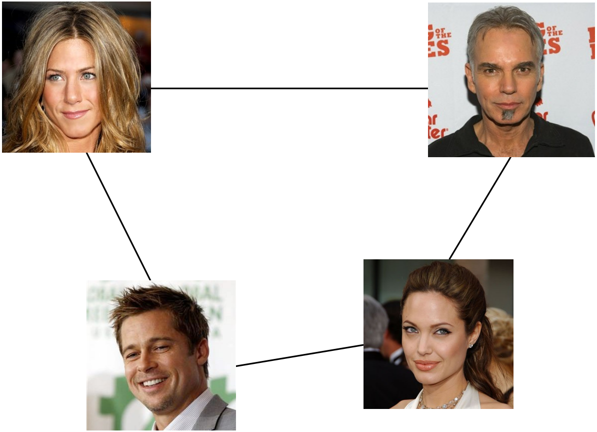
Metric Matching
Cheap or Stable … or Fast?

Roger Wattenhofer

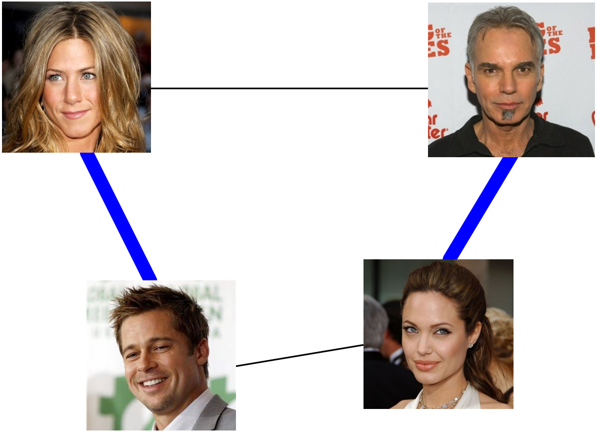
Disclaimer


OSDI
SenSys
HotNets
Multimedia
Ubicomp
PODC
Mobicom
STOC
FOCS
ICALP
SIGCOMM
SPAA
SODA
EC

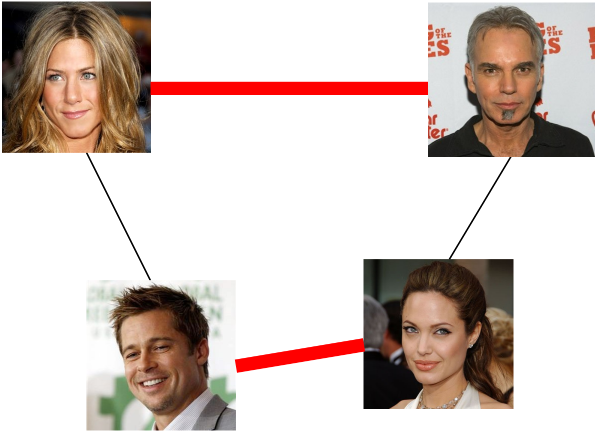
Matchings



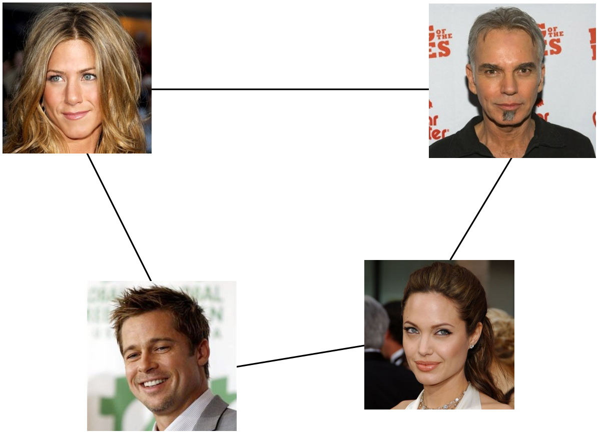
Matchings



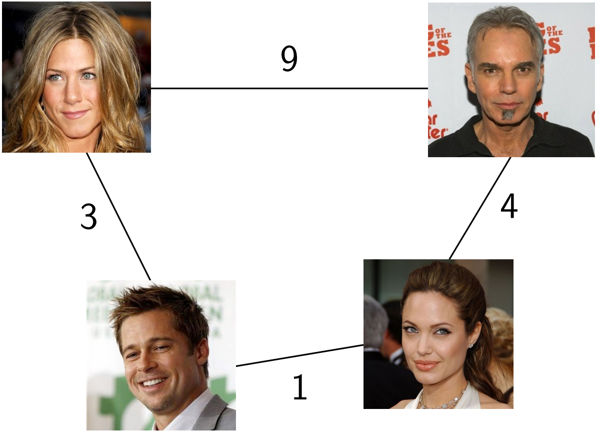
Matchings



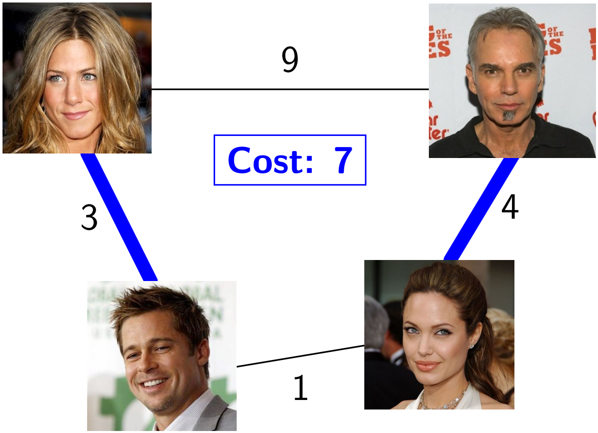
Matchings



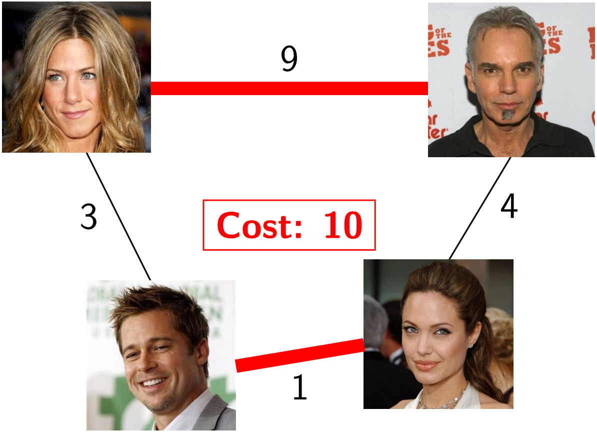
Matchings



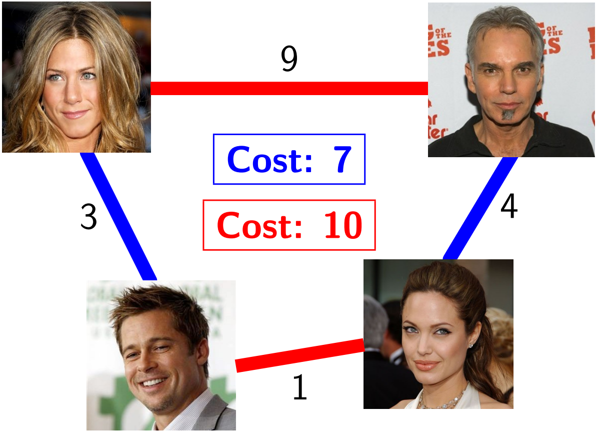
Matchings



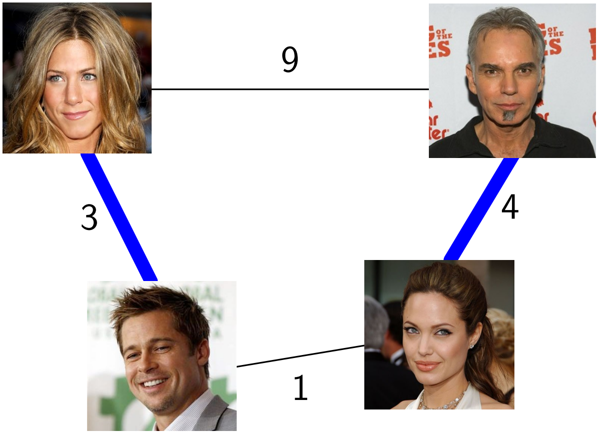
Matchings



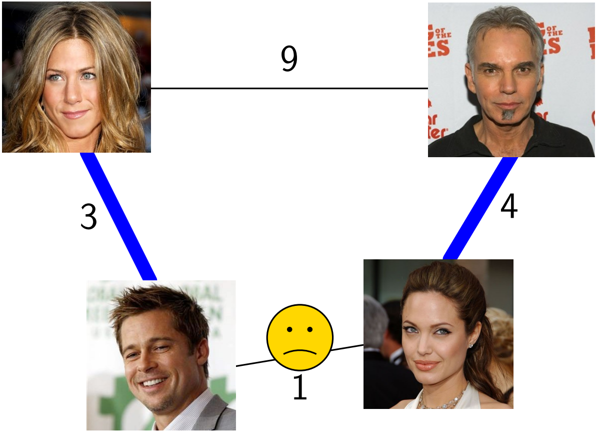
Matchings



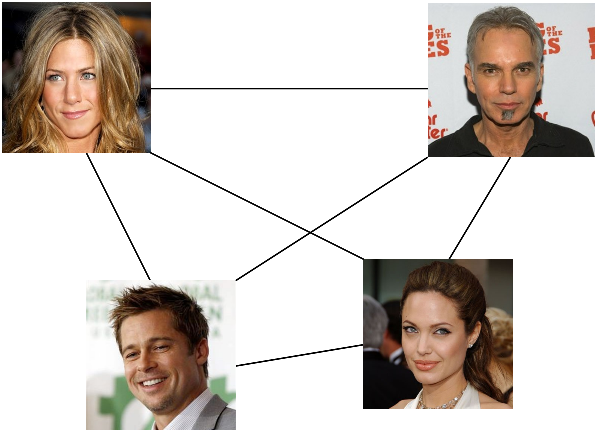
Matchings



Matchings

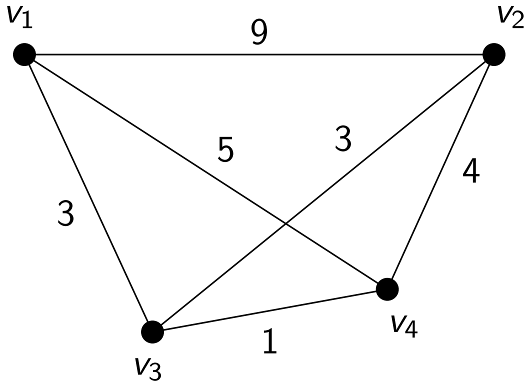


Matchings



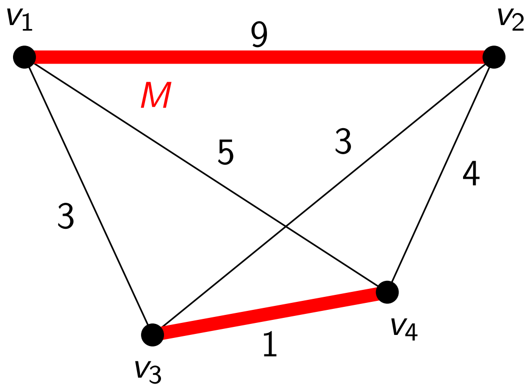
Weighted Perfect Matching

Weighted complete graph $G = (V, V \times V, w)$



Minimum-Cost Perfect Matching

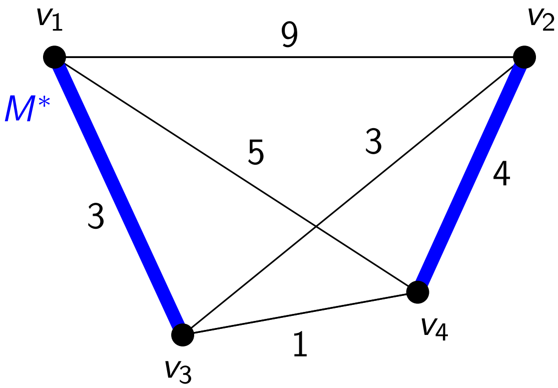
Perfect matching $M \subseteq V \times V$



$$c(M) = 1 + 9 = 10$$

Minimum-Cost Perfect Matching

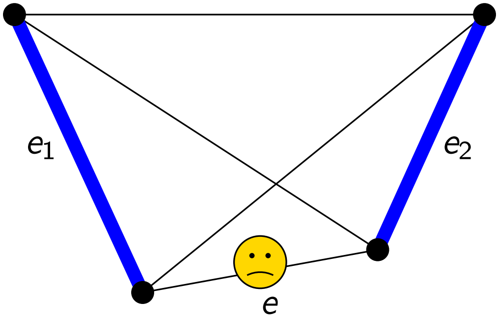
Minimum-cost perfect matching $M^* \subseteq V \times V$



$c(M^*) = 3 + 4 = 7$

Stable Matching

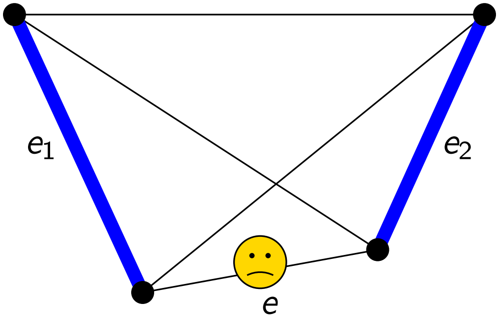
α-unstable edge $e \notin M$



$$w(e) < \frac{1}{\alpha} \cdot \min\{w(e_1), w(e_2)\}$$

Stable Matching

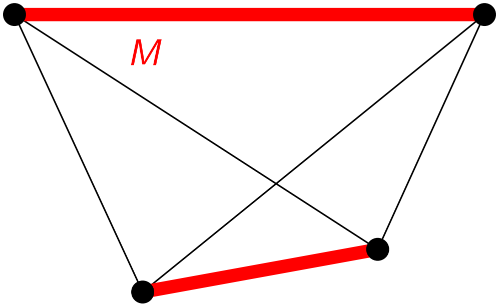
Example: **2-unstable edge** $e \notin M$



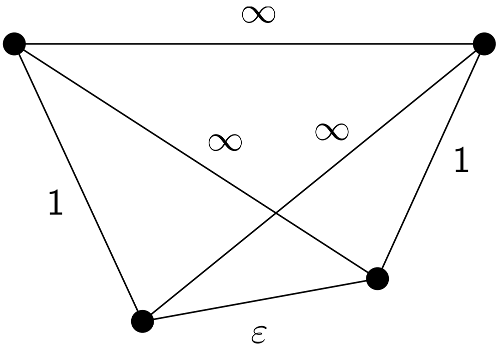
$$w(e) < \frac{1}{2} \cdot \min\{w(e_1), w(e_2)\}$$

Stable Matching

α-stable matching: without α-unstable edge

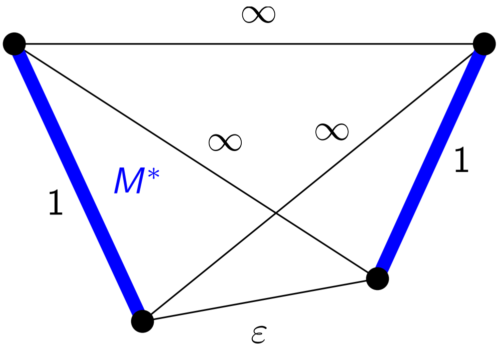


Stable vs. Cheap



Stable vs. Cheap

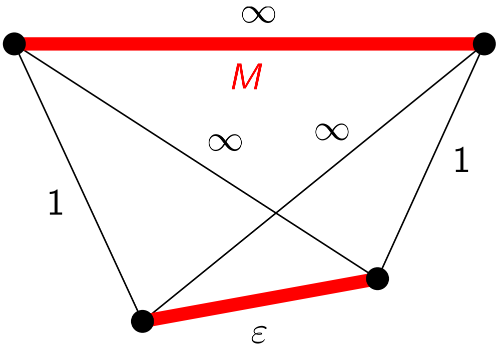
min-cost matching M^*



$c(M^*) = 2$

Stable vs. Cheap

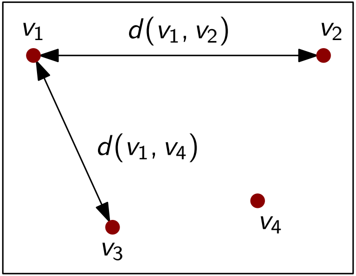
α-stable matching M



$c(M) = \infty$

Metric Graphs

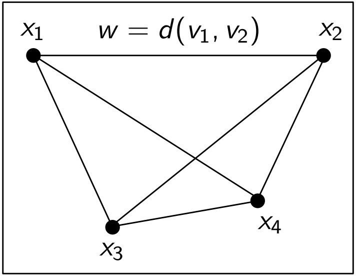
Points in metric



$v_i \sim x_i$

$\Rightarrow$

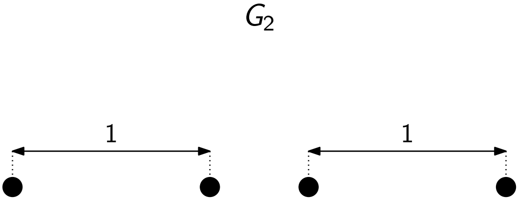
Metric graph G



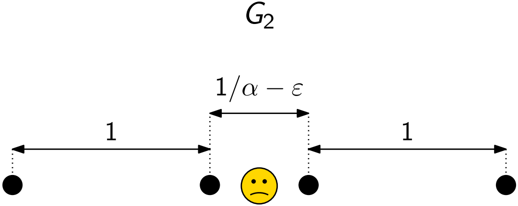
$w((x_i, x_j)) = d(v_i, v_j)$

Stable Matchings Can Be Expensive

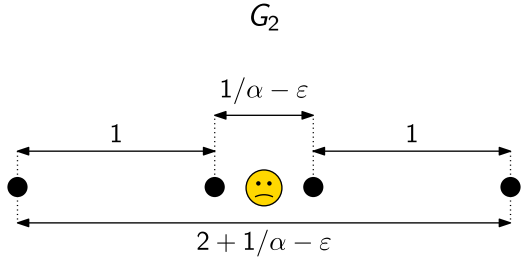
Graph Construction



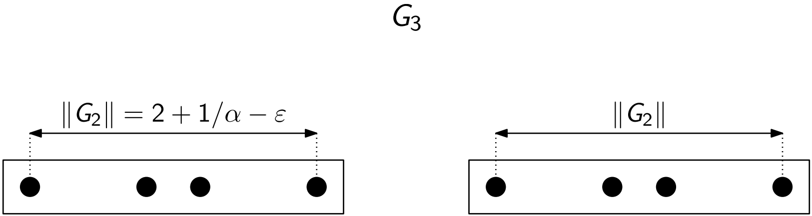
Graph Construction



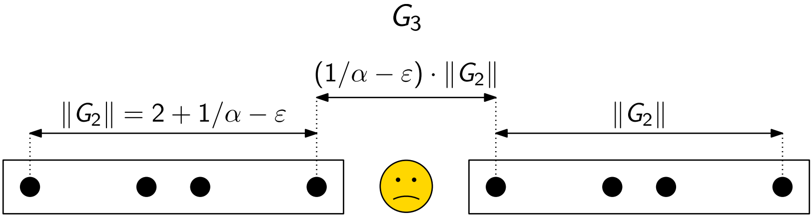
Graph Construction



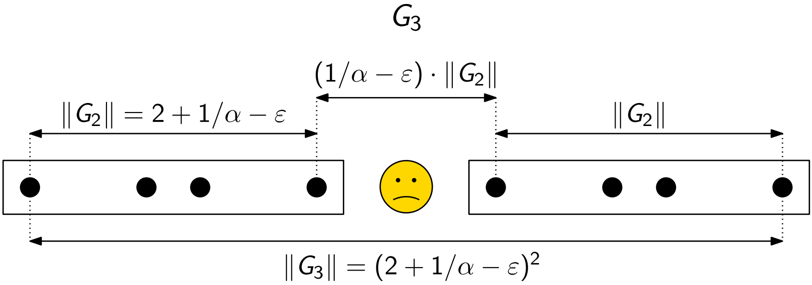
Graph Construction



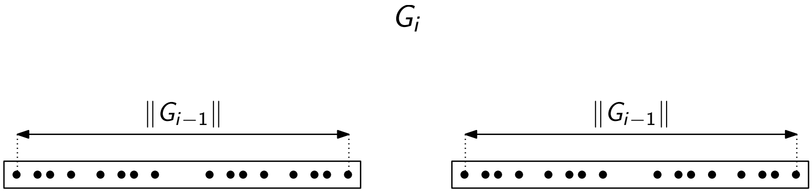
Graph Construction



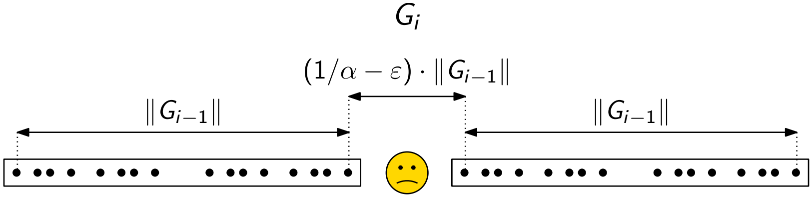
Graph Construction



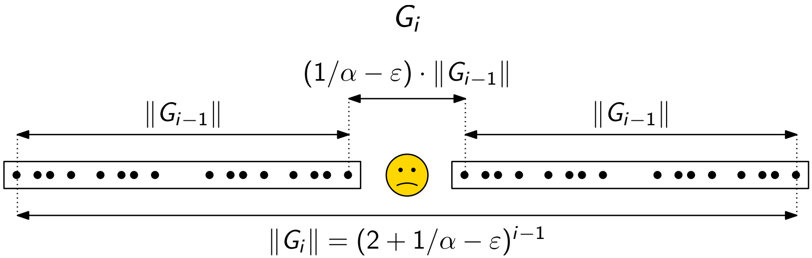
Graph Construction



Graph Construction



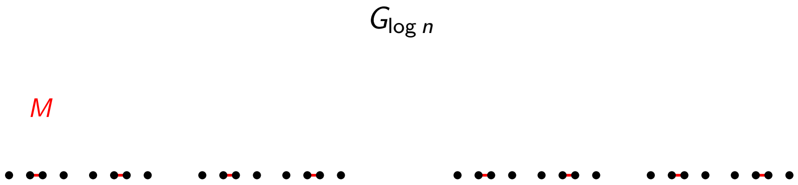
Graph Construction



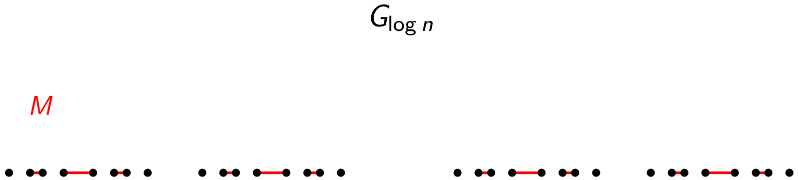
Matchings

$G_{\log n}$

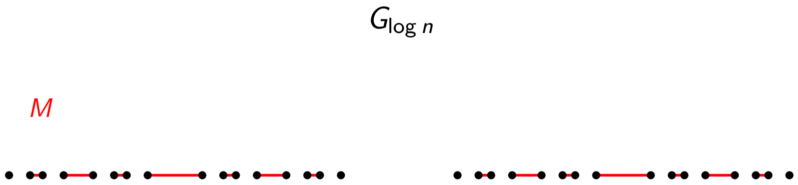
Matchings



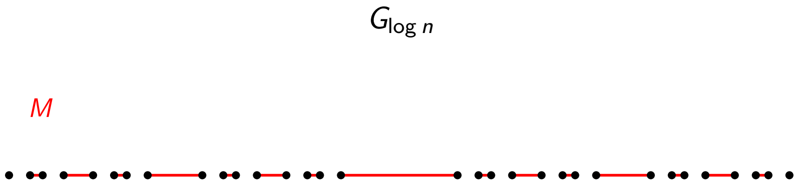
Matchings



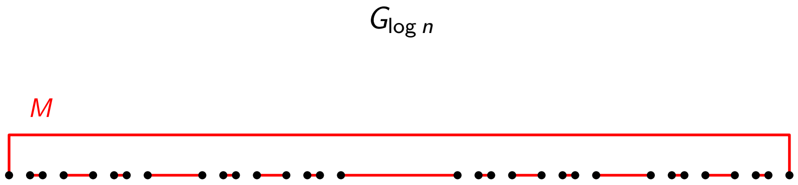
Matchings



Matchings



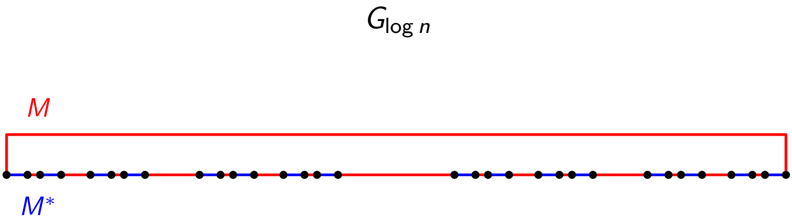
Matchings



Matchings

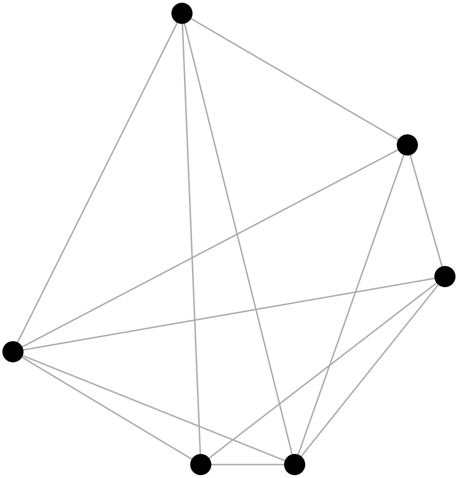
$G_{\log n}$

Matchings

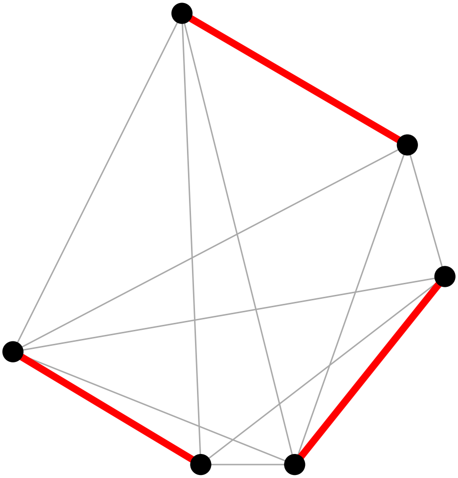


Finding Cheap Stable Matchings

Greedy Algorithm

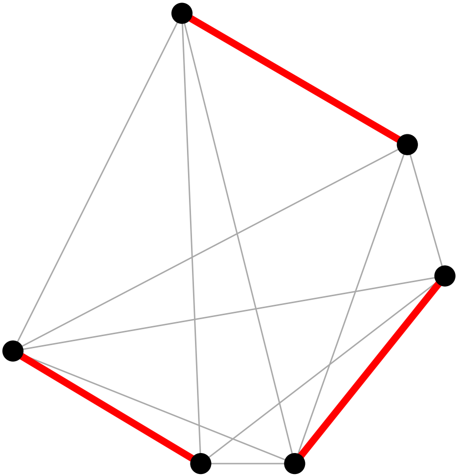


Greedy Algorithm



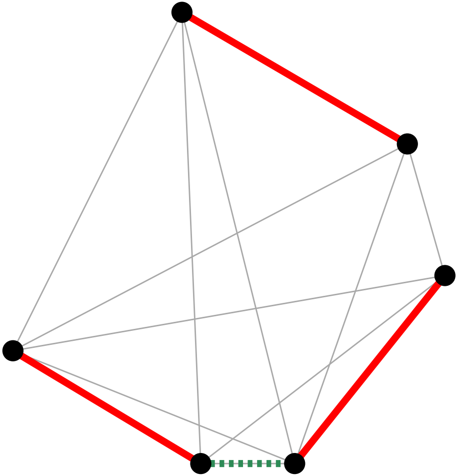
Start with a **minimum-cost matching**

Greedy Algorithm



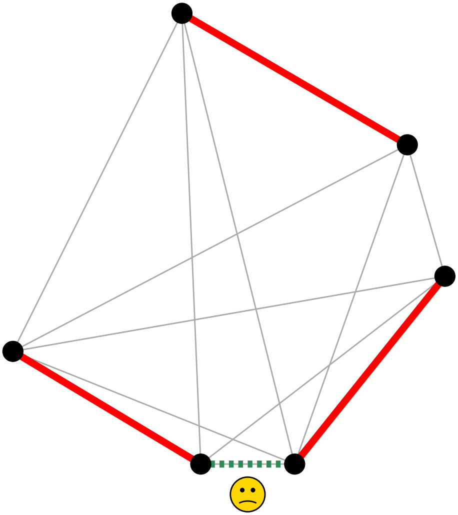
can be efficiently calculated by algorithm of **Lovasz & Plummer** (1986) based on **Edmonds**' work (1965)

Greedy Algorithm



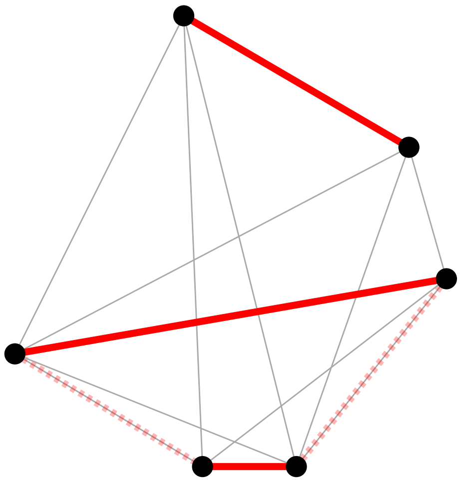
Consider edges $\notin M$ ordered by **ascending weights**

Greedy Algorithm



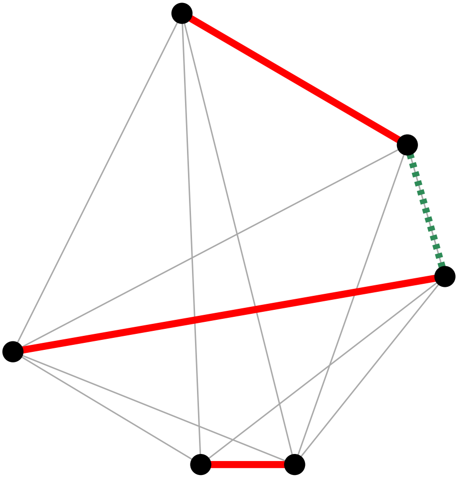
If edge is **unstable** . . .

Greedy Algorithm



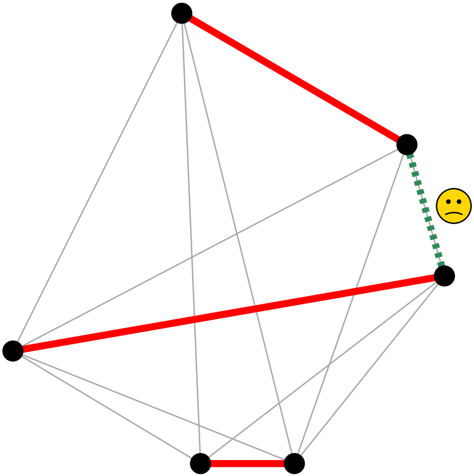
. . . **flip** it!

Greedy Algorithm



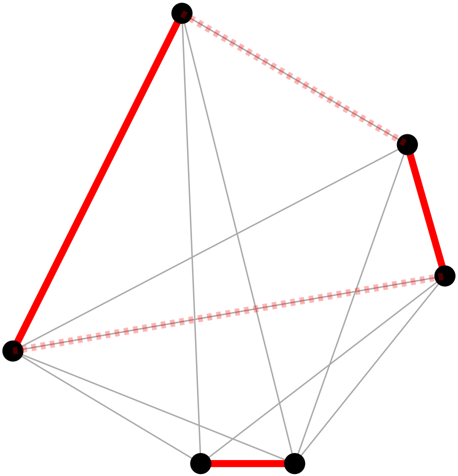
Consider next edge

Greedy Algorithm



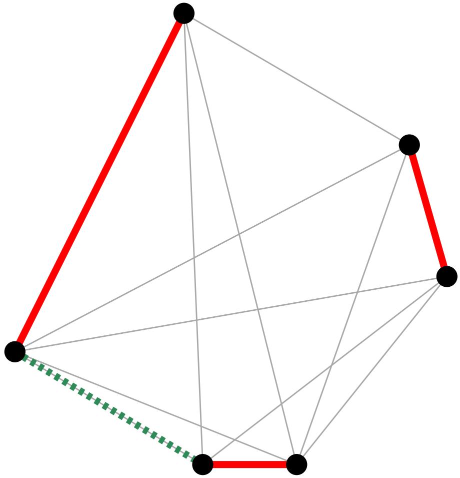
Edge is **unstable** . . .

Greedy Algorithm



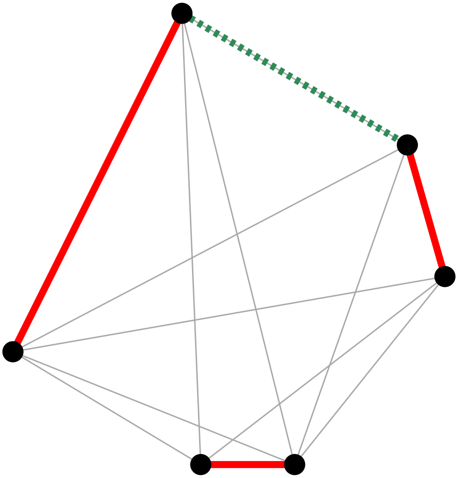
. . . **flip** again!

Greedy Algorithm



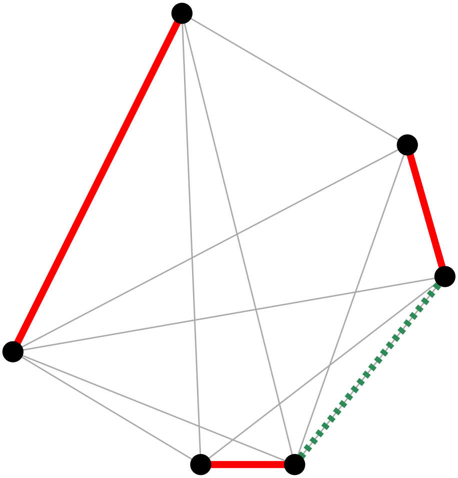
Repeat for remaining edges

Greedy Algorithm



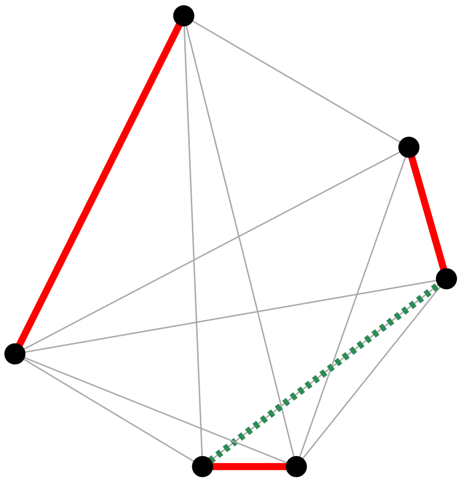
Repeat for remaining edges

Greedy Algorithm



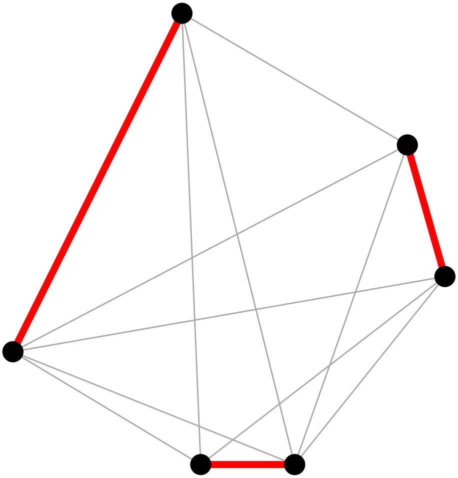
Repeat for remaining edges

Greedy Algorithm



Repeat for remaining edges

Greedy Algorithm



Return **stable** matching

Tight Trade-Off

Theorem (Upper Bound)

Let M_α be the matching returned by Greedy for some $\alpha \geq 1$. Then,

$$\frac{c(M_\alpha)}{c(M^*)} \in \mathcal{O}\left(n^{\log(1+1/(2\alpha))}\right) \ .$$

Theorem (Lower Bound)

For every $\alpha \geq 1$, there exists a metric graph such that for any α-stable matching M_α,

$$\frac{c(M_\alpha)}{c(M^*)} \in \Omega\left(n^{\log(1+1/(2\alpha))}\right) \ .$$

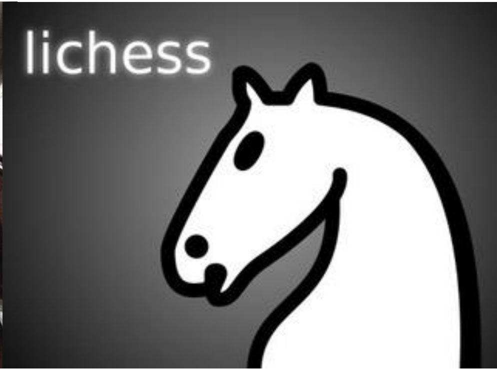

AND NOW FOR
SOMETHING
COMPLETELY
DIFFERENT

“Game Theory”

$100B Revenue

¾ Online

Online Two Player Games



Match Players Fast

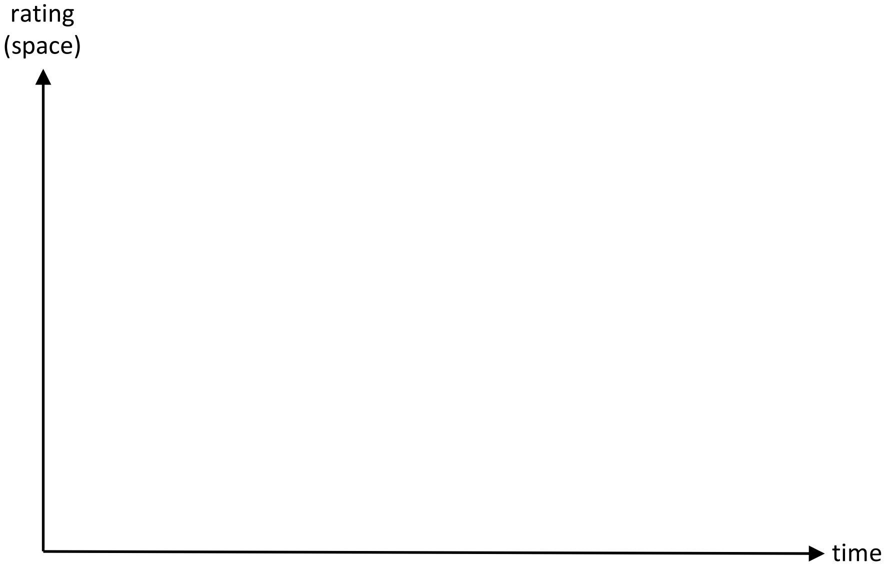
Waiting is Boooooring

Match Players Well

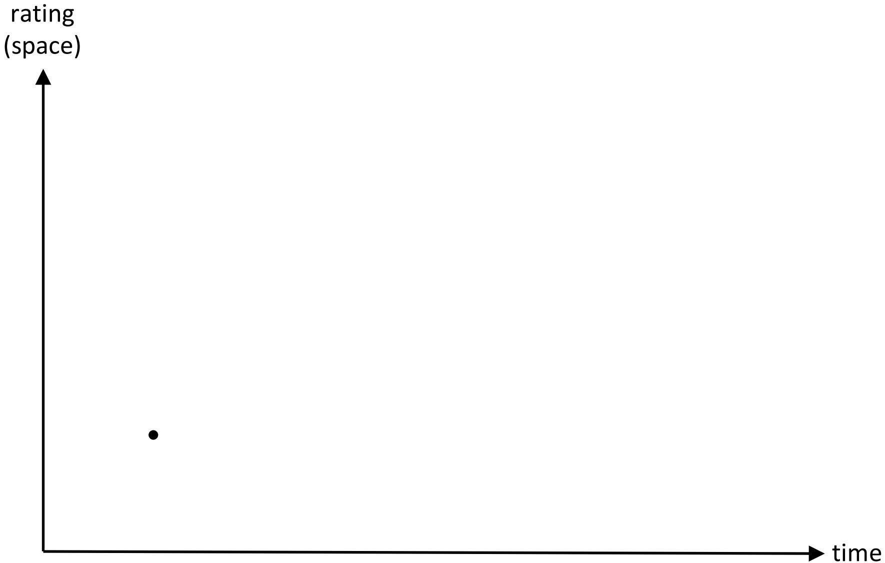
Similar Rating, Location, etc.

Min-Cost Perfect Matching With Delays (MPMD)

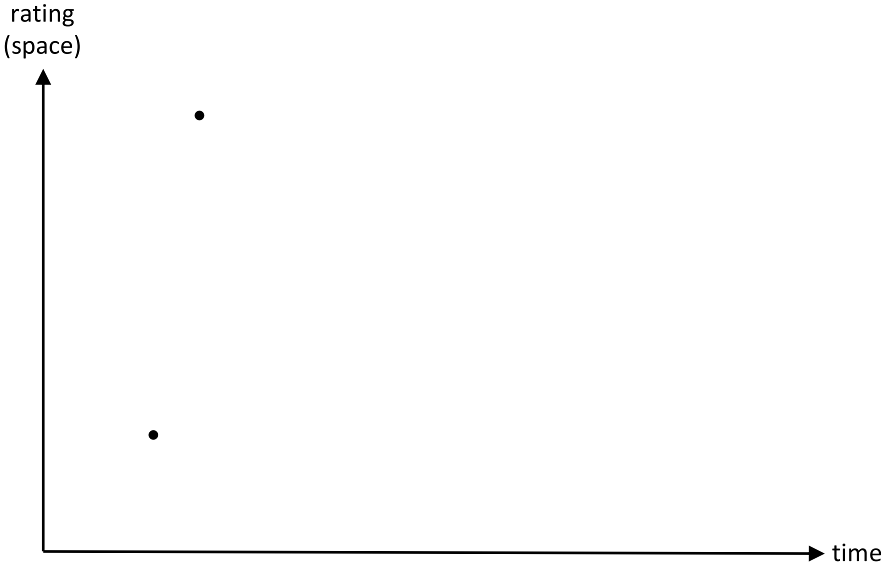
MPMD Example



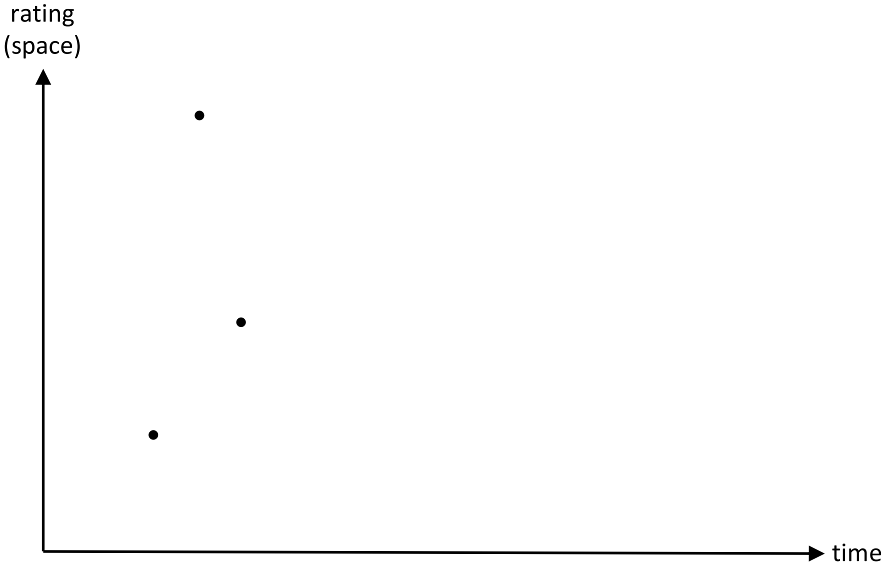
MPMD Example



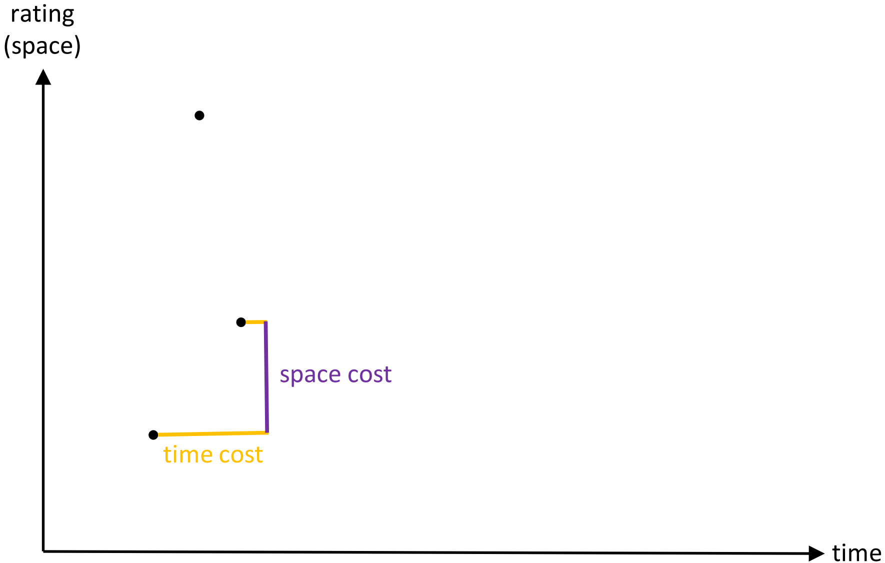
MPMD Example



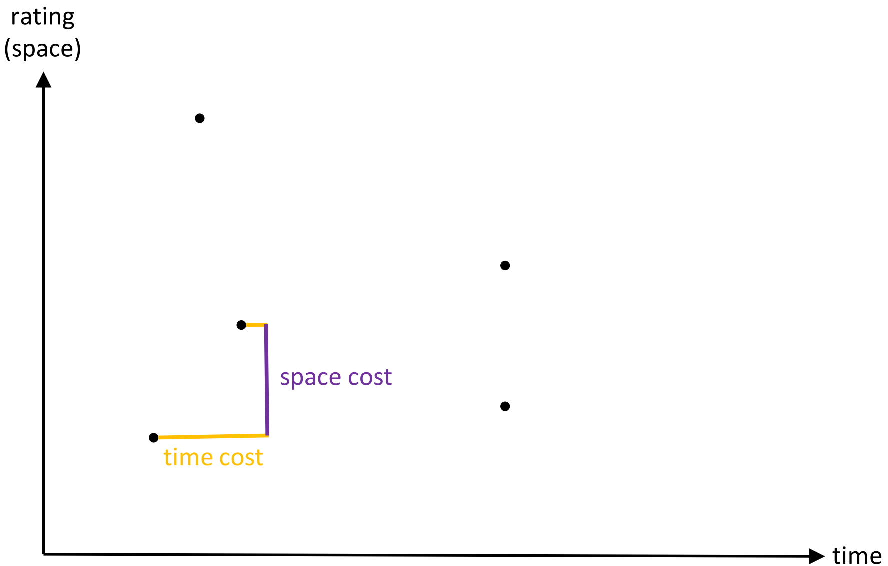
MPMD Example



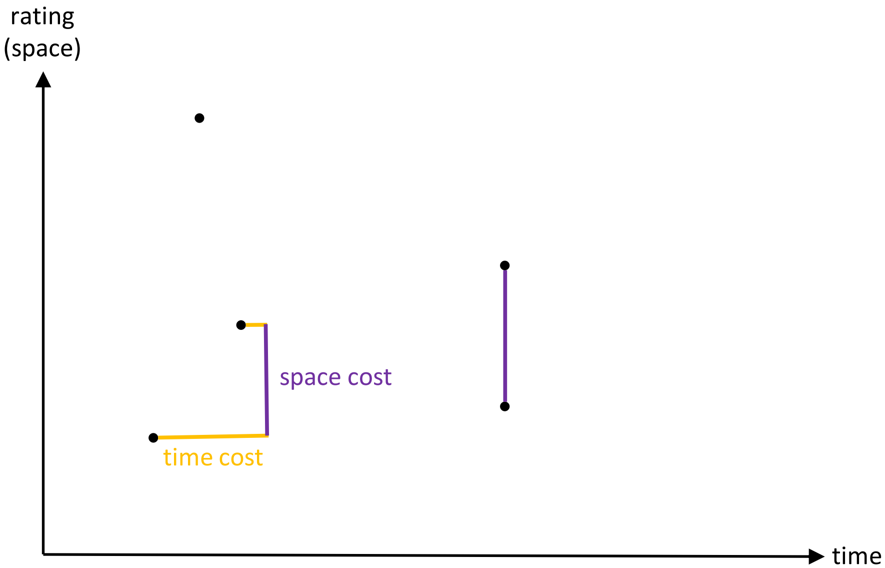
MPMD Example



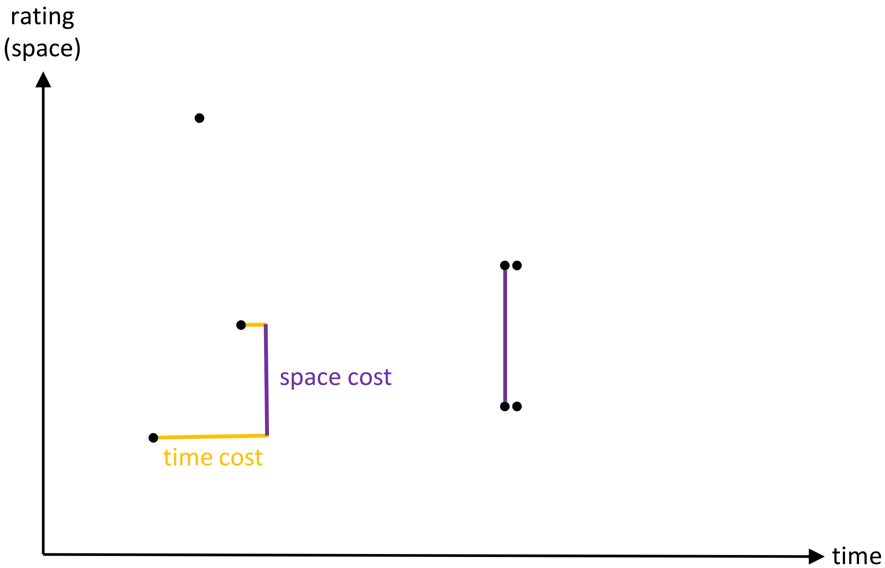
MPMD Example



MPMD Example

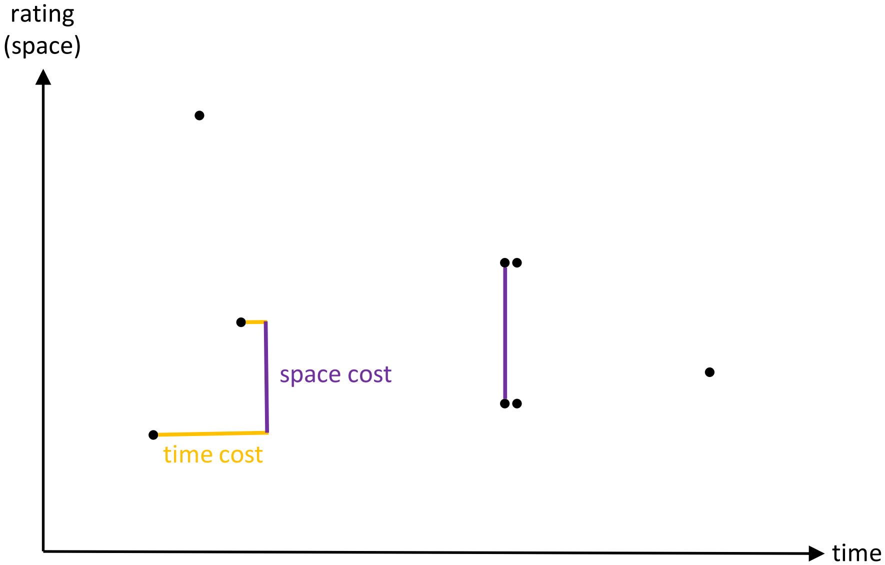


MPMD Example

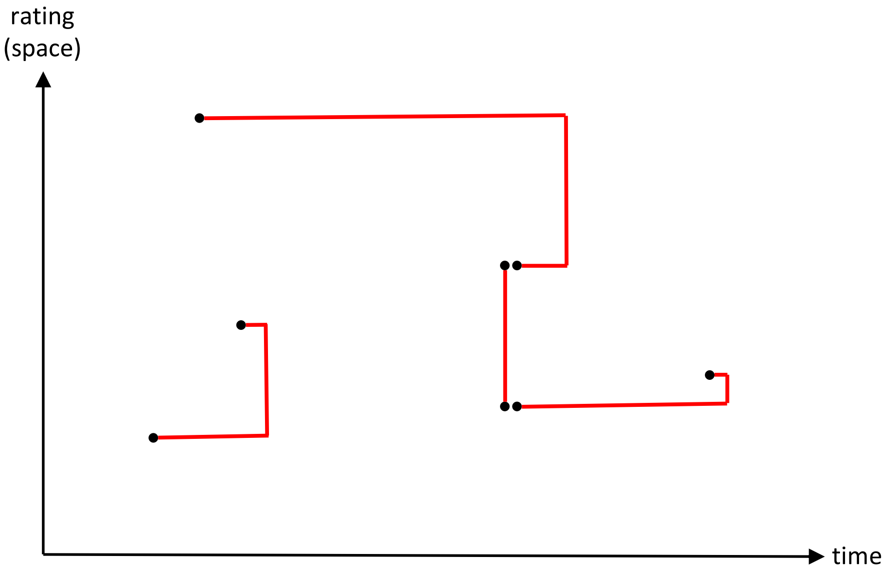


Haste Makes Waste!

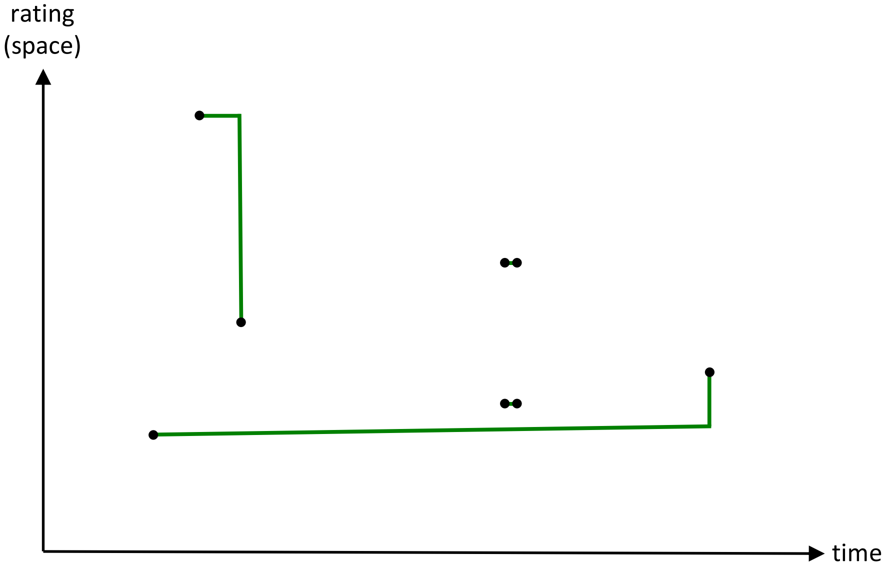
MPMD Example



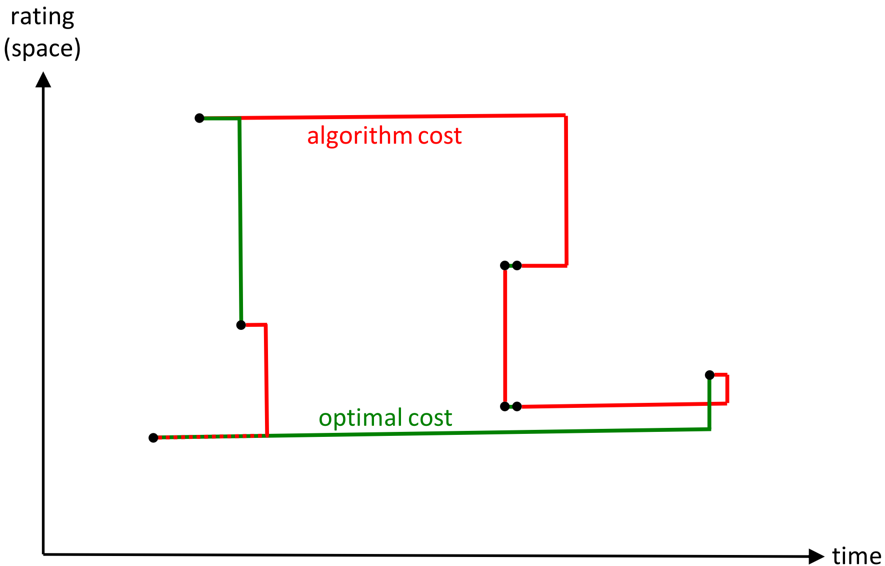
MPMD Example



MPMD Example



MPMD Example



Online Matching Literature

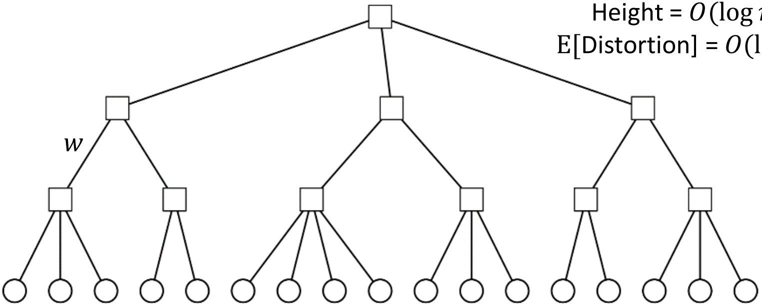
- Bipartite graph, left side is known, right side revealed online
  - Maximum cardinality matching
    [KVV1990, BM2008, GM2008, DJK2013, M2014, NW2015]
  - Maximum vertex weighted matching
    [AGKM2011, DJK2013, NW2015]
  - Maximum capacitated assignment (the AdWords problem)
    [MSVV2005, BJN2007, GM2008, AGKM2011, NW2015]
  - Metric maximum weight matching
    [KP1993, KMV1994]
  - Metric minimum cost perfect matching
    [KP1993, MNP2006, BBGN2014]
  - Metric minimum capacitated assignment (transportation)
    [KP2000]
- MPMD: known graph, both sides revealed online

MPD Results

- Finite metric space $\mathcal{M} = (V, \delta)$
  - $n = |V|$
  - $\Delta = \frac{\max_{x \neq y \in V} \delta(x,y)}{\min_{x \neq y \in V} \delta(x,y)}$
- $O(\log^2 n + \log \Delta)$-competitive randomized algorithm
  [Emek, Kutten, W 2016]
- $O(\log n)$-competitive (almost) deterministic algorithm
  Lower bound of $\Omega(\sqrt{\log n})$
  [Azar, Chiplunkar, Kaplan 2017]
- $O(\log n)$-competitive (almost) det. bipartite algorithm
  $\Omega(\sqrt{\log n / \log \log n})$ lower bound for bipartite
  $\Omega(\log n / \log \log n)$ lower bound for non-bipartite
  [Wang et al., in submission]

The $O(\log n)$ Algorithm

Approximate Metric by Tree

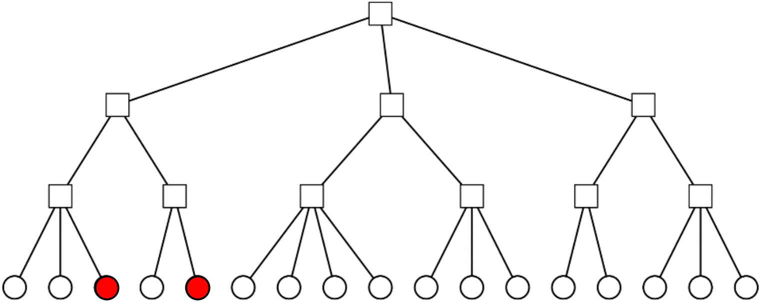


Height = $O(\log n)$
E[Distortion] = $O(\log n)$

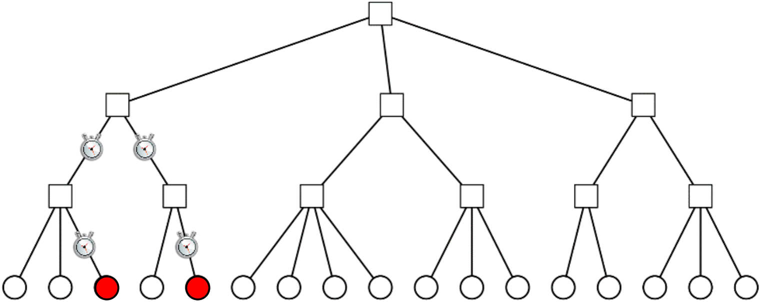
Leaves = Nodes in Metric Space

[Fakcharoenphol, Rao, Talwar 2004], [Bansal, Buchbinder, Gupta, Naor 2015]

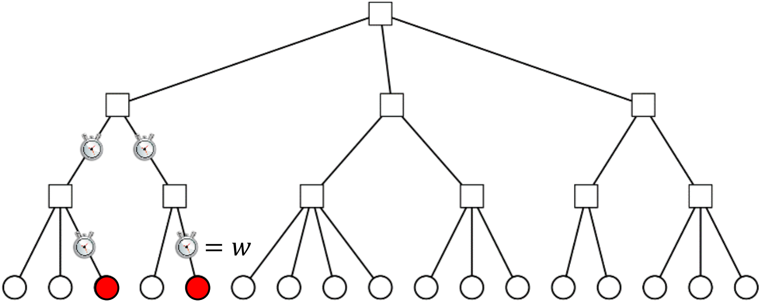
Algorithm



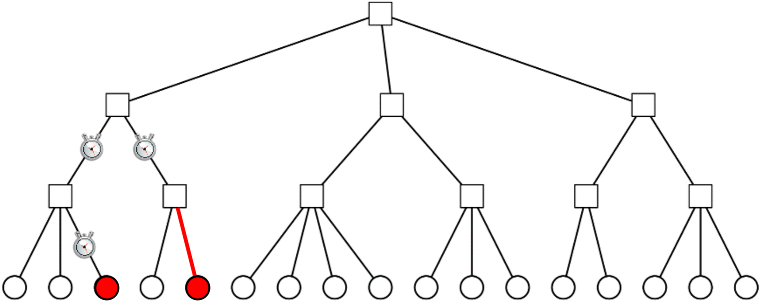
Algorithm



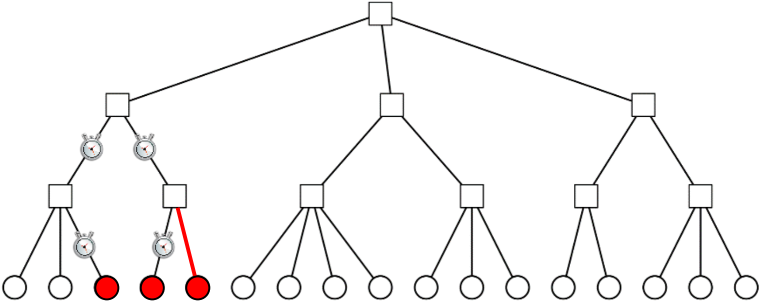
Algorithm



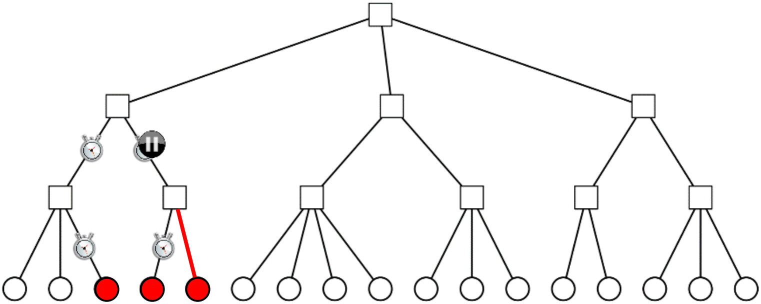
Algorithm



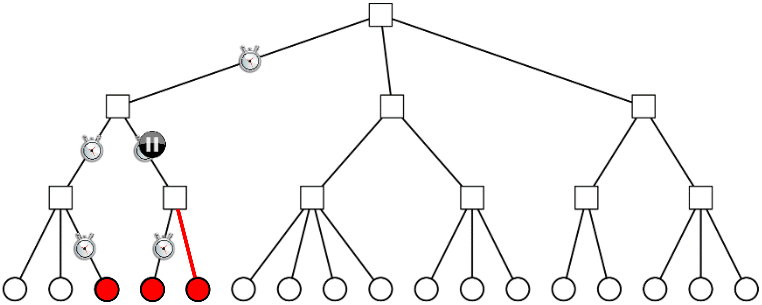
Algorithm



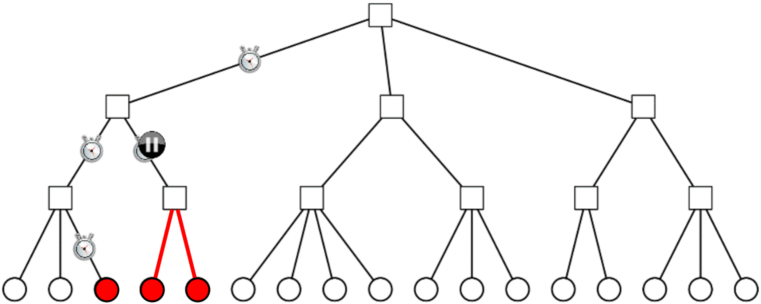
Algorithm



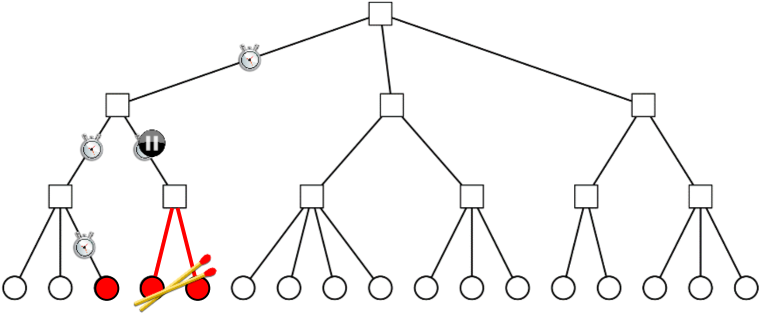
Algorithm



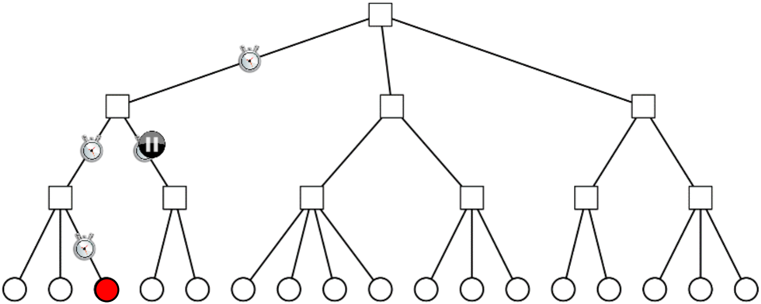
Algorithm



Algorithm

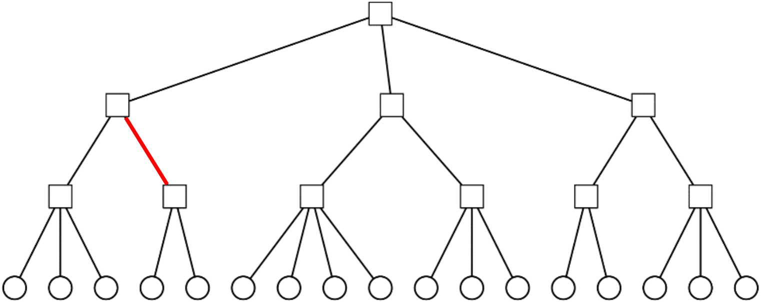


Algorithm

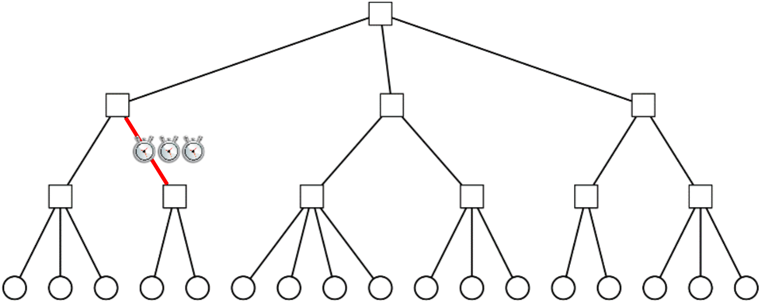


Proof

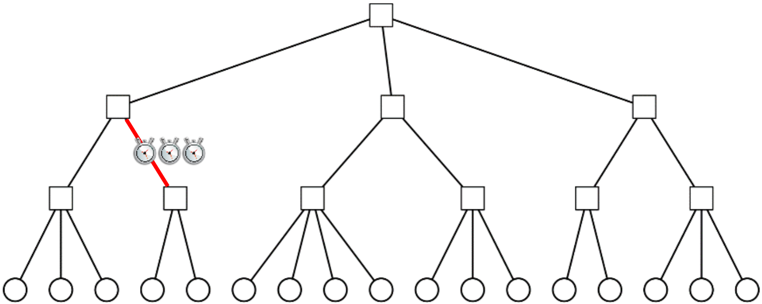
Proof



Proof

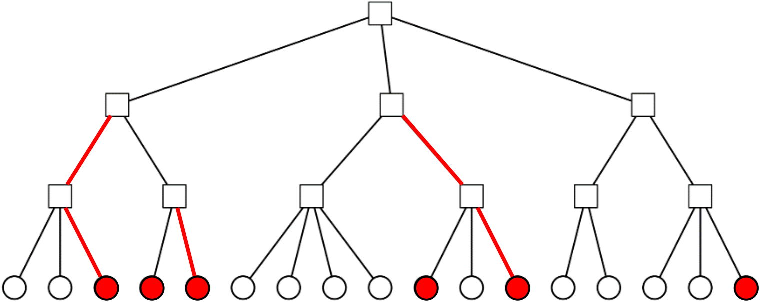


Proof

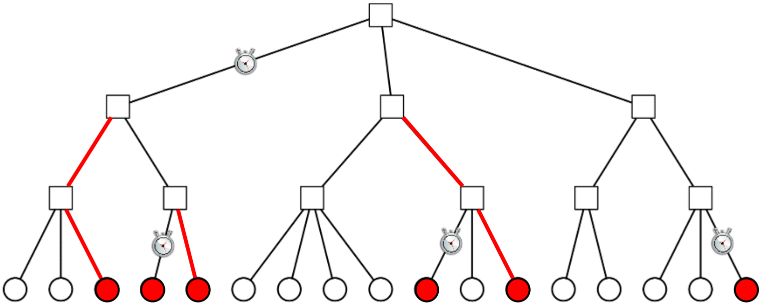


Total space cost = $\sum$ ⏱

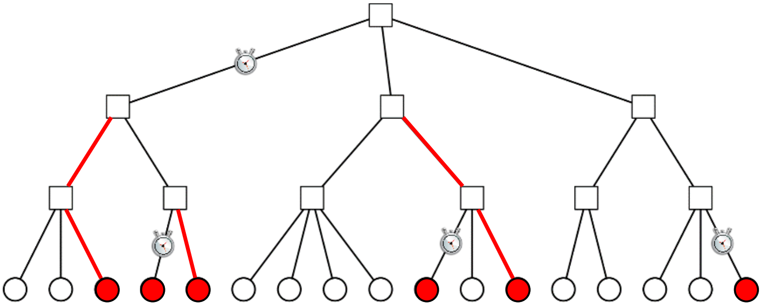
Proof



Proof



Proof



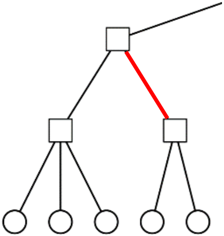
For each pair at least one timer running

Total time cost ≤ 2 ∑ ⏱

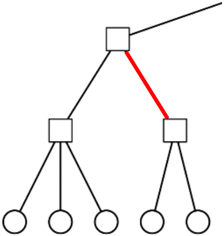
Total Algorithm Cost = $O(\sum$  $)$

What about OPT?

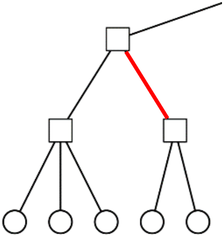
Proof

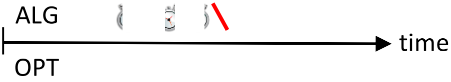


Proof

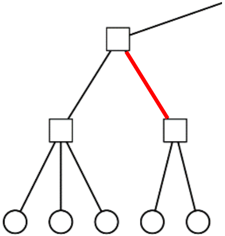


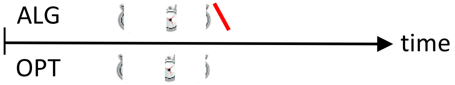
Proof



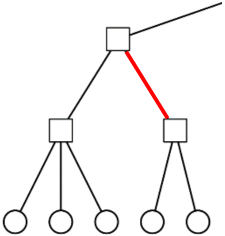


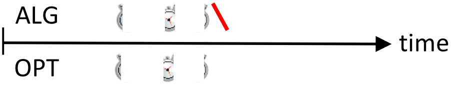
Proof



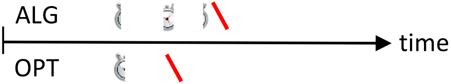


Proof

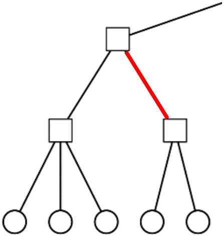


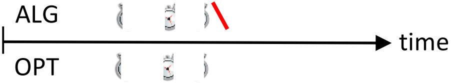


or

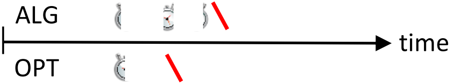


Proof





or

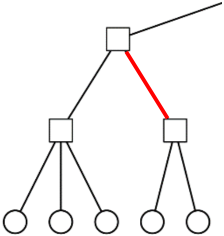


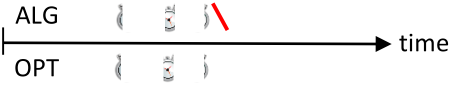
cost = cost

Done?

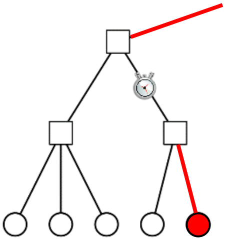
Just One Little Thing…

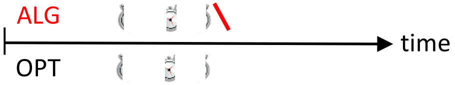
Proof



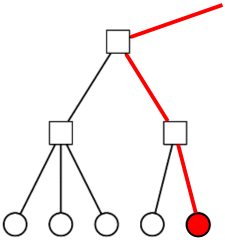


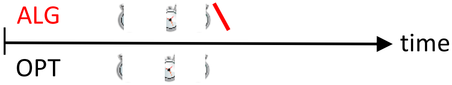
Proof



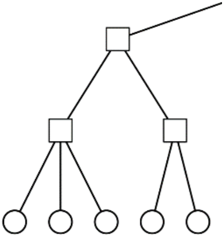


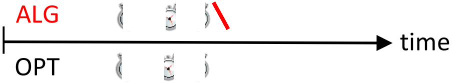
Proof



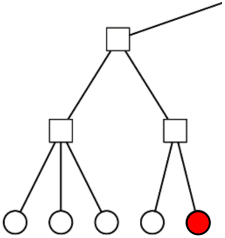


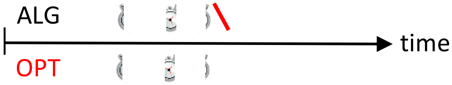
Proof



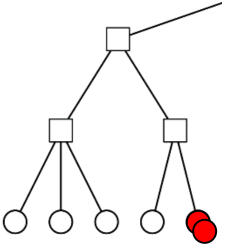


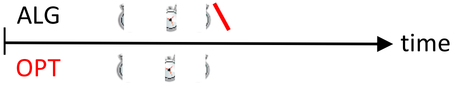
Proof



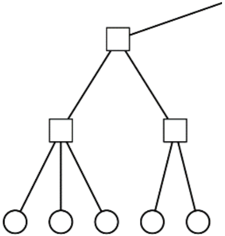


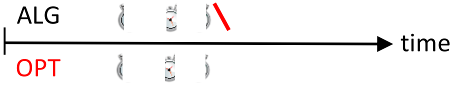
Proof



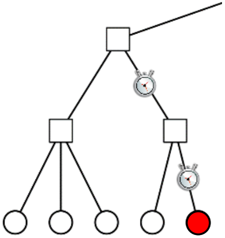


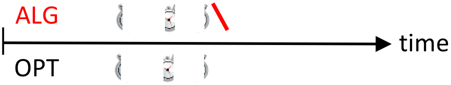
Proof



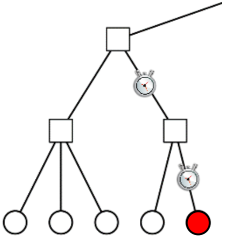


Proof





Proof



OPT has an easy time…

… but only every other phase!

Total OPT Cost = $\Omega(\sum$  $)$

Where is the $\log n$ coming from?

Height = $O(\log n)$ for time
E[Distortion] = $O(\log n)$ for space

Summary

Matching in Metric Spaces

Cheap or Stable

Good or Fast

Thank You!

Questions & Comments?

Thanks to my co-authors
ESA 2015: Yuval Emek, Tobias Langner
STOC 2016: Yuval Emek, Shay Kutten
In Submission: Yuyi Wang

Abstract: My talk is about matchings in a metric space. In the first part, we connect two classic approaches in matching, (i) a global optimization angle à la Edmonds, and (ii) a local selfish angle à la Gale and Shapley. We analyze the price of anarchy of metric matching when combining the two. The second part of the talk deals with an online version of metric matching. Consider an online gaming platform supporting two-player games such as Chess or Street Fighter 4. The platform tries to find a suitable opponent for each player, minimizing two criteria: (i) matching similar players, so that the game is challenging for both players; and (ii) the waiting time until a player is matched and can start playing since waiting is boring. It turns out that these two minimization criteria are often conflicting. To cope with this challenge, we must allow the platform to delay its service in a rent-or-buy manner.

The first part of my talk is based on an ESA 2015 paper with Yuval Emek and Tobias Langner. The second part is based on an STOC 2016 paper with Yuval Emek and Shay Kutten, and on unpublished work with Yuyi Wang and others.