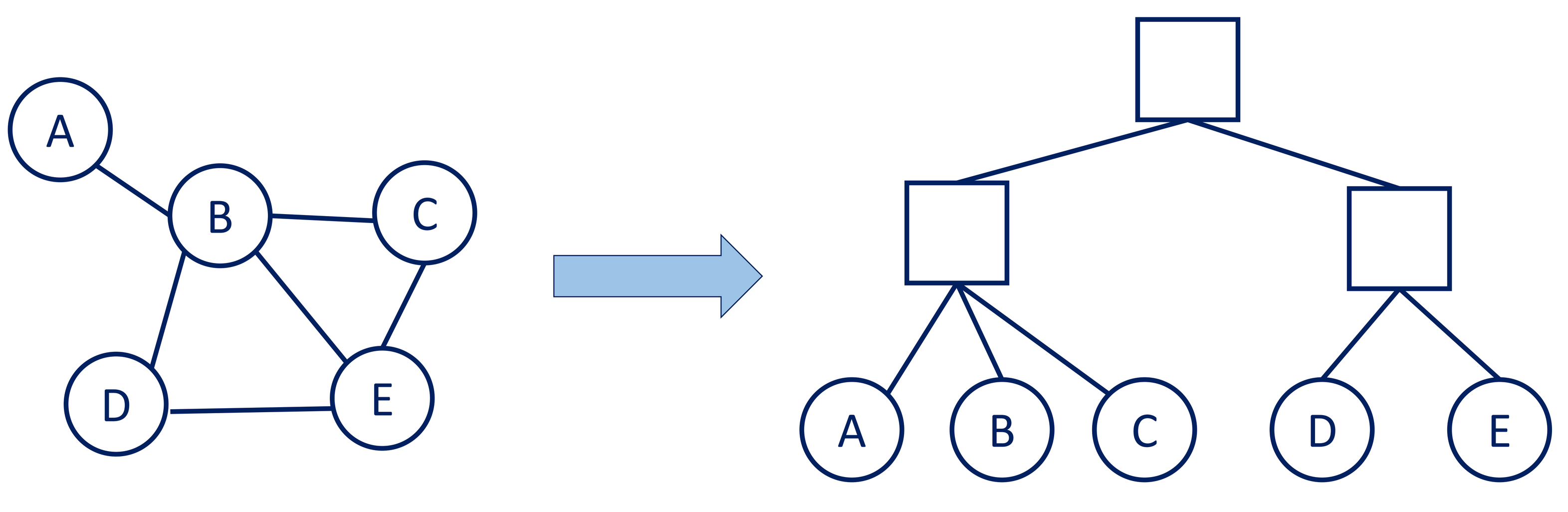
- n-point metric space
  - Distances satisfy triangle inequality
- Requests arrive over time
  - Requests identified by arrival time and location
- Goal: Output a perfect matching minimizing ∑distances + ∑delay
- Our benchmark: the competitive ratio

$$\max_{input\ I} \frac{E[ALG(I)]}{OPT(I)}$$

- Min-cost *bipartite* perfect matching with delays (MBPMD)
  - Each request also has polarity (+ or –)
  - Equal number of requests of each polarity
  - Can only match requests of opposite polarities (+ with –)

Embedding Arbitrary Metrics into Trees

- Standard technique in approximation and online algorithms
- An arbitrary finite metric space can be randomly embedded it into a tree metric such that the distances are distorted by $O(\log n)$ [FRT04].
- The points of the original metric space are the leaves of the tree.
- The height of the trees can be made $O(\log n)$ [BBMN15].
- Allows us to reduce the problem to trees



Our Results

- Definition: ALG is (β, γ)-competitive if for every SOL,
  $$ALG \le \beta \cdot SOL_{dist} + \gamma \cdot SOL_{delay}$$
  where SOL_{dist} is the cost due to the distances and SOL_{delay} is the cost due to the delays in SOL.

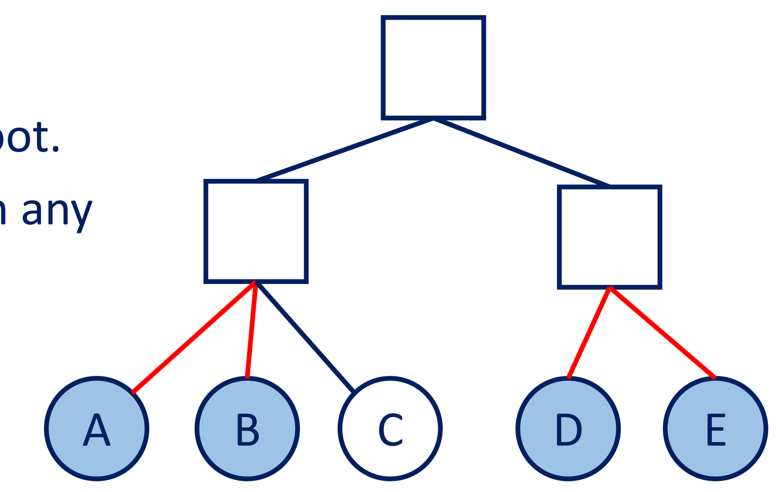
Algorithms

- $\big(O(1), O(height)\big)$-competitive algorithms for MPMD and MBPMD on trees
- Implies $O(\log n)$-competitive algorithms for general metrics

Lower Bounds

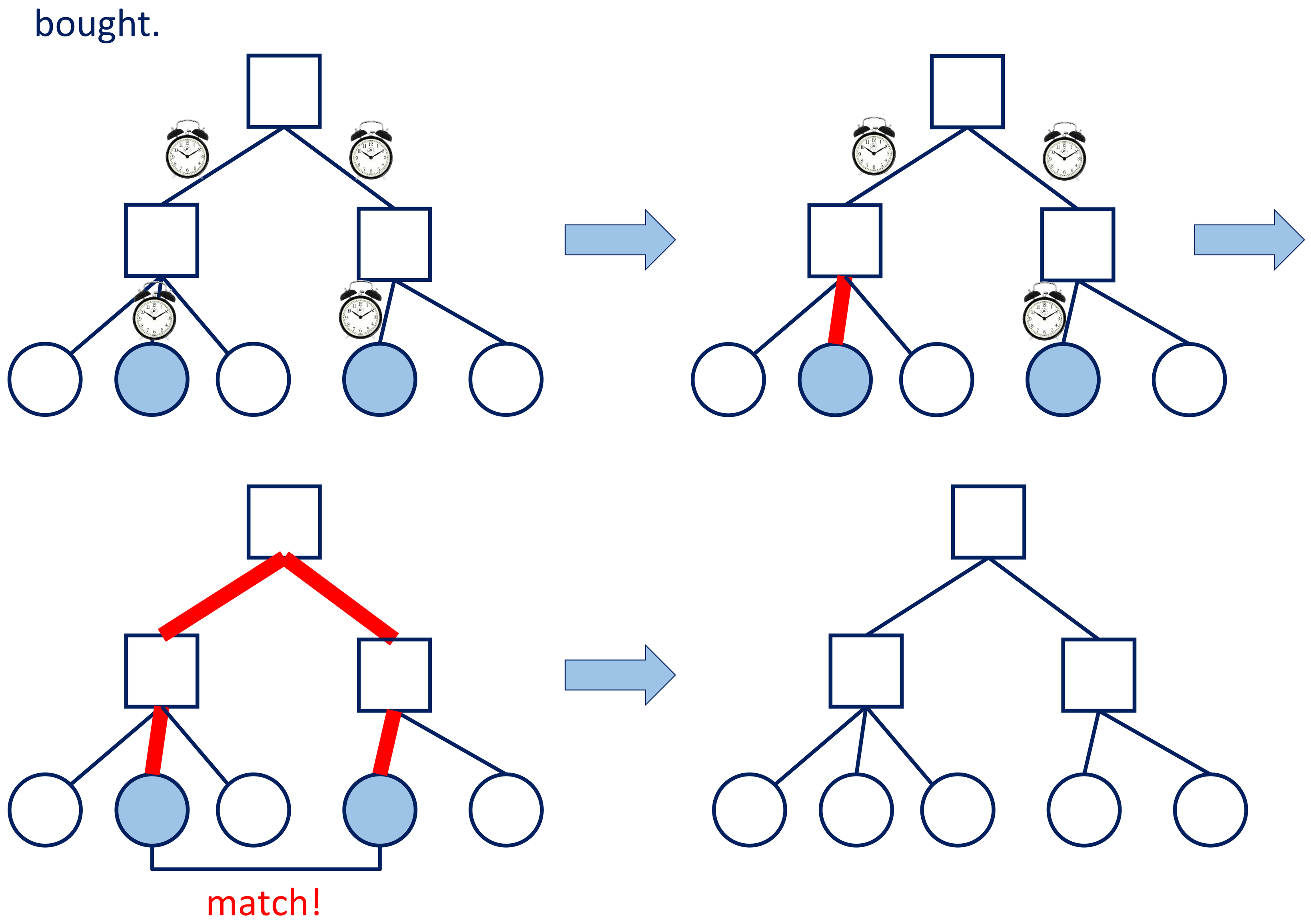
- An $\Omega(\log n / \log\log n)$ lower bound on randomized algorithms for MPMD
- An $\Omega\big(\sqrt{\log n / \log\log n}\big)$ lower bound on randomized algorithms for MBPMD

The Algorithms for Tree Metrics



- Tentative greedy matching
  - Consider vertices from leaves to the root.
  - When considering v, tentatively match any possible unmatched requests under it.
- Optimal when the number of requests is balanced

- We maintain a counter for each edge.
- When an edge is used in the greedy matching, we increase its counter.
  - For MPMD, the rate is uniform.
  - For MBPMD, the rate depends on the imbalance between positive and negative requests under the edge.
- When the counter reaches the weight of the edge, that edge is bought.
- Two requests are matched when the entire path connecting them has been bought.



Extensions and Open Problems

- In MBPMD, there is a gap: $O(\log n)$ vs. $\Omega\big(\sqrt{\log n / \log\log n}\big)$
- Deterministic algorithms for general metrics
- Decentralized markets
- Many-to-one matching (in order to model carpooling)
- Different input assumptions, e.g., more structured metric spaces

References

[AACCGKMWW17] Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul Makhijani, Yuyi Wang, and Roger Wattenhofer. Min-cost bipartite perfect matching with delays. To appear in APPROX'17.
[ACK17] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. SODA'17.
[BBMN15] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. A polylogarithmic-competitive algorithm for the k-server problem. J. ACM, 62(5):40, 2015.
[EKW16] Yuval Emek, Shay Kutten, and Roger Wattenhofer. Online matching: haste makes waste! STOC'16.
[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. J. Comput. Syst. Sci., 69(3):485-497, 2004.