# Online Matching from an Economics Viewpoint

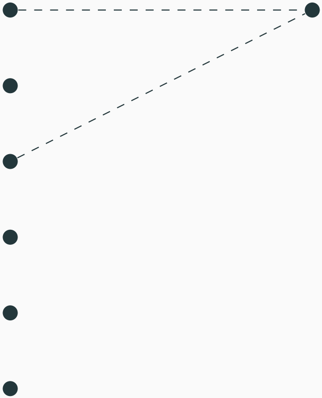
Thorben Tröbst

Theory Seminar, October 23, 2020

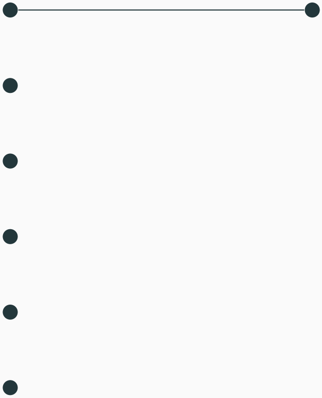
Department of Computer Science, University of California, Irvine

# Online Bipartite Matching

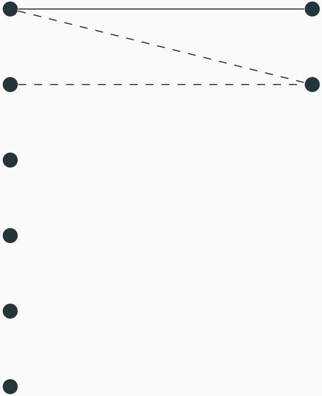
# Online Bipartite Matching



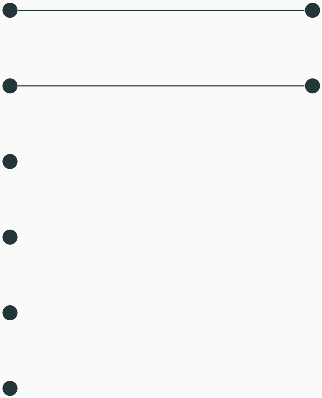
# Online Bipartite Matching



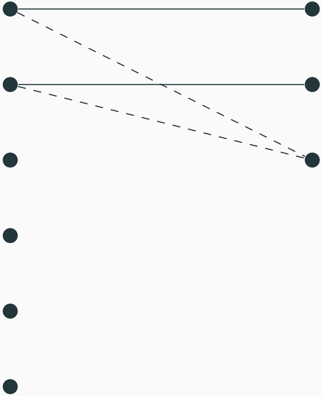
# Online Bipartite Matching



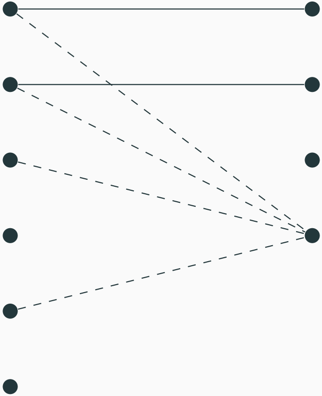
# Online Bipartite Matching



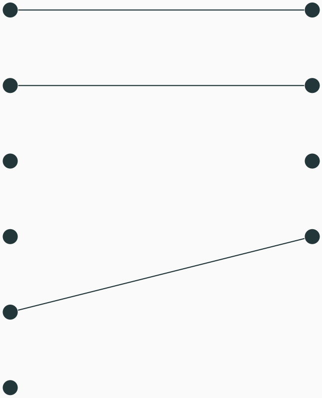
# Online Bipartite Matching



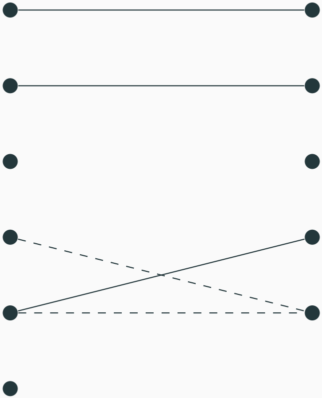
# Online Bipartite Matching



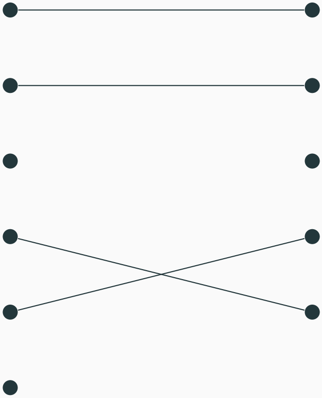
# Online Bipartite Matching



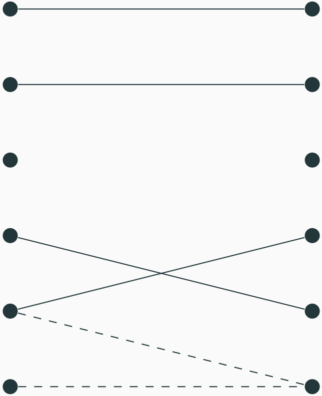
# Online Bipartite Matching



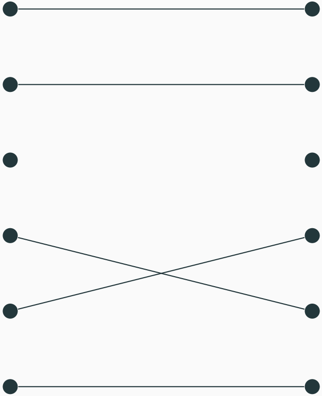
# Online Bipartite Matching



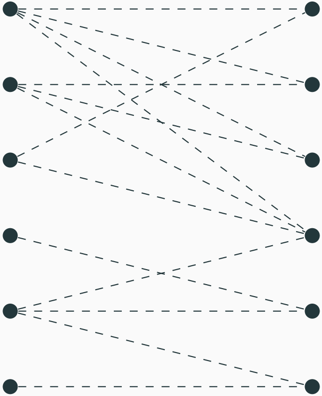
# Online Bipartite Matching



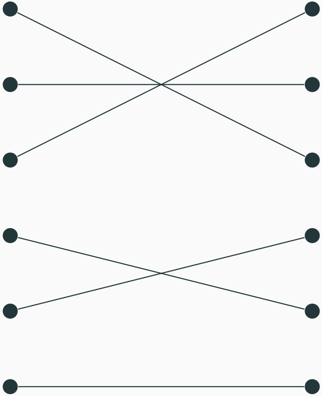
# Online Bipartite Matching



# Online Bipartite Matching



# Online Bipartite Matching



# Online Bipartite Matching II

$G = (S, B, E)$ is a bipartite graph consisting of offline vertices $S$ and online vertices $B$.

# Online Bipartite Matching II

$G = (S, B, E)$ is a bipartite graph consisting of offline vertices $S$ and online vertices $B$.

Online vertices arrive one by one in adverserial order.

# Online Bipartite Matching II

$G = (S, B, E)$ is a bipartite graph consisting of offline vertices $S$ and online vertices $B$.

Online vertices arrive one by one in adverserial order.

The algorithm must irrevocably and immediately match revealed online vertices.

# Online Bipartite Matching II

$G = (S, B, E)$ is a bipartite graph consisting of offline vertices $S$ and online vertices $B$.

Online vertices arrive one by one in adverserial order.

The algorithm must irrevocably and immediately match revealed online vertices.

The goal is to maximize the competitive ratio, i.e.

$$\frac{|M_{\text{online}}|}{\text{OPT}_{\text{offline}}}.$$

There are two main algorithmic ideas for online matching problems:

There are two main algorithmic ideas for online matching problems:

- RANKING

# Algorithms for Online Matching Problems

There are two main algorithmic ideas for online matching problems:

- RANKING
  - Idea: randomly permute offline vertices and then match online vertices to the first available offline vertex.

# Algorithms for Online Matching Problems

There are two main algorithmic ideas for online matching problems:

- Ranking
    - Idea: randomly permute offline vertices and then match online vertices to the first available offline vertex.
    - Provides integral solution in a randomized algorithm.

# Algorithms for Online Matching Problems

There are two main algorithmic ideas for online matching problems:

- Ranking
    - Idea: randomly permute offline vertices and then match online vertices to the first available offline vertex.
    - Provides integral solution in a randomized algorithm.
- Water-Filling / Balancing

# Algorithms for Online Matching Problems

There are two main algorithmic ideas for online matching problems:

- Ranking
    - Idea: randomly permute offline vertices and then match online vertices to the first available offline vertex.
    - Provides integral solution in a randomized algorithm.
- Water-Filling / Balancing
    - Idea: continuously allocate online vertices to the least-matched offline vertices.

# Algorithms for Online Matching Problems

There are two main algorithmic ideas for online matching problems:

- Ranking
    - Idea: randomly permute offline vertices and then match online vertices to the first available offline vertex.
    - Provides integral solution in a randomized algorithm.
- Water-Filling / Balancing
    - Idea: continuously allocate online vertices to the least-matched offline vertices.
    - Provides fractional solution in a deterministic algorithm.

# History of the Economic Viewpoint

- Karp, Vazirani, Vazirani 1990: Ranking algorithm is $(1 - 1/e)$-competitive for the Online Bipartite Matching Problem

# History of the Economic Viewpoint

- Karp, Vazirani, Vazirani 1990: Ranking algorithm is $(1 - 1/e)$-competitive for the Online Bipartite Matching Problem
- Goel and Mehta 2008: first significant simplication of the Ranking analysis

# History of the Economic Viewpoint

- Karp, Vazirani, Vazirani 1990: Ranking algorithm is $(1 - 1/e)$-competitive for the Online Bipartite Matching Problem
- Goel and Mehta 2008: first significant simplication of the Ranking analysis
- Devanur, Jain, Kleinberg 2013: primal-dual analysis further simplifies Ranking analysis and allows for easy extension to vertex weights

# History of the Economic Viewpoint

- Karp, Vazirani, Vazirani 1990: RANKING algorithm is $(1 - 1/e)$-competitive for the Online Bipartite Matching Problem
- Goel and Mehta 2008: first significant simplication of the RANKING analysis
- Devanur, Jain, Kleinberg 2013: primal-dual analysis further simplifies RANKING analysis and allows for easy extension to vertex weights
- **Eden, Feldman, Fiat, Segal 2018:** economics viewpoint yields the most elegant analysis yet

# History of the Economic Viewpoint

- Karp, Vazirani, Vazirani 1990: RANKING algorithm is $(1 - 1/e)$-competitive for the Online Bipartite Matching Problem
- Goel and Mehta 2008: first significant simplication of the RANKING analysis
- Devanur, Jain, Kleinberg 2013: primal-dual analysis further simplifies RANKING analysis and allows for easy extension to vertex weights
- **Eden, Feldman, Fiat, Segal 2018:** economics viewpoint yields the most elegant analysis yet

$\Rightarrow$ Ideas can be extended to WATER-FILLING and many interesting settings!

# Ranking through the Economic Viewpoint

# The Ranking Algorithm

The classic Ranking algorithm proceeds as follows:

# The Ranking Algorithm

The classic Ranking algorithm proceeds as follows:

- Pick a uniformly random bijection $\tau : S \to \{1, \ldots, |S|\}$.

# The Ranking Algorithm

The classic Ranking algorithm proceeds as follows:

- Pick a uniformly random bijection $\tau : S \to \{1, \ldots, |S|\}$.
- Whenever a vertex $i \in B$ arrives, let $N(i)$ be the unmatched neighbors. Match $i$ to $j$ minimizing $\tau(j)$.

# The Economic Viewpoint

In the economic viewpoint, the graph $G$ represents a market.

## The Economic Viewpoint

In the economic viewpoint, the graph $G$ represents a market.

Online vertices $B$ are buyers whereas offline vertices $S$ are sellers.

# The Economic Viewpoint

In the economic viewpoint, the graph $G$ represents a market.

Online vertices $B$ are buyers whereas offline vertices $S$ are sellers.

Sellers randomly pick a fixed price for their item according to some distribution $\mathcal{D}$ on $[0, 1]$.

# The Economic Viewpoint

In the economic viewpoint, the graph $G$ represents a market.

Online vertices $B$ are buyers whereas offline vertices $S$ are sellers.

Sellers randomly pick a fixed price for their item according to some distribution $\mathcal{D}$ on $[0, 1]$.

Buyers seek to maximize their utility, i.e. $1 - p_j$ where $j$ is the item they buy.

# The Economic Viewpoint

In the economic viewpoint, the graph $G$ represents a market.

Online vertices $B$ are buyers whereas offline vertices $S$ are sellers.

Sellers randomly pick a fixed price for their item according to some distribution $\mathcal{D}$ on $[0, 1]$.

Buyers seek to maximize their utility, i.e. $1 - p_j$ where $j$ is the item they buy.

$\Rightarrow$ The resulting algorithm is identical to RANKING no matter what $\mathcal{D}$ is!

# Economic Ranking Example

- 
- 
- 
- 
- 
- 

# Economic Ranking Example

0.6

0.5

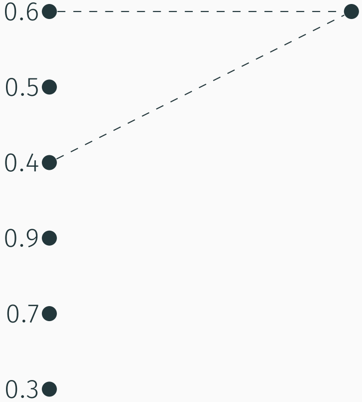
0.4

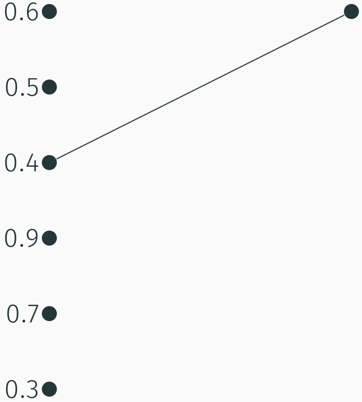
0.9

0.7

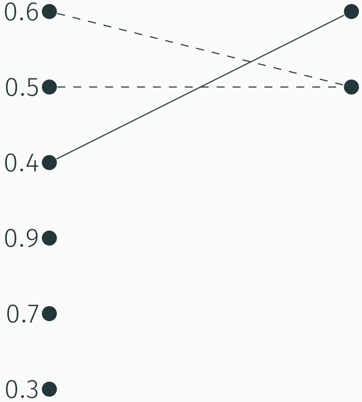
0.3

# Economic Ranking Example

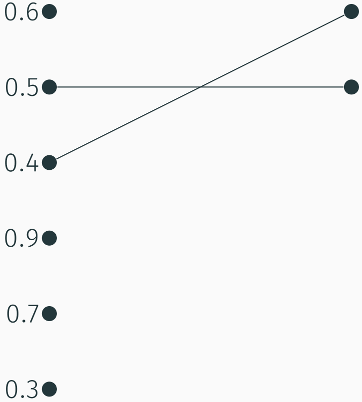



0.6
0.5
0.4
0.9
0.7
0.3

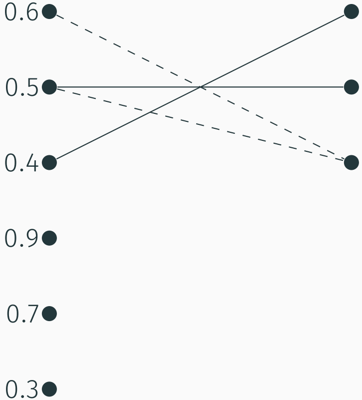
# Economic Ranking Example



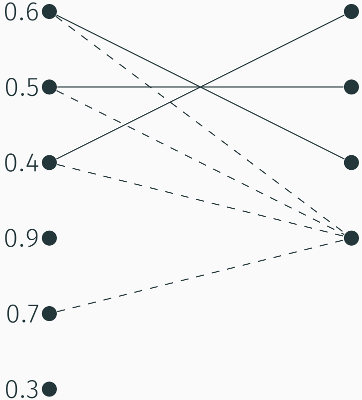
# Economic RANKING Example



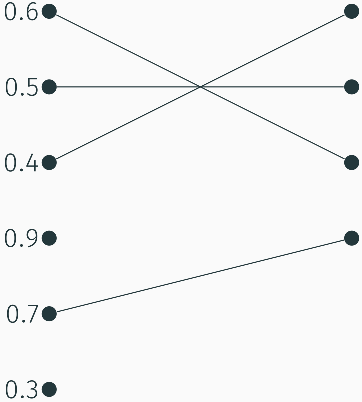
# Economic RANKING Example



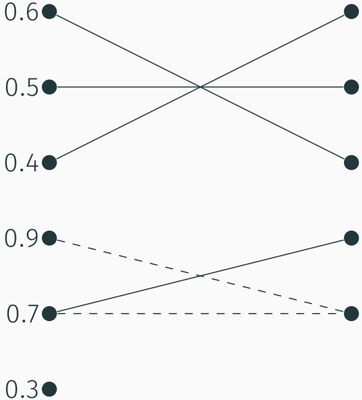
# Economic RANKING Example


0.6
0.5
0.4
0.9
0.7
0.3

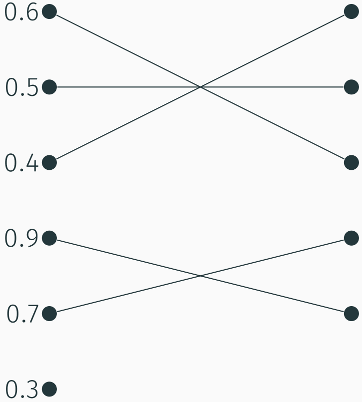
# Economic RANKING Example



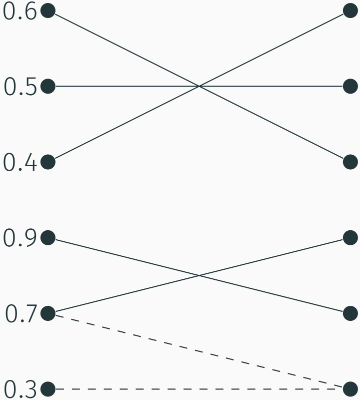
# Economic Ranking Example



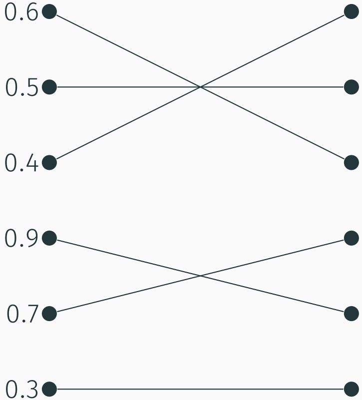
# Economic RANKING Example



# Economic RANKING Example



# Economic RANKING Example



# Revenue and Utility

When a sale is made at price $p$, the seller earns a revenue $r_j$ of $p$ and the buyer a utility $u_i$ of $1 - p$.

# Revenue and Utility

When a sale is made at price $p$, the seller earns a revenue $r_j$ of $p$ and the buyer a utility $u_i$ of $1 - p$.

The social welfare is defined as

$$\mathrm{SW} := \sum_{j \in S} r_j + \sum_{i \in B} u_i.$$

# Revenue and Utility

When a sale is made at price $p$, the seller earns a revenue $r_j$ of $p$ and the buyer a utility $u_i$ of $1 - p$.

The social welfare is defined as

$$\mathrm{SW} := \sum_{j \in S} r_j + \sum_{i \in B} u_i.$$

$\Rightarrow$ $\mathrm{SW}$ is the size of the matching created by the algorithm!

# Analysis of the Algorithm

**Lemma**

*Let $(j, i) \in E$ be arbitrary and fix all the prices except for $p_j$. Let $u^*$ be the utility of buyer $i$ if seller $j$ were removed. Then:*

1. *No matter what $p_j$ is, $u_i \geq u^*$.*
2. *If $1 - p_j > u^*$, then $j$ will be sold.*

**Lemma**

*Let $(j, i) \in E$ be arbitrary and fix all the prices except for $p_j$. Let $u^*$ be the utility of buyer $i$ if seller $j$ were removed. Then:*

1. *No matter what $p_j$ is, $u_i \geq u^*$.*
2. *If $1 - p_j > u^*$, then $j$ will be sold.*

1. If new sellers enter the market, the utilities of all buyers can only increase. This is called monotonicity of the market.

# Analysis of the Algorithm

**Lemma**

*Let $(j, i) \in E$ be arbitrary and fix all the prices except for $p_j$. Let $u^*$ be the utility of buyer $i$ if seller $j$ were removed. Then:*

1. *No matter what $p_j$ is, $u_i \geq u^*$.*
2. *If $1 - p_j > u^*$, then $j$ will be sold.*

1. If new sellers enter the market, the utilities of all buyers can only increase. This is called monotonicity of the market.

2. If $j$ was not matched, then $i$ would buy $j$ when they arrive to the market. $\square$

# Analysis of the Algorithm II

Assume that prices are chosen such that $p_j = e^{y_j - 1}$ where $y_j \in [0, 1]$ is uniformly distributed.

**Lemma**

*Let $(j, i) \in E$ be arbitrary. Then*

$$\mathbb{E}[r_j + u_i] \geq 1 - \frac{1}{e}.$$

# Analysis of the Algorithm III

**Proof.** Fix all prices except for $p_j$. By the previous lemma, $u_i \geq u^*$.

# Analysis of the Algorithm III

**Proof.** Fix all prices except for $p_j$. By the previous lemma, $u_i \geq u^*$.

We know that $j$ will be sold if $1 - p_j > u^*$. Let $y^* := 1 + \ln(1 - u^*)$, then

$$\mathbb{E}[r_j] \geq \int_0^{y^*} e^{y_j - 1} \, \mathrm{d}y_j$$
$$= 1 - u^* - \frac{1}{e}.$$

# Analysis of the Algorithm III

**Proof.** Fix all prices except for $p_j$. By the previous lemma, $u_i \geq u^*$.

We know that $j$ will be sold if $1 - p_j > u^*$. Let $y^* := 1 + \ln(1 - u^*)$, then

$$\begin{aligned} \mathbb{E}[r_j] &\geq \int_0^{y^*} e^{y_j - 1} \, \mathrm{d}y_j \\ &= 1 - u^* - \frac{1}{e}. \end{aligned}$$

Thus $\mathbb{E}[r_j + u_i] \geq 1 - \frac{1}{e}$. $\square$

# Analysis of the Algorithm IV

**Theorem**

*RANKING is* $(1 - 1/e)$*-competitive.*

**Theorem**

*Ranking is $(1 - 1/e)$-competitive.*

**Proof.** Let $M$ be a maximum matching. Then

$$\mathbb{E}[\mathrm{SW}] = \mathbb{E}\left[\sum_{j \in S} r_j + \sum_{i \in B} u_i\right]$$

# Analysis of the Algorithm IV

**Theorem**

*RANKING is $(1-1/e)$-competitive.*

**Proof.** Let $M$ be a maximum matching. Then

$$\mathbb{E}[\mathrm{SW}] = \mathbb{E}\left[\sum_{j\in S} r_j + \sum_{i\in B} u_i\right]$$
$$\geq \sum_{(j,i)\in M} \mathbb{E}[r_j + u_i]$$

## Analysis of the Algorithm IV

**Theorem**

*Ranking is* $(1 - 1/e)$*-competitive.*

**Proof.** Let $M$ be a maximum matching. Then

$$\mathbb{E}[\mathrm{SW}] = \mathbb{E}\left[\sum_{j \in S} r_j + \sum_{i \in B} u_i\right]$$

$$\geq \sum_{(j,i) \in M} \mathbb{E}[r_j + u_i]$$

$$\geq \sum_{(j,i) \in M} \left(1 - \frac{1}{e}\right) = \left(1 - \frac{1}{e}\right) |M|. \quad \square$$

# Water-Filling through Continuous Pricing

# Fractional vs. Integral Matchings

It is often simpler to allow our algorithm to compute a fractional matching.

# Fractional vs. Integral Matchings

It is often simpler to allow our algorithm to compute a fractional matching.

Recall that in a bipartite graph, any fractional matching is a convex combination of integral matchings.

# Fractional vs. Integral Matchings

It is often simpler to allow our algorithm to compute a fractional matching.

Recall that in a bipartite graph, any fractional matching is a convex combination of integral matchings.

$\Rightarrow$ The size of a maximum fractional matching is equal to the size of a maximum integral matching.

# Fractional vs. Integral Matchings

It is often simpler to allow our algorithm to compute a fractional matching.

Recall that in a bipartite graph, any fractional matching is a convex combination of integral matchings.

$\Rightarrow$ The size of a maximum fractional matching is equal to the size of a maximum integral matching.

$\Rightarrow$ Computing good fractional matchings online is easier than integral ones.

# Economic Water-Filling

The Water-Filling algorithm (from an economic viewpoint) to construct a fractional matching $(x_{j,i})_{j \in S, i \in B}$:

# Economic Water-Filling

The Water-Filling algorithm (from an economic viewpoint) to construct a fractional matching $(x_{j,i})_{j \in S, i \in B}$:

- The price of each seller is initialized to 0.

The Water-Filling algorithm (from an economic viewpoint) to construct a fractional matching $(x_{j,i})_{j \in S, i \in B}$:

- The price of each seller is initialized to 0.
- Buyers arrive continuously to the market.

# Economic Water-Filling

The Water-Filling algorithm (from an economic viewpoint) to construct a fractional matching $(x_{j,i})_{j \in S, i \in B}$:

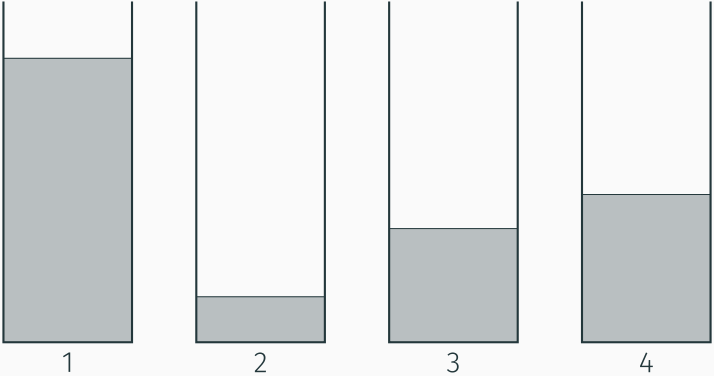
- The price of each seller is initialized to 0.
- Buyers arrive continuously to the market.
- Buyers fractionally buy from sellers $j$ maximizing $1 - p_j$ where $p_j := g\left(\sum_{i \in B} x_{j,i}\right)$ depends monotonically on the fill-level.

The Water-Filling algorithm (from an economic viewpoint) to construct a fractional matching $(x_{j,i})_{j \in S, i \in B}$:

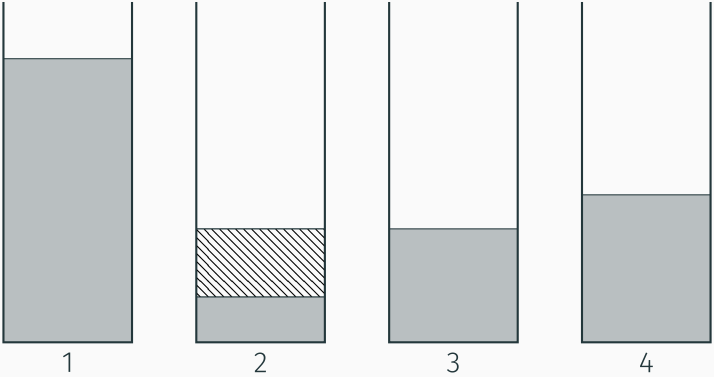
- The price of each seller is initialized to 0.
- Buyers arrive continuously to the market.
- Buyers fractionally buy from sellers $j$ maximizing $1 - p_j$ where $p_j := g\left(\sum_{i \in B} x_{j,i}\right)$ depends monotonically on the fill-level.

It turns out that $g(w) := e^{w-1}$ is the optimal choice for $g$.

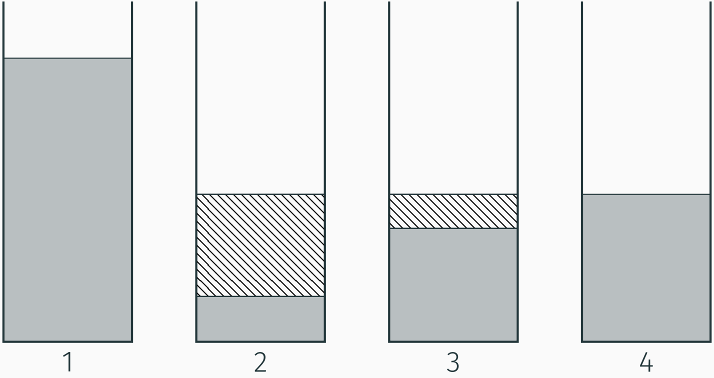
# Water-Filling Example



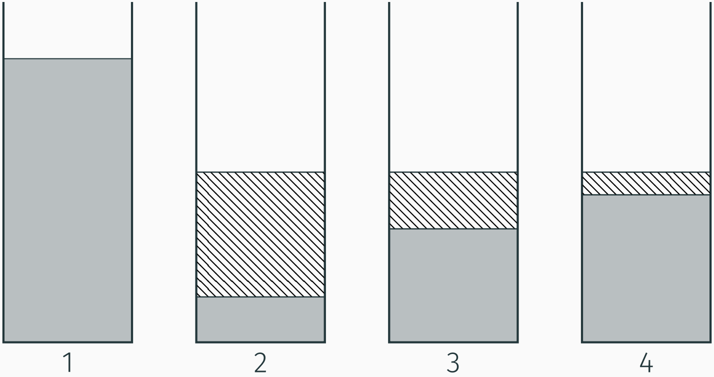
# Water-Filling Example



# Water-Filling Example



# Water-Filling Example



Define revenue and utility as before. Note that social welfare still measures the size of the fractional matching.

**Lemma**

*For any* $(j, i) \in E$*, we have* $r_j + u_i \geq 1 - \frac{1}{e}$ *at the end of the algorithm.*

**Proof.** Let $w$ be the fill-level of seller $j$ at the end of the algorithm. Clearly

$$r_j = \int_0^w e^{t-1}\,\mathrm{d}t = e^{w-1} - \frac{1}{e}.$$

**Proof.** Let $w$ be the fill-level of seller $j$ at the end of the algorithm. Clearly

$$r_j = \int_0^w e^{t-1}\,\mathrm{d}t = e^{w-1} - \frac{1}{e}.$$

Since the price of $j$ would have been at most $e^{w-1}$ at any time in the algorithm, we have

$$u_i \geq 1 - e^{w-1}.$$

**Proof.** Let $w$ be the fill-level of seller $j$ at the end of the algorithm. Clearly

$$r_j = \int_0^w e^{t-1}\,\mathrm{d}t = e^{w-1} - \frac{1}{e}.$$

Since the price of $j$ would have been at most $e^{w-1}$ at any time in the algorithm, we have

$$u_i \geq 1 - e^{w-1}.$$

Thus $r_j + u_i \geq 1 - \frac{1}{e}$. $\square$

## Analysis of WATER-FILLING II

**Theorem**

*WATER-FILLING is $(1 - 1/e)$-competitive.*

# Analysis of Water-Filling II

**Theorem**

*Water-Filling is $(1 - 1/e)$-competitive.*

**Proof.** The proof is identical to Ranking. □

## Remarks

Despite solving a fractional relaxation, Water-Filling has the same competitive ratio as Ranking.

# Remarks

Despite solving a fractional relaxation, Water-Filling has the same competitive ratio as Ranking.

However, Water-Filling is deterministic whereas Ranking is randomized.

# Remarks

Despite solving a fractional relaxation, Water-Filling has the same competitive ratio as Ranking.

However, Water-Filling is deterministic whereas Ranking is randomized.

Both algorithms can be easily generalized to weights on the sellers by scaling both the price and value of each seller by the seller's weight.

# Edge-Weighted Online Bipartite Matching

# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

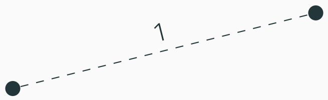
Unfortunately, no algorithm has constant competitive ratio:

# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

Unfortunately, no algorithm has constant competitive ratio:



In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

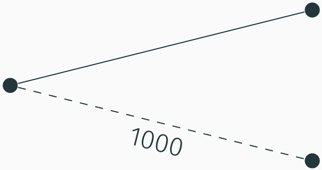
Unfortunately, no algorithm has constant competitive ratio:

# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

Unfortunately, no algorithm has constant competitive ratio:



# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

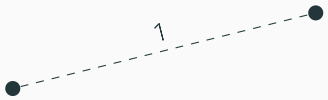
Unfortunately, no algorithm has constant competitive ratio:

# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

Unfortunately, no algorithm has constant competitive ratio:

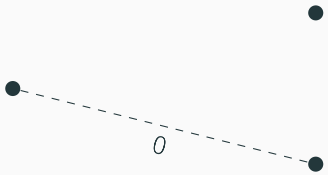


# Edge-Weighted Online Bipartite Matching

In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

Unfortunately, no algorithm has constant competitive ratio:



In the Edge-Weighted Online Bipartite Matching Problem, every edge comes with a value $v_{ji}$.

The goal is to maximize the value of matched edges.

Unfortunately, no algorithm has constant competitive ratio:

# Free Disposal

In order to obtain a meaningful setting, we need an extra condition:

**Definition**

An online matching problem allows free disposal if the offline vertices are allowed drop previously matched online vertices.

# Free Disposal

In order to obtain a meaningful setting, we need an extra condition:

**Definition**

An online matching problem allows free disposal if the offline vertices are allowed drop previously matched online vertices.

$\Rightarrow$ Greedy algorithm is now $\frac{1}{2}$-competitive!

# Free Disposal

In order to obtain a meaningful setting, we need an extra condition:

**Definition**

An online matching problem allows free disposal if the offline vertices are allowed drop previously matched online vertices.

$\Rightarrow$ GREEDY algorithm is now $\frac{1}{2}$-competitive!

$\Rightarrow$ From the economic viewpoint, we allow new buyers to buy out previous buyers!

# Description of the Algorithm

We want to use WATER-FILLING to provide a $(1 - \frac{1}{e})$-competitive algorithm.

# Description of the Algorithm

We want to use Water-Filling to provide a $(1 - \frac{1}{e})$-competitive algorithm.

Buyers maximize utility, i.e. $v_{j,i} - p_{j,i}$ over sellers $j$.

# Description of the Algorithm

We want to use WATER-FILLING to provide a $(1 - \frac{1}{e})$-competitive algorithm.

Buyers maximize utility, i.e. $v_{j,i} - p_{j,i}$ over sellers $j$.

We assume free disposal so buyers can continously buy out previous buyers.

# Description of the Algorithm

We want to use Water-Filling to provide a $(1 - \frac{1}{e})$-competitive algorithm.

Buyers maximize utility, i.e. $v_{j,i} - p_{j,i}$ over sellers $j$.

We assume free disposal so buyers can continously buy out previous buyers.

Wlog. we assume every vertex is initially matched along value 0 edges.

# Description of the Algorithm

We want to use WATER-FILLING to provide a $(1 - \frac{1}{e})$-competitive algorithm.

Buyers maximize utility, i.e. $v_{j,i} - p_{j,i}$ over sellers $j$.

We assume free disposal so buyers can continously buy out previous buyers.

Wlog. we assume every vertex is initially matched along value 0 edges.

**How do we choose the prices?**

# Contour Pricing

Define a $w_{x,j} : \mathbb{R}_{\geq 0} \to [0, 1]$ for any fractional matching $x$ and seller $j \in S$:

# Contour Pricing

Define a $w_{x,j} : \mathbb{R}_{\geq 0} \to [0,1]$ for any fractional matching $x$ and seller $j \in S$:

$w_{x,j}(t)$ represents the fraction of good $j$ which is assigned in $x$ on edges of value at least $t$, i.e.

$$w_{x,j}(t) := \sum_{i:v_{j,i} \geq t} x_{i,j}.$$

# Contour Pricing

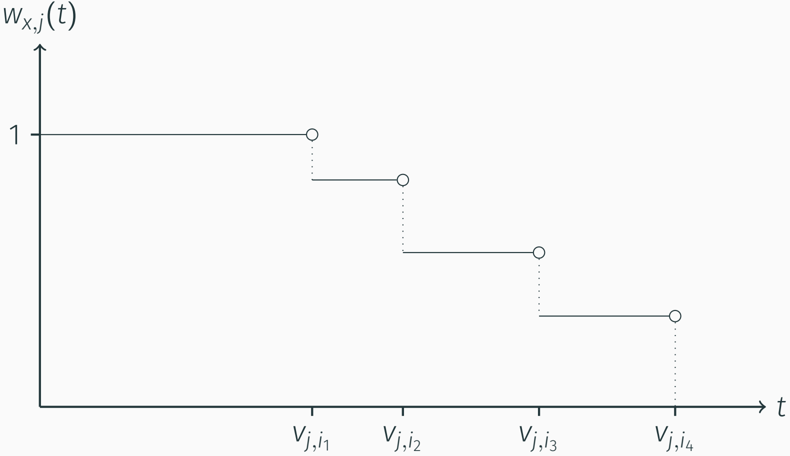
Define a $w_{x,j} : \mathbb{R}_{\geq 0} \to [0, 1]$ for any fractional matching $x$ and seller $j \in S$:

$w_{x,j}(t)$ represents the fraction of good $j$ which is assigned in $x$ on edges of value at least $t$, i.e.

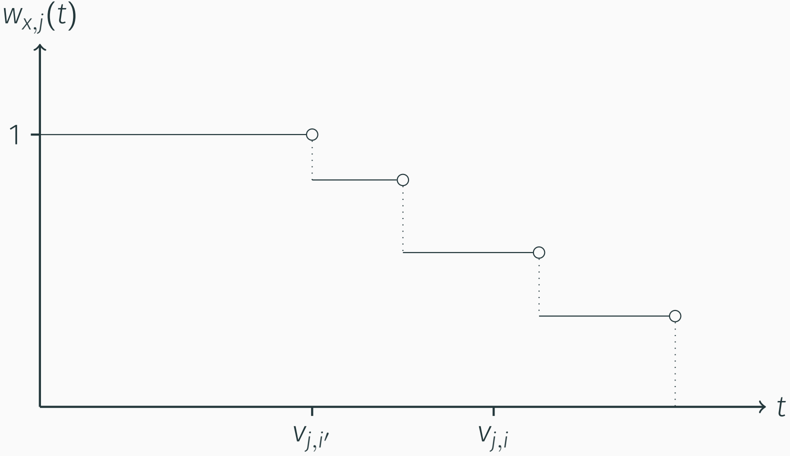
$$w_{x,j}(t) := \sum_{i : v_{j,i} \geq t} x_{i,j}.$$

This is the value contour.

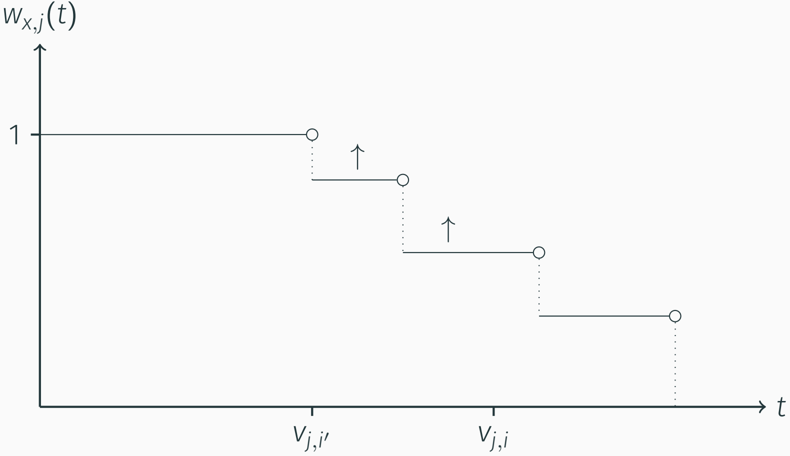
# Contour Pricing II



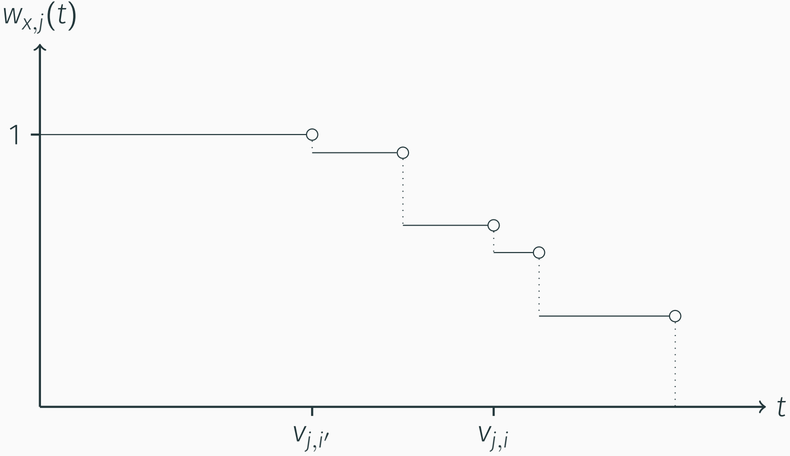
# Contour Pricing II



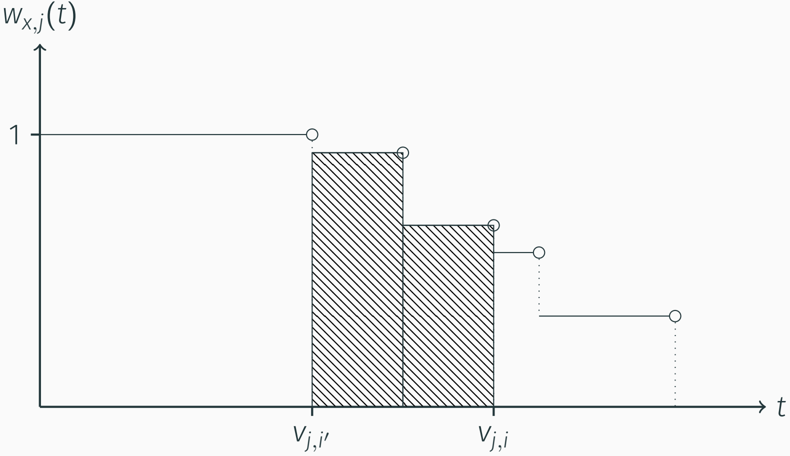
# Contour Pricing II



# Contour Pricing II



# Contour Pricing II



# Contour Pricing III

This leads us to hypothesize a price of

$$\int_{v_{j,i'}}^{v_{j,i}} g(w_{x,j}(t))\,\mathrm{d}t.$$

# Contour Pricing III

This leads us to hypothesize a price of

$$\int_{v_{j,i'}}^{v_{j,i}} g(w_{x,j}(t))\,\mathrm{d}t.$$

It turns out that $g(w) = e^{w-1}$ is once again the optimal choice!

# Contour Pricing III

This leads us to hypothesize a price of

$$\int_{v_{j,i'}}^{v_{j,i}} g(w_{x,j}(t))\,\mathrm{d}t.$$

It turns out that $g(w) = e^{w-1}$ is once again the optimal choice!

However, buyer $i$ must also buy out $i'$, so the total price is

$$v_{j,i'} + \int_{v_{j,i'}}^{v_{j,i}} e^{w_{x,j}(t)-1}\,\mathrm{d}t = \int_{0}^{v_{j,i}} e^{w_{x,j}(t)-1}\,\mathrm{d}t.$$

# Revenue and Utility

Revenue $r_j$ and utility $u_i$ are still defined as before.

# Revenue and Utility

Revenue $r_j$ and utility $u_i$ are still defined as before.

We even get

$$\mathrm{SW} = \sum_{(j,i) \in E} v_{j,i} x_{j,i} = \sum_{j \in S} r_j + \sum_{i \in B} u_i .$$

# Revenue and Utility

Revenue $r_j$ and utility $u_i$ are still defined as before.

We even get

$$\mathrm{SW} = \sum_{(j,i)\in E} v_{j,i} x_{j,i} = \sum_{j\in S} r_j + \sum_{i\in B} u_i.$$

Note that due to buyouts, utilities never decrease!

# Analysis of the Algorithm

**Lemma**

*For any $(j, i) \in E$, we have $r_j + u_i \geq \left(1 - \frac{1}{e}\right) v_{j,i}$ at the end of the algorithm.*

**Lemma**

*For any $(j, i) \in E$, we have $r_j + u_i \geq \left(1 - \frac{1}{e}\right) v_{j,i}$ at the end of the algorithm.*

**Proof.** Let $x$ be the allocation at the end of the algorithm. Then the price of $j$ for $i$ would have been at most

$$\int_0^{v_{j,i}} e^{w_{x,j}(t)-1} \, \mathrm{d}t$$

at any point in the algorithm.

**Lemma**

*For any* $(j, i) \in E$*, we have* $r_j + u_i \geq \left(1 - \frac{1}{e}\right) v_{j,i}$ *at the end of the algorithm.*

**Proof.** Let $x$ be the allocation at the end of the algorithm. Then the price of $j$ for $i$ would have been at most

$$\int_0^{v_{j,i}} e^{w_{x,j}(t)-1}\,\mathrm{d}t$$

at any point in the algorithm.

$$\Rightarrow u_i \geq v_{j,i} - \int_0^{v_{j,i}} e^{w_{x,j}(t)-1}\,\mathrm{d}t.$$

# Analysis of the Algorithm II

However, one can see that

$$r_j = \int_0^\infty \int_0^{w_{x,j}(t)} e^{s-1}\,\mathrm{d}s\,\mathrm{d}t = \int_0^\infty \left( e^{w_{x,j}(t)-1} - \frac{1}{e} \right) \mathrm{d}t.$$

However, one can see that

$$r_j = \int_0^\infty \int_0^{w_{x,j}(t)} e^{s-1}\,\mathrm{d}s\,\mathrm{d}t = \int_0^\infty \left(e^{w_{x,j}(t)-1} - \frac{1}{e}\right)\,\mathrm{d}t.$$

$$\begin{aligned}\Rightarrow r_j + u_i &\geq v_{j,i} - \int_0^{v_{j,i}} e^{w_{x,j}(t)-1}\,\mathrm{d}t + \int_0^\infty \left(e^{w_{x,j}(t)-1} - \frac{1}{e}\right)\,\mathrm{d}t \\ &\geq v_{j,i} - \int_0^{v_{j,i}} \frac{1}{e} = \left(1 - \frac{1}{e}\right) v_{j,i}. \quad \square\end{aligned}$$

# Analysis of the Algorithm III

**Theorem**

*WATER-FILLING is $(1 - 1/e)$-competitive for edge weights under the free disposal assumption.*

# Analysis of the Algorithm III

**Theorem**

*WATER-FILLING is* $(1 - 1/e)$*-competitive for edge weights under the free disposal assumption.*

**Proof.** Let $M$ be the matching maximizing total edge value.

$$\mathrm{SW} = \sum_{j \in S} r_j + \sum_{i \in B} u_i$$

# Analysis of the Algorithm III

**Theorem**

*Water-Filling is* $(1-1/e)$*-competitive for edge weights under the free disposal assumption.*

**Proof.** Let $M$ be the matching maximizing total edge value.

$$\begin{aligned} \mathrm{SW} &= \sum_{j \in S} r_j + \sum_{i \in B} u_i \\ &\geq \sum_{(j,i) \in M} r_j + u_i \end{aligned}$$

# Analysis of the Algorithm III

## Theorem

*Water-Filling is $(1 - 1/e)$-competitive for edge weights under the free disposal assumption.*

**Proof.** Let $M$ be the matching maximizing total edge value.

$$\begin{aligned}
\mathrm{SW} &= \sum_{j \in S} r_j + \sum_{i \in B} u_i \\
&\geq \sum_{(j,i) \in M} r_j + u_i \\
&\geq \sum_{(j,i) \in M} \left(1 - \frac{1}{e}\right) v_{j,i} = \left(1 - \frac{1}{e}\right) v(M). \quad \square
\end{aligned}$$

# Conclusion

# Final Remarks

The economic viewpoint is a powerful and simple way to think about online matching problems.

# Final Remarks

The economic viewpoint is a powerful and simple way to think about online matching problems.

There are still many open problems in online matching theory, especially when it comes to general graphs, wait costs, stochasticity, etc.

# Final Remarks

The economic viewpoint is a powerful and simple way to think about online matching problems.

There are still many open problems in online matching theory, especially when it comes to general graphs, wait costs, stochasticity, etc.

I hope that beyond being a useful tool for exposition, the economic viewpoint may inspire solutions to new variants of online matching!

Thank You!