Metric Matching

Cheap or Stable ... or Fast?

Roger Wattenhofer
Disclaimer
Matchings
Matchings
Matchings
Matchings
Matchings

3  9  4

1
Matchings

Cost: 7
Matchings

Cost: 10
Matchings

Cost: 7

Cost: 10
Matchings

Match 1: Jennifer Aniston - Angelina Jolie
Match 2: Brad Pitt - Angelina Jolie
Match 3: Jennifer Aniston - Brad Pitt
Match 4: Angelina Jolie - Brad Pitt

Scores:
- Jennifer Aniston - Brad Pitt: 9
- Jennifer Aniston - Angelina Jolie: 3
- Brad Pitt - Angelina Jolie: 4
- Angelina Jolie - Brad Pitt: 1
Matchings

1

3

9

4
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 $M^* \subseteq V \times V$

$$c(M^*) = 3 + 4 = 7$$
$\alpha$-unstable edge $e \not\in M$

\[ w(e) < \frac{1}{\alpha} \cdot \min\{w(e_1), w(e_2)\} \]
Example: \textbf{2-unstable edge} $e \notin M$

$$w(e) < \frac{1}{2} \cdot \min\{w(e_1), w(e_2)\}$$
$\alpha$-stable matching: without $\alpha$-unstable edge
Stable vs. Cheap
Stable vs. Cheap

min-cost matching $M^*$

$$c(M^*) = 2$$
α-stable matching $M$

$c(M) = \infty$
Metric Graphs

Points in metric

\[ v_1 \sim x_1 \]

\[ d(v_1, v_4) \]

\[ d(v_1, v_2) \]

Metric graph \( G \)

\[ w((x_i, x_j)) = d(v_i, v_j) \]

\[ \Rightarrow \]
Stable Matchings Can Be Expensive
Graph Construction

$G_2$
Graph Construction

$G_2$

1/α − ε

1

1

…”
Graph Construction

\[ G_2 \]

\[ 1 + \frac{1}{\alpha - \varepsilon} \]

\[ 2 + 1 + \frac{1}{\alpha - \varepsilon} \]
Graph Construction

\[ \| G_2 \| = 2 + \frac{1}{\alpha} - \varepsilon \]
Graph Construction

\[ \| G_2 \| = 2 + \frac{1}{\alpha - \varepsilon} \]

\[ (1/\alpha - \varepsilon) \cdot \| G_2 \| \]

G_3
Graph Construction

\[ \| G_2 \| = 2 + \frac{1}{\alpha - \varepsilon} \]

\[ \| G_3 \| = (2 + \frac{1}{\alpha - \varepsilon})^2 \]
Graph Construction

$G_i$

$\|G_{i-1}\|$
Graph Construction

\[ G_i \]

\[ \frac{1}{\alpha} - \varepsilon \cdot \| G_{i-1} \| \]
Graph Construction

\[ G_i \]

\[
(1/\alpha - \varepsilon) \cdot \|G_{i-1}\|
\]

\[
\|G_i\| = (2 + 1/\alpha - \varepsilon)^{i-1}
\]
$G_{\log n}$
\[ M \]

\[ G_{\log n} \]
Matchings

\[ G_{\log n} \]

\[ M \]

\[ c(M) \]

\[ n^{1/2} \]

\[ (2 + \frac{1}{\epsilon'}) \log n \]

\[ 2^{1 + \frac{1}{\epsilon'}} \log(2 + \frac{1}{\epsilon'}) \]

\[ c(M) \]

\[ n^{1/2} \]

\[ (2 + \frac{1}{\epsilon'}) \log n \]

\[ 2^{1 + \frac{1}{\epsilon'}} \log(1 + \frac{1}{(2 - \epsilon')}) \]
Matchings

\[ G_{\log n} \]

\[ M \]
Matchings

$G_{\log n}$

$M$

$\log n$
$G_{\log n}$

$M^*$
Matchings

$G_{\log n}$

$M$

$M^*$

$\log n$
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
Theorem (Upper Bound)

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

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

Theorem (Lower Bound)

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

$$\frac{c(M_\alpha)}{c(M^*)} \in \Omega(n^{\log(1+1/(2\alpha))}) .$$
AND NOW FOR SOMETHING COMPLETELY DIFFERENT
“Game Theory”
$100B Revenue
¾ Online
Online Two Player Games

Match Players Fast
Waiting is Booooorrring

Match Players Well
Similar Rating, Location, etc.
Min-Cost Perfect Matching With Delays (MPMD)
MPMD Example

rating (space)

\[
\text{rating (space)} \rightarrow \text{time}
\]
MPMD Example

rating (space)

\[ \text{(coordinate point)} \]

\[ (t, r) \]
MPMD Example

rating (space) vs time
MPMD Example

rating (space)

time

space cost
time cost
MPMD Example

rating (space)

<table>
<thead>
<tr>
<th>time cost</th>
<th>space cost</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>time</th>
<th>rating</th>
</tr>
</thead>
</table>


Haste Makes Waste!
MPMD Example

rating (space)

time
MPMD Example

rating (space)

algorithm cost

optimal cost

time
Online Matching Literature

- Bipartite graph, left side is known, right side revealed online
  - Maximum cardinality matching
  - Maximum vertex weighted matching
    [AGKM2011, DJK2013, NW2015]
  - Maximum capacitated assignment (the AdWords problem)
  - 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
**MPMD Results**

- Finite metric space $\mathcal{M} = (\mathcal{V}, \delta)$
  - $n = |\mathcal{V}|$
  - $\Delta = \frac{\max_{x \neq y \in \mathcal{V}} \delta(x, y)}{\min_{x \neq y \in \mathcal{V}} \delta(x, y)}$

- $O(\log^2 n + \log \Delta)$-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

Leaves = Nodes in Metric Space

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

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

= \( w \)
Algorithm
Proof
Proof
Proof

Total space cost = \( \sum \)
Proof
Proof
Proof

For each pair at least one timer running

Total time cost $\leq 2 \sum \text{Timer}$
Total Algorithm Cost = $O\left(\sum \text{\text{clock}}\right)$
What about OPT?
Proof
Proof
Proof
Proof

ALG \rightarrow \text{time} \rightarrow \text{OPT}
Proof

ALG \quad \text{(or)} \quad \text{OPT}

\text{time}

or

ALG \quad \text{(or)} \quad \text{OPT}

\text{time}
Proof

cost = \text{cost}
Done?
Just One Little Thing...
Proof
Proof
Proof

[Diagram showing a tree structure with labeled nodes ALG and OPT, and a timeline marked with time]
Proof
Proof
Proof
Proof
Proof
Proof
OPT has an easy time...
... but only every other phase!
Total OPT Cost = $\Omega\left(\sum \right)$
Where is the $\log n$ coming from?

Height = $\mathcal{O}(\log n)$ for time
$\mathbb{E}[\text{Distortion}] = \mathcal{O}(\log n)$ 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

www.disco.ethz.ch
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.