(Fractional) Online Stochastic Matching via Fine-Grained Offline Statistics

Zhihao Gavin Tang[1] Hongxun Wu[2] Jinzhao Wu[3]

[1] ITCS, Shanghai University of Finance and Economics

[2] IIIS, Tsinghua University

[3] CFCS, Peking University

Introduction

Our framework

Algorithms / Analysis

Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.

Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

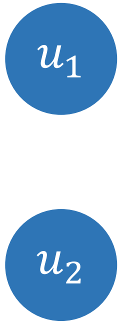
- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.
  - Upon arrival, the set $t_j \in 2^L$ of its neighbors is sampled.
  - $t_j \sim D_j$. Distributions are known upfront.

Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.
  - Upon arrival, the set $t_j \in 2^L$ of its neighbors is sampled.
  - $t_j \sim D_j$. Distributions are known upfront.

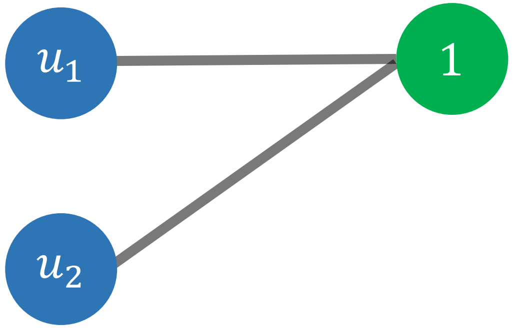


Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.
  - Upon arrival, the set $t_j \in 2^L$ of its neighbors is sampled.
  - $t_j \sim D_j$. Distributions are known upfront.



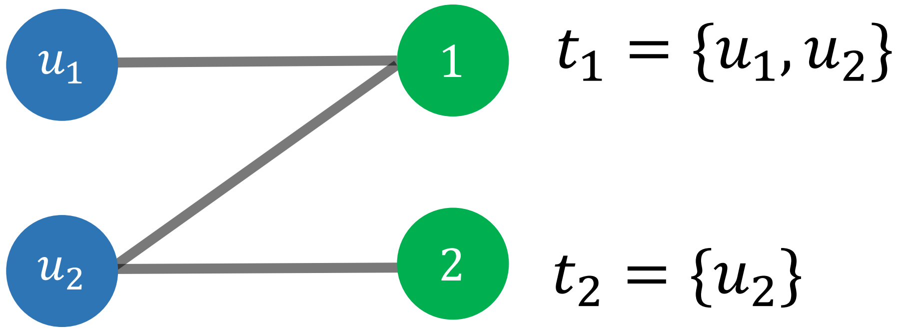
$t_1 = \{u_1, u_2\}$

Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.
  - Upon arrival, the set $t_j \in 2^L$ of its neighbors is sampled.
  - $t_j \sim D_j$. Distributions are known upfront.



Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.
  - Upon arrival, the set $t_j \in 2^L$ of its neighbors is sampled.
  - $t_j \sim D_j$. Distributions are known upfront.

OUTPUT Algorithm must decide irrevocably matching for $j \in R$.

Online Stochastic Matching

[Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

INPUT Bipartite graph $G = (L \cup R, E)$

- Vertices $u \in L$ are **offline**.
- Vertices $j \in R$ are **online**.
  - Upon arrival, the set $\mathrm{t_j} \in 2^L$ of its neighbors is sampled.
  - $t_j \sim D_j$. Distributions are known upfront.

Goal

- Unweighted: Maximize the cardinality of the matching
- Vertex Weighted: Maximize the total weight of matched $\mathrm{u} \in L$.

Competitive Ratio

Metric The ratio between algorithm and offline optimum.

$$\mu = \min_{G, D_1, D_2, \cdots, D_n} \frac{\mathrm{E}\,[\,\mathrm{ALG}(G)\,]}{\mathrm{E}\,[\,\mathrm{OPT}(G)\,]}$$

Previous Works

- **IID arrival:** Type distributions D_j are the same for all $j \in R$.

Arrival	Goal	Ratio	
IID	Unweighted	0.711	[Huang and Shu, 2021]
	Vertex-weighted	0.701	
Non-IID	Vertex-weighted	$1 - 1/e \approx 0.632$	[Aggarwal, Goel, Karande, and Mehta, 2011]

Our Results

- **IID arrival:** Type distributions D_j are the same for all $j \in R$.

Arrival	Goal	Ratio	Ours
IID	Unweighted	0.711	0.704
	Vertex-weighted	0.701	
Non-IID	Vertex-weighted	$1 - 1/e \approx 0.632$	0.666

Our Results

- **IID arrival:** Type distributions D_j are the same for all $j \in R$.

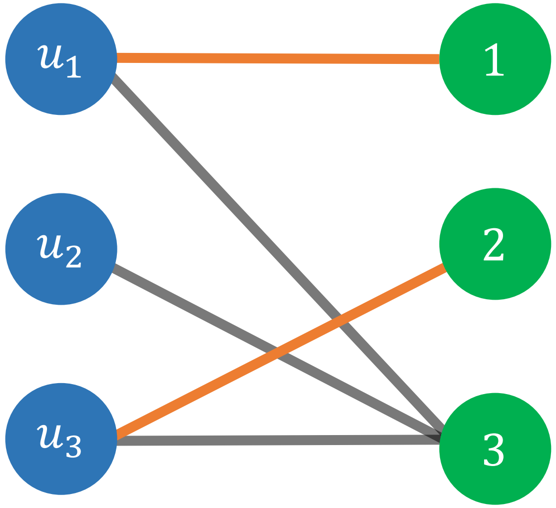
Arrival	Goal	Ratio	Ours
IID	Unweighted	0.711	0.704
	Vertex-weighted	0.701	
Non-IID	Vertex-weighted	$1 - 1/e \approx 0.632$	0.666

Parallel to our work, Huang, Shu, and Yan improved the ratio for IID vertex-weighted setting to 0.716 .

Key Idea

- **Warm up:** One – Choices Algorithm
  - When j arrives, we sample **one** neighbor of it.
  - For each neighbor u,

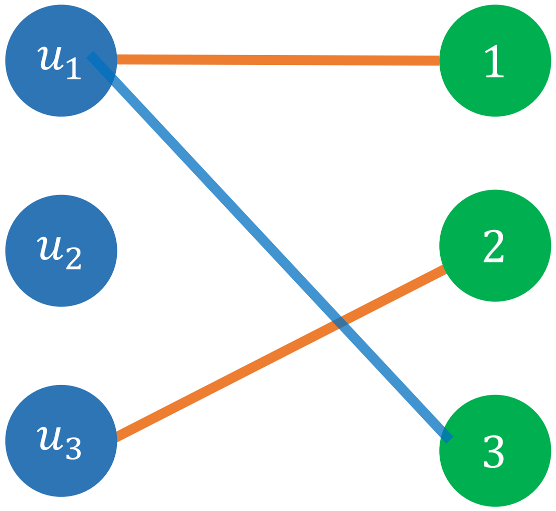
    it is sampled with probability $\Pr[\,(u,j) \in \mathrm{OPT} \mid t_j\,]$.



Key Idea

- **Warm up:** One – Choices Algorithm
  - When j arrives, we sample **one** neighbor of it.
  - For each neighbor u,

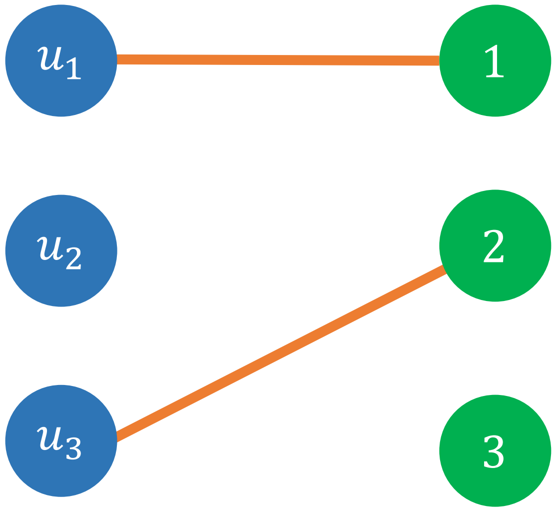
    it is sampled with probability $\Pr[\,(u,j) \in \mathrm{OPT} \mid t_j\,]$.
  - Then we try to match j with the sampled neighbor.



Key Idea

- **Warm up:** One – Choices Algorithm
  - When j arrives, we sample **one** neighbor of it.
  - For each neighbor u,

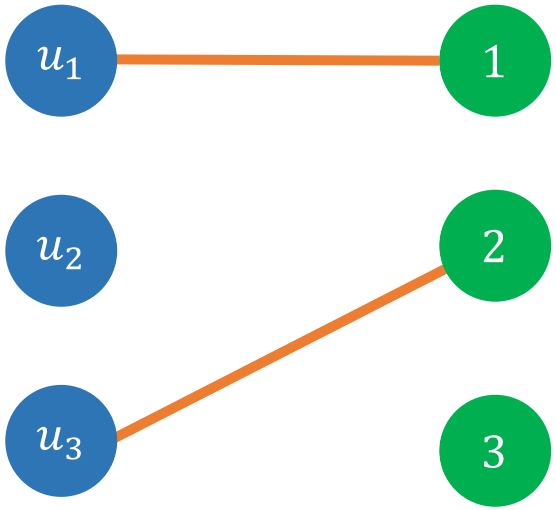
    it is sampled with probability $\Pr[\,(u,j) \in \text{OPT} \mid t_j\,]$.
  - Then we try to match j with the sampled neighbor.



  - In total, We try each (u,j) with probability $p_{u,j} = \Pr[(u,j) \in \text{OPT}]$.

Key Idea

- **Warm up:** One – Choices Algorithm
  - When j arrives, we sample **one** neighbor of it.
  - For each neighbor u,
    it is sampled with probability $\Pr[\,(u,j) \in \mathrm{OPT} \mid t_j\,]$.
  - Then we try to match j with the sampled neighbor.

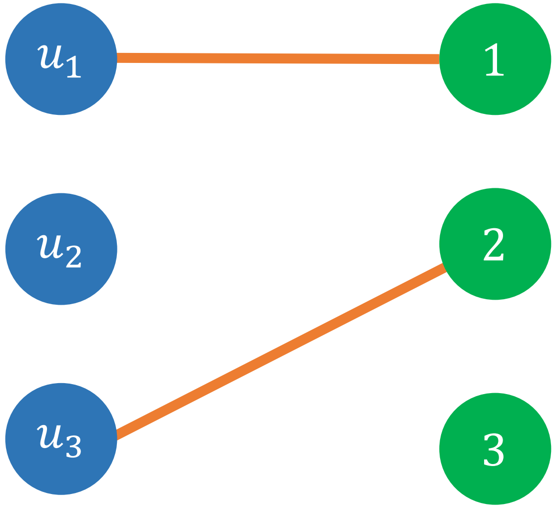


  - In total, We try each (u,j) with probability $p_{u,j} = \Pr[(u,j) \in \mathrm{OPT}]$.
  - Define $p_u \coloneqq \sum_j p_{u,j} = \Pr[u \in \mathrm{OPT}]$.

Key Idea

- **Warm up:** One – Choices Algorithm
  - When j arrives, we sample **one** neighbor of it.
  - For each neighbor u,

    it is sampled with probability $\Pr[\,(u,j) \in \text{OPT} \mid t_j\,]$.
  - Then we try to match j with the sampled neighbor.

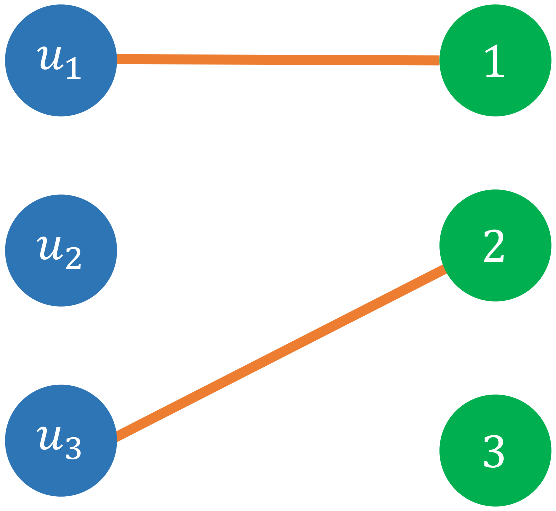


  - In total, We try each (u,j) with probability $p_{u,j} = \Pr[(u,j) \in \text{OPT}]$.
  - Define $p_u \coloneqq \sum_j p_{u,j} = \Pr[u \in \text{OPT}]$.
  - $\Pr[\,u \in \text{ALG}\,] = 1 - \prod_j (1 - p_{u,j}) \geq 1 - e^{-p_u} \geq \left(1 - \frac{1}{e}\right) p_u$

Key Idea

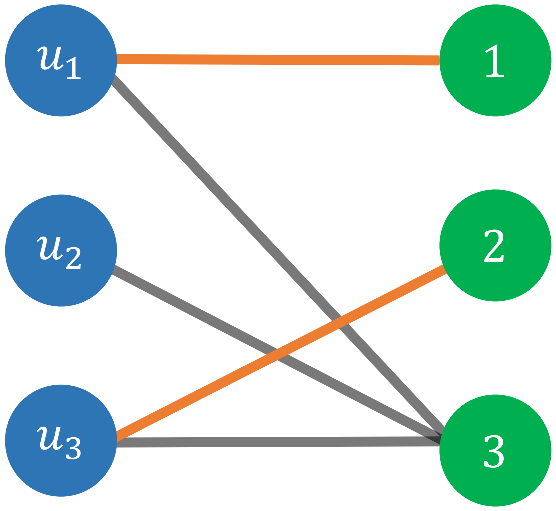
- **Warm up:** One – Choices Algorithm
  - Define $p_u \coloneqq \Pr[\, u \in \mathrm{OPT}] = \sum_j p_{u,j}$.
  - $\Pr[\, u \in \mathrm{ALG}\,] \geq \left(1 - \frac{1}{e}\right) p_u$

$$
\begin{aligned}
\mathrm{E}[\,\mathrm{ALG}\,] &\geq \sum_u w_u \cdot \Pr[u \in ALG] \\
&\geq \left(1 - \frac{1}{e}\right) \sum_u w_u p_u \\
&= \left(1 - \frac{1}{e}\right) \mathrm{E}[\mathrm{OPT}]
\end{aligned}
$$



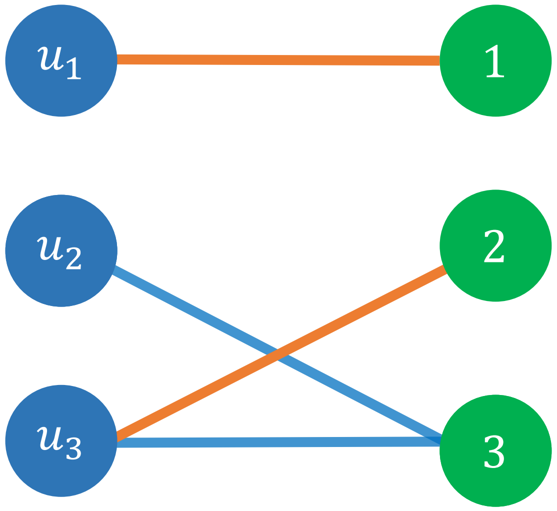
Key Idea

- **Previous approach:** Two – Choice Algorithm
  [Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]
  [Manshadi, Gharan, and Saberi, 2012]
  [Jaillet and Lu, 2014]
  [Brubach, Sankararaman, Srinivasan, and Xu, 2016]
  [Huang and Shu, 2021]



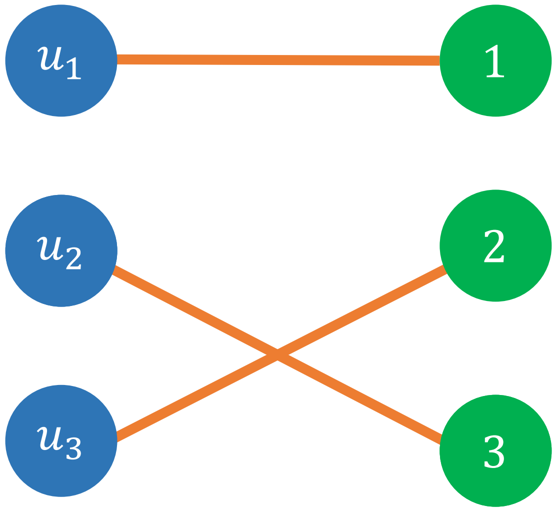
Key Idea

- **Previous approach:** Two – Choice Algorithm
  [Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]
  [Manshadi, Gharan, and Saberi, 2012]
  [Jaillet and Lu, 2014]
  [Brubach, Sankararaman, Srinivasan, and Xu, 2016]
  [Huang and Shu, 2021]



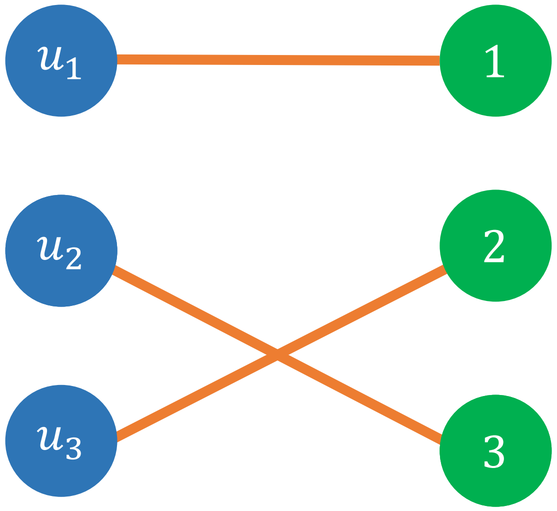
Key Idea

- **Previous approach:** Two – Choice Algorithm
  [Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]
  [Manshadi, Gharan, and Saberi, 2012]
  [Jaillet and Lu, 2014]
  [Brubach, Sankararaman, Srinivasan, and Xu, 2016]
  [Huang and Shu, 2021]



Key Idea

- **Previous approach:** Two – Choice Algorithm
  [Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]
  [Manshadi, Gharan, and Saberi, 2012]
  [Jaillet and Lu, 2014]
  [Brubach, Sankararaman, Srinivasan, and Xu, 2016]
  [Huang and Shu, 2021]



Key Idea: Multiway Online Selection.
[Gao, He, Huang, Nie, Yuan, and Zhong, 2021]
[Blanc and Charikar, 2021]

Key Idea

- **Previous approach:** Two – Choice Algorithm

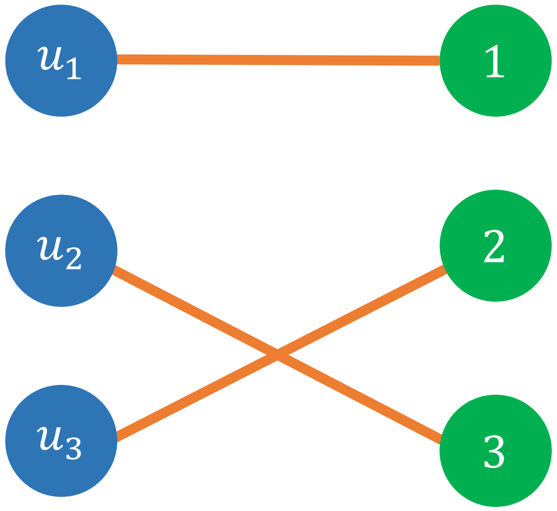
  [Feldman, Mehta, Mirrokni, and Muthukrishnan, 2009]

  [Manshadi, Gharan, and Saberi, 2012]

  [Jaillet and Lu, 2014]

  [Brubach, Sankararaman, Srinivasan, and Xu, 2016]

  [Huang and Shu, 2021]



Key Idea: Multiway Online Selection.

[Gao, He, Huang, Nie, Yuan, and Zhong, 2021]

[Blanc and Charikar, 2021]

Parallel to our work, Huang, Shu, and Yan also exploit the power of multi-selection.

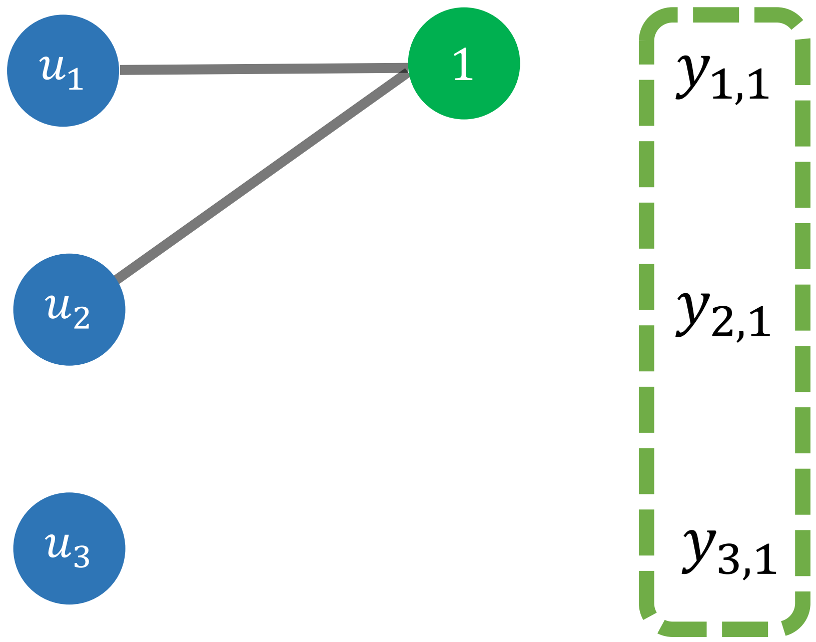
Introduction

Our framework

Algorithms / Analysis

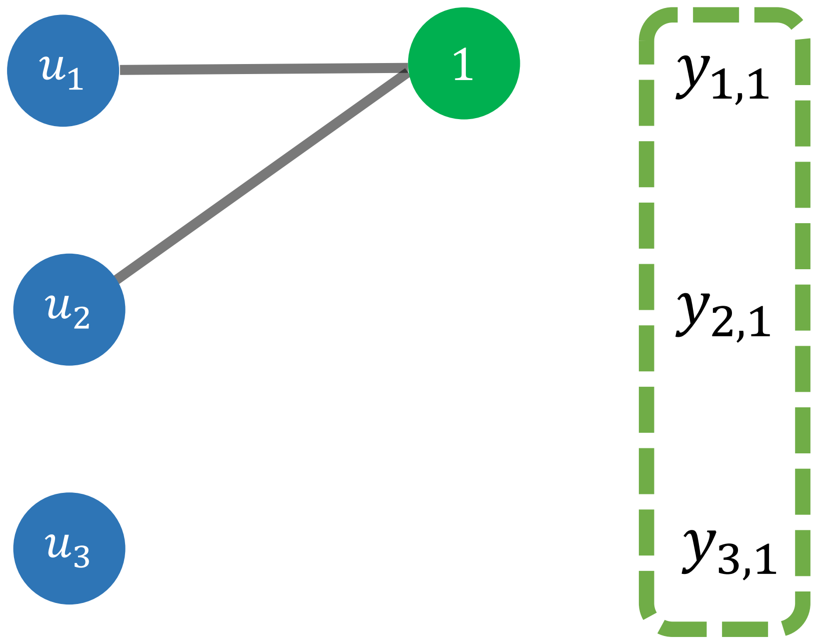
Fractional Matching

Upon the arrival of each $j \in R$, the algorithm matches it with each $u \in L$ with fraction $y_{u,j}$.



Fractional Matching

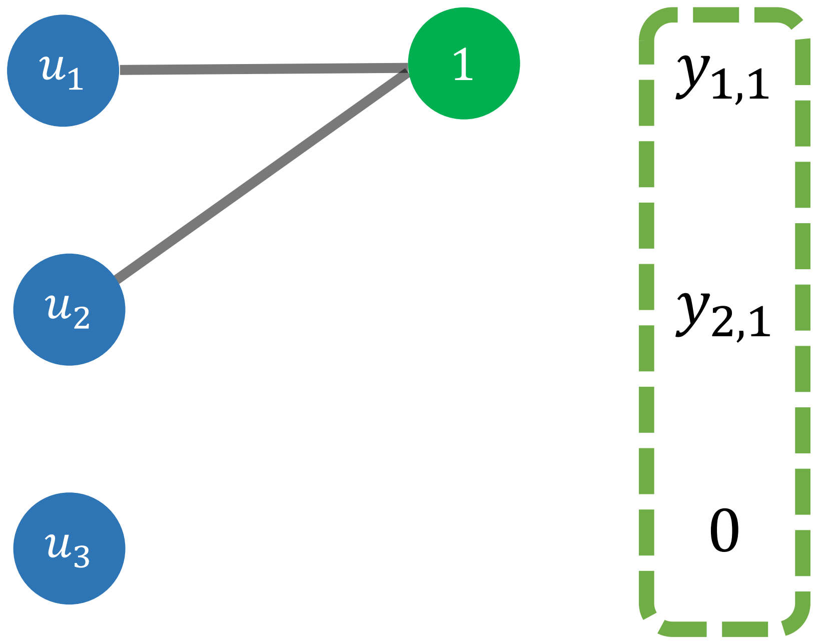
Upon the arrival of each $j \in R$, the algorithm matches it with each $u \in L$ with fraction $y_{u,j}$.



- $\sum_u y_{u,j} \leq 1$.

Fractional Matching

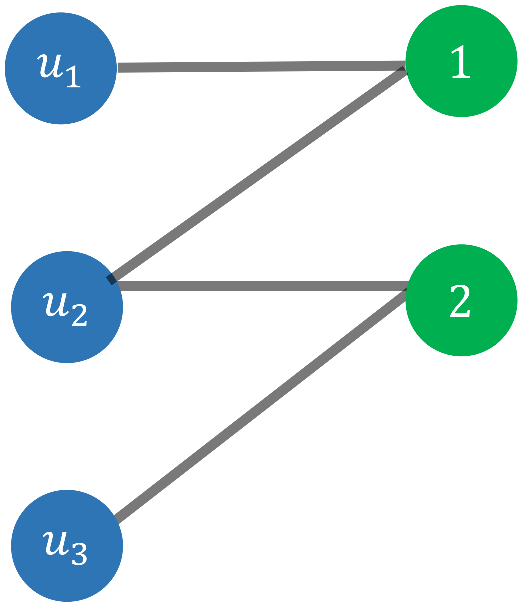
Upon the arrival of each $j \in R$, the algorithm matches it with each $u \in L$ with fraction $y_{u,j}$.



- $\sum_u y_{u,j} \leq 1$.
- $y_{u,j} = 0$ if there is no such edge, i.e. $u \notin t_j$

Fractional Matching

Upon the arrival of each $j \in R$, the algorithm matches it with each $u \in L$ with fraction $y_{u,j}$.

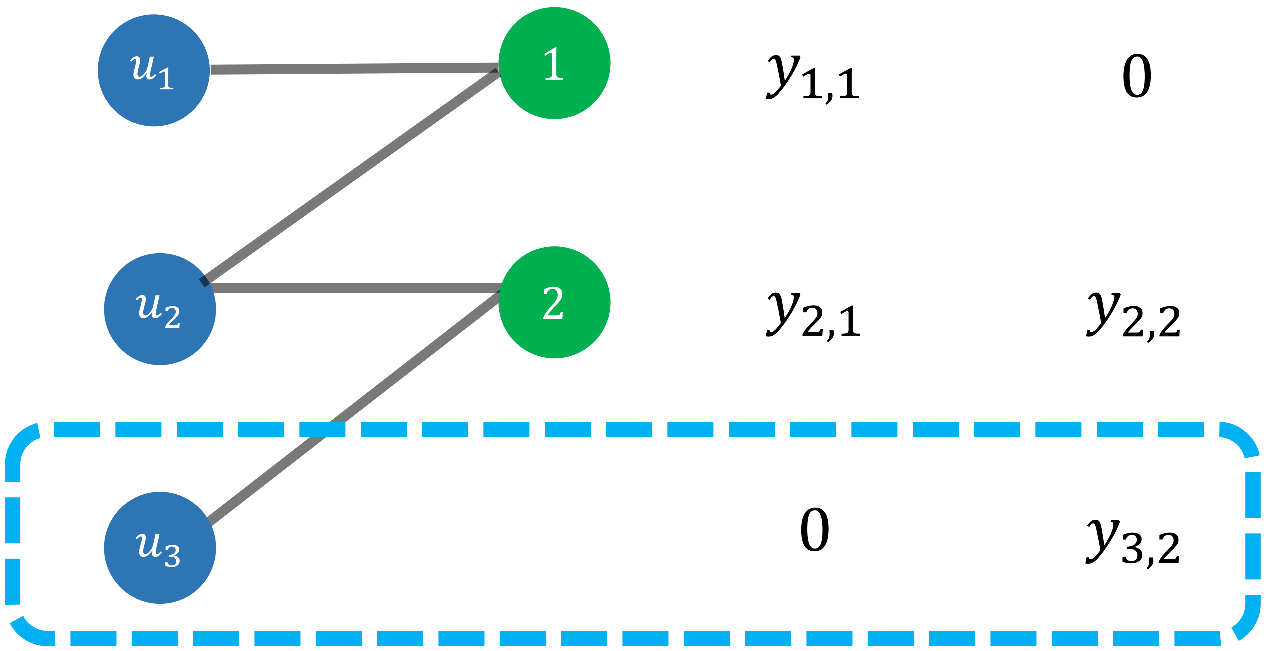


$y_{1,1}$	0
$y_{2,1}$	$y_{2,2}$
0	$y_{3,2}$

- $\sum_u y_{u,j} \leq 1$.
- $y_{u,j} = 0$ if there is no such edge, i.e. $u \notin t_j$

Fractional Matching

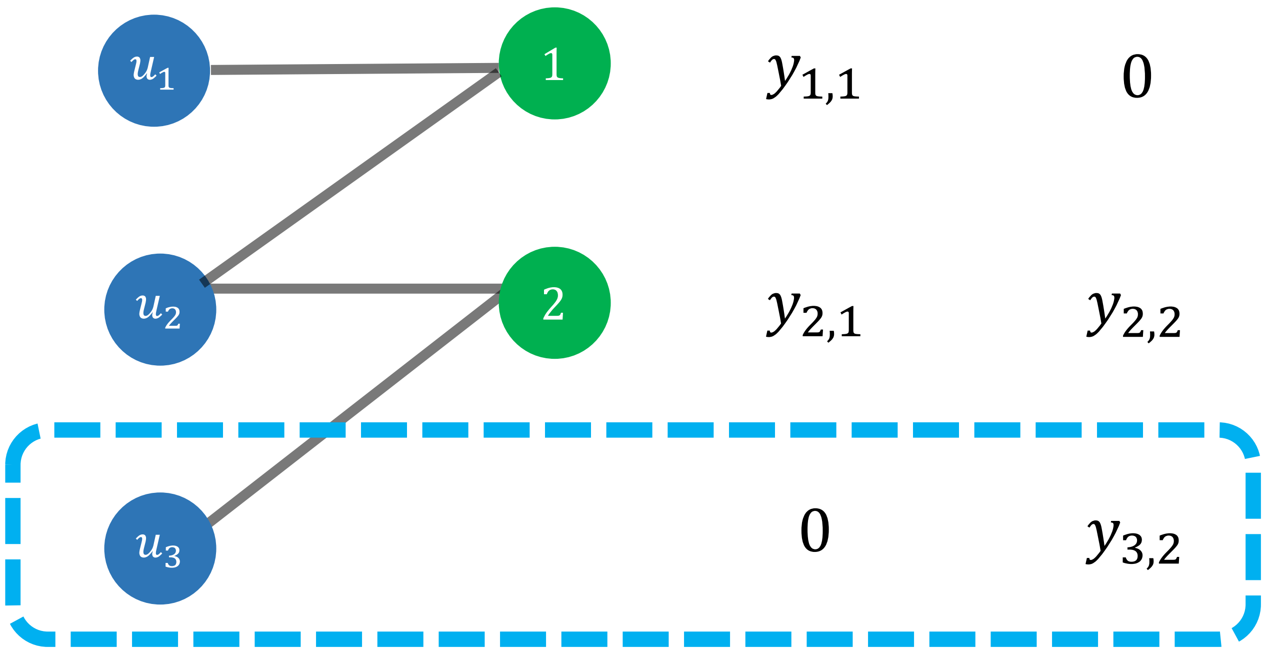
Upon the arrival of each $j \in R$, the algorithm matches it with each $u \in L$ with fraction $y_{u,j}$.



- We define $y_u \coloneqq \sum_j y_{u,j}$.

Fractional Matching

Upon the arrival of each $j \in R$, the algorithm matches it with each $u \in L$ with fraction $y_{u,j}$.

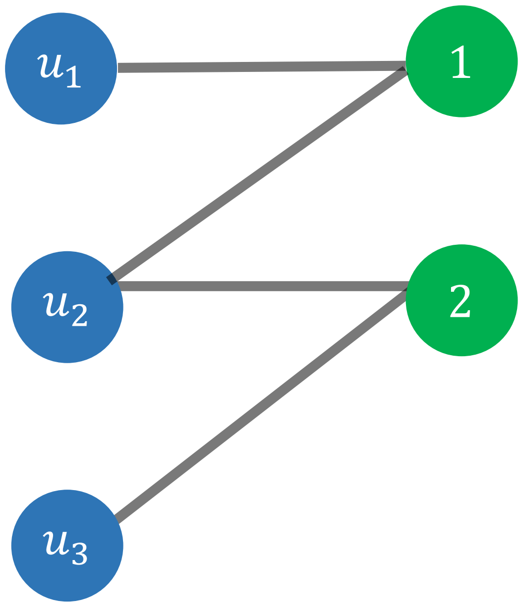


- We define $y_u := \sum_j y_{u,j}$.
- The weight of fractional matching is defined as

$$\mathrm{FRAC} := \sum_u w_u \cdot \min(y_u, 1)$$

Independent Rounding

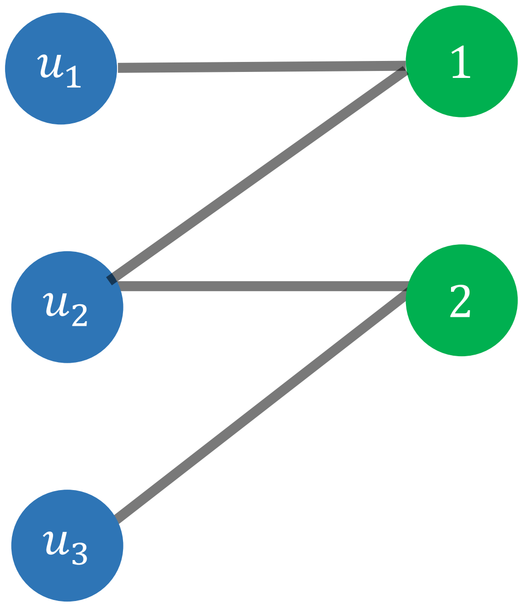
For each $j \in R$, we sample **one** neighbor and try.
Each neighbor u is sampled with probability $y_{u,j}$



$y_{1,1}$	0
$y_{2,1}$	$y_{2,2}$
0	$y_{3,2}$

Independent Rounding

For each $j \in R$, we sample **one** neighbor and try.
Each neighbor u is sampled with probability $y_{u,j}$



$y_{1,1}$	0
$y_{2,1}$	$y_{2,2}$
0	$y_{3,2}$

- The weight of matching afterward is

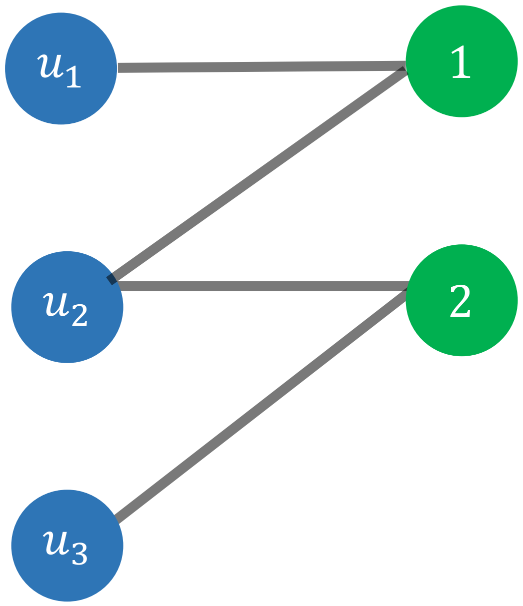
$$\sum_u w_u \cdot \left(1 - \prod_j (1 - y_{u,j})\right)$$

$$= \sum_u w_u \cdot (1 - e^{-y_u})$$

Rounding with Online Correlated Selection

[Gao, He, Huang, Nie, Yuan, and Zhong, 2021] [Blanc and Charikar, 2021]

For each $j \in R$, we apply OCS with y_j as input.

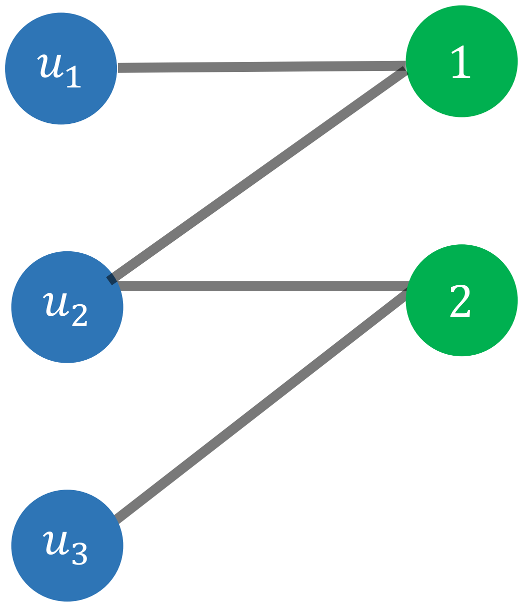


$y_{1,1}$	0
$y_{2,1}$	$y_{2,2}$
0	$y_{3,2}$

Rounding with Online Correlated Selection

[Gao, He, Huang, Nie, Yuan, and Zhong, 2021] [Blanc and Charikar, 2021]

For each $j \in R$, we apply OCS with y_j as input.



$y_{1,1}$	0
$y_{2,1}$	$y_{2,2}$
0	$y_{3,2}$

- The weight of matching afterward is

$$\sum_u w_u \cdot \left(1 - e^{-y_u - 0.5\, y_u^2 - 0.17\, y_u^3}\right)$$

Our Framework

Lemma.

Let $y_u = \sum_j y_{u,j}$. The algorithm would achieve performance

$$\sum_u w_u \cdot f(y_u)$$

where

$$f = \begin{cases} \min(1, y_u) & \text{for fractional matching} \\ & \end{cases}$$

Our Framework

Lemma.

Let $y_u = \sum_j y_{u,j}$. The algorithm would achieve performance

$$\sum_u w_u \cdot f(y_u)$$

where

$$f = \begin{cases} \min(1, y_u) & \text{for fractional matching} \\ 1 - e^{-y_u} & \text{with independent rounding} \end{cases}$$

Our Framework

Lemma.

Let $y_u = \sum_j y_{u,j}$. The algorithm would achieve performance

$$\sum_u w_u \cdot f(y_u)$$

where

$$f = \begin{cases} \min(1, y_u) & \text{for fractional matching} \\ 1 - e^{-y_u - 0.5y_u^2 - 0.18\, y_u^3} & \text{with OCS rounding} \\ 1 - e^{-y_u} & \text{with independent rounding} \end{cases}$$

Introduction

Our framework

Algorithms / Analysis

Unbiased Estimators

Example. When apply $y_{u,j} = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$ and independent sampling, we get exactly one-choice algorithm. (**Independent Estimators**)

Unbiased Estimators

Example. When apply $y_{u,j} = \Pr[(u,j) \in \text{OPT} \mid t_j]$ and independent sampling, we get exactly one-choice algorithm. (**Independent Estimators**)

We consider fractional algorithm with **unbiased estimators**.

Unbiased Estimators

Example. When apply $y_{u,j} = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$ and independent sampling, we get exactly one-choice algorithm. (**Independent Estimators**)

We consider fractional algorithm with **unbiased estimators**.

Estimators $\{\, y_{u,j}(t_1, t_2, \ldots, t_j) \,\}_{u \in L, j \in R}$ are unbiased if

Unbiased Estimators

Example. When apply $y_{u,j} = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$ and independent sampling, we get exactly one-choice algorithm. (**Independent Estimators**)

We consider fractional algorithm with **unbiased estimators**.

Estimators $\{ y_{u,j}(t_1, t_2, \dots, t_j) \}_{u \in L, j \in R}$ are unbiased if

- $\sum_u y_{u,j} \le 1$.

Unbiased Estimators

Example. When apply $y_{u,j} = \mathrm{Pr}[(u, j) \in \mathrm{OPT} \mid t_j]$ and independent sampling, we get exactly one-choice algorithm. (**Independent Estimators**)

We consider fractional algorithm with **unbiased estimators**.

Estimators $\{ y_{u,j}(t_1, t_2, \dots, t_j) \}_{u \in L, j \in R}$ are unbiased if

- $\sum_u y_{u,j} \leq 1$.
- $y_{u,j} = 0$ if there is no such edge.

Unbiased Estimators

Example. When apply $y_{u,j} = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$ and independent sampling, we get exactly one-choice algorithm. (**Independent Estimators**)

We consider fractional algorithm with **unbiased estimators**.

Estimators $\{ y_{u,j}(t_1, t_2, \dots, t_j) \}_{u \in L, j \in R}$ are unbiased if

- $\sum_u y_{u,j} \le 1$.
- $y_{u,j} = 0$ if there is no such edge.
- $y_u = \sum_j y_{u,j}$ has $\mathrm{E}[y_u] = \Pr[\, u \in \mathrm{OPT} \,]$

Main Difficulty

Lemma.

Any **unbiased estimators** $y_{\mathrm{u,j}}$ with $\mathrm{E}[\, f(y_u)\,] \geq \mu \cdot \mathrm{E}[\, y_u \,]$ implies a μ-competitive algorithm.

Main Difficulty

Lemma.

Any **unbiased estimators** $y_{\mathrm{u,j}}$ with $\mathrm{E}[\, f(y_u) \,] \geq \mu \cdot \mathrm{E}[\, y_u \,]$ implies a μ-competitive algorithm.

Proof

$$\mathrm{E}[\,\mathrm{ALG}\,] = \mathrm{E}\left[\sum_u f(y_u) \cdot w_u\right] \geq \mu \cdot \sum_u E[y_u] \cdot w_u$$

$$= \mu \cdot \sum_u \Pr[\, u \in OPT \,] \cdot w_u \;= \mu \cdot \mathrm{E}[\,\mathrm{OPT}\,]$$

Main Difficulty

Lemma.

Any **unbiased estimators** $y_{\mathrm{u,j}}$ with $\mathrm{E}[\, f(y_u)\,] \geq \mu \cdot \mathrm{E}[\, y_u\,]$ implies a μ-competitive algorithm.

Main Difficulty.

For example, let $f(y) = \min(1, y)$.

Main Difficulty

Lemma.

Any **unbiased estimators** $y_{\mathrm{u,j}}$ with $\mathrm{E}[\, f(y_u)\,] \geq \mu \cdot \mathrm{E}[\, y_u \,]$ implies a μ-competitive algorithm.

Main Difficulty.

For example, let $f(y) = \min(1, y)$.

We know $\mathrm{E}[\, y_u \,] = \Pr[\, u \in \mathrm{OPT} \,] \in [0,1]$.

For any **deterministic** y ∈ $[0,1]$, we do have $f(y) = y$.

Main Difficulty

Lemma.

Any **unbiased estimators** $y_{\mathrm{u,j}}$ with $\mathrm{E}[\, f(y_u)\,] \geq \mu \cdot \mathrm{E}[\, y_u \,]$ implies a μ-competitive algorithm.

Main Difficulty.

For example, let $f(y) = \min(1, y)$.

We know $\mathrm{E}[\, y_u \,] = \Pr[\, u \in \mathrm{OPT} \,] \in [0,1]$.
For any **deterministic** y ∈ $[0,1]$, we do have $f(y) = y$.
But y_u is a **random variable**.

Main Difficulty

Lemma.

Any **unbiased estimators** $y_{\mathrm{u,j}}$ with $\mathrm{E}[\, f(y_u)\,] \geq \mu \cdot \mathrm{E}[\, y_u \,]$ implies a μ-competitive algorithm.

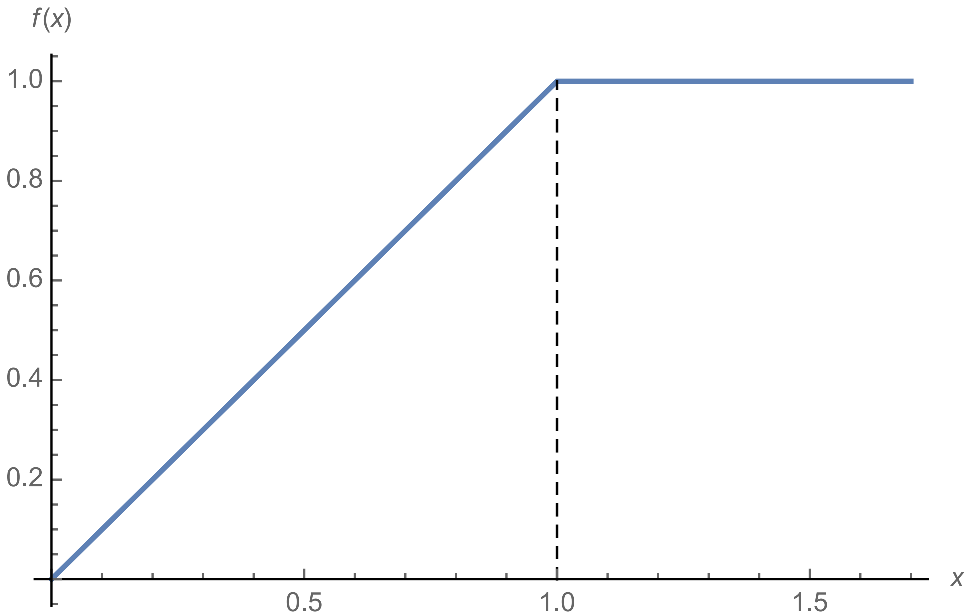
Main Difficulty.

For example, let $f(y) = \min(1, y)$.

We know $\mathrm{E}[\, y_u \,] = \Pr[\, u \in \mathrm{OPT} \,] \in [0,1]$.
For any **deterministic** y ∈ $[0,1]$, we do have $f(y) = y$.
But y_u is a **random variable**. It can be larger than 1!

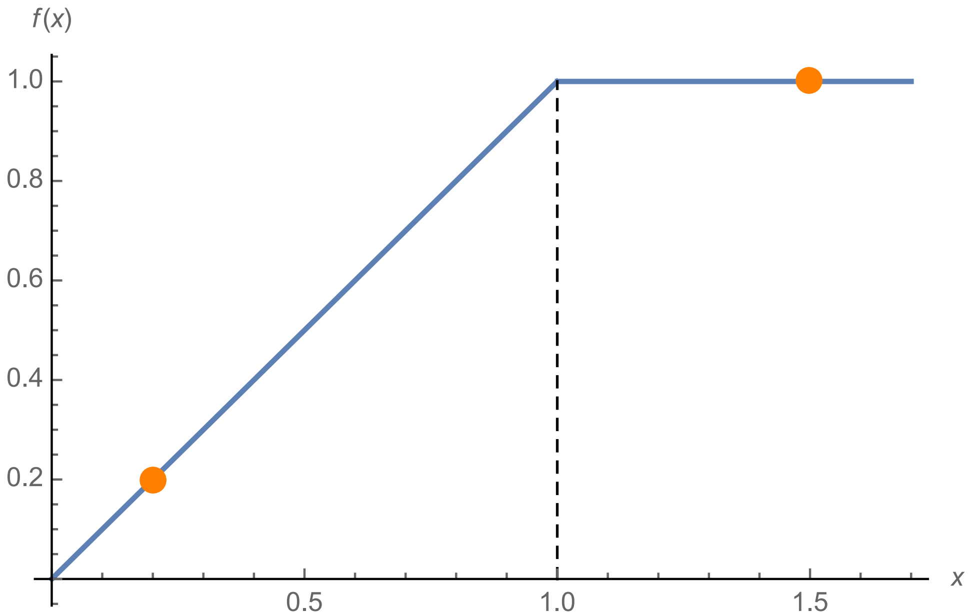
Main Difficulty

Function $f(y)$ is **concave**.



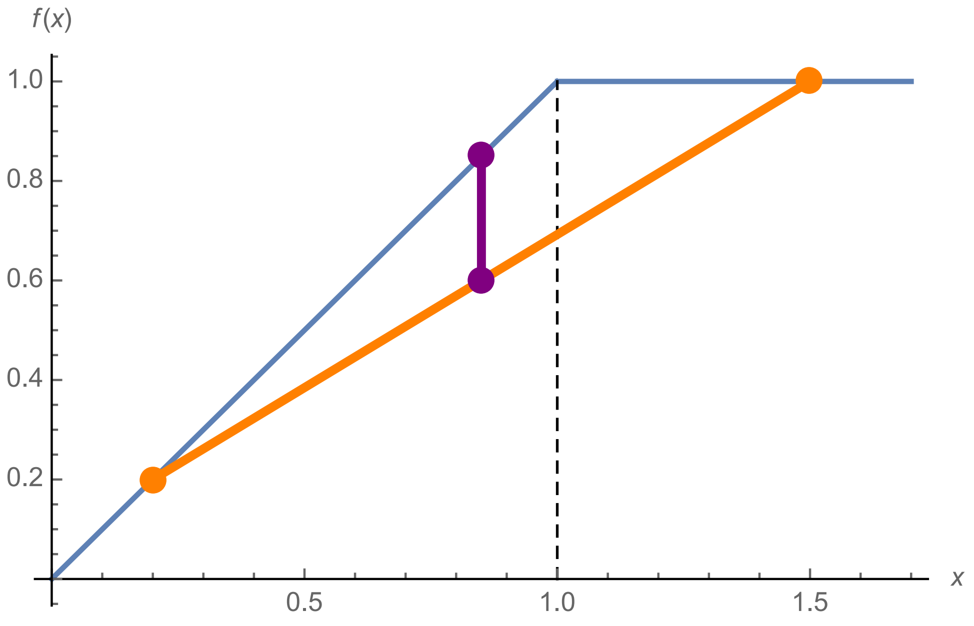
Main Difficulty

Function $f(y)$ is **concave**.
Suppose y_u is a even mixing of 0.2 and 1.5.



Main Difficulty

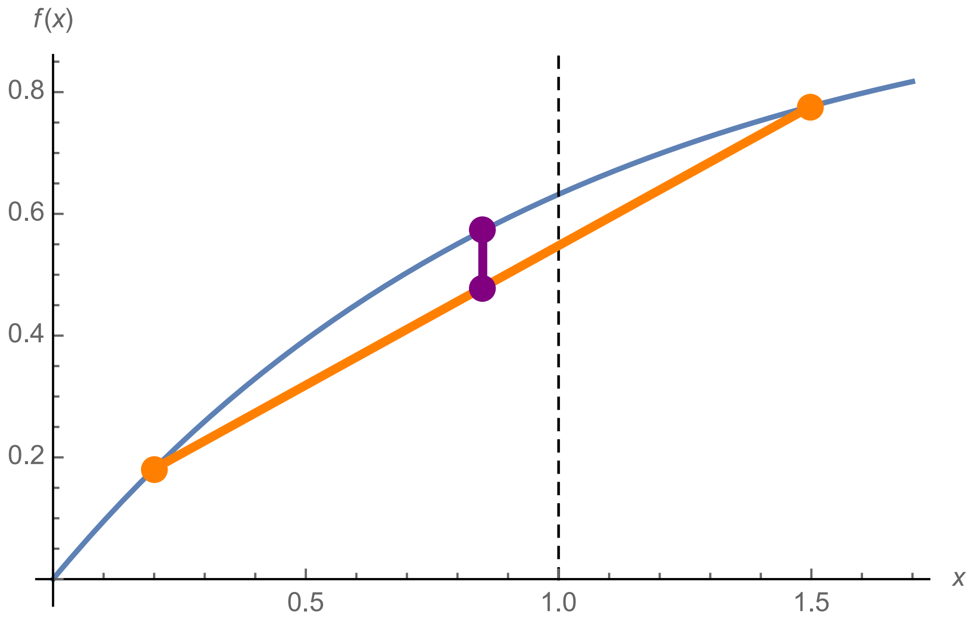
Function $f(y)$ is **concave**.
Suppose y_u is a even mixing of 0.2 and 1.5.

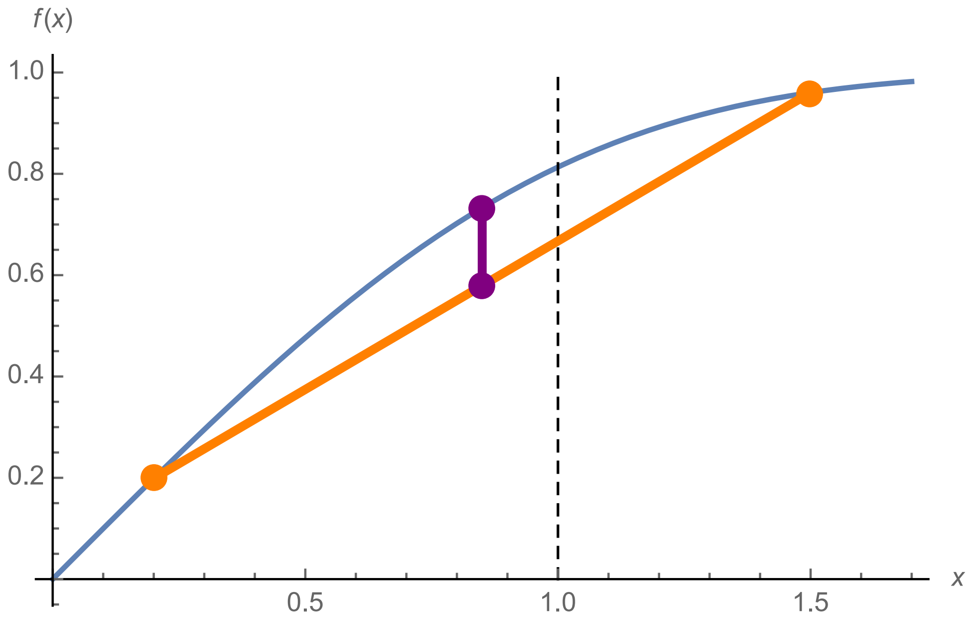


$$\mathrm{E}[\, f(y_u)\,] \ll \mathrm{E}[\, y_u\,]$$

Main Difficulty

Same issue for other two functions.

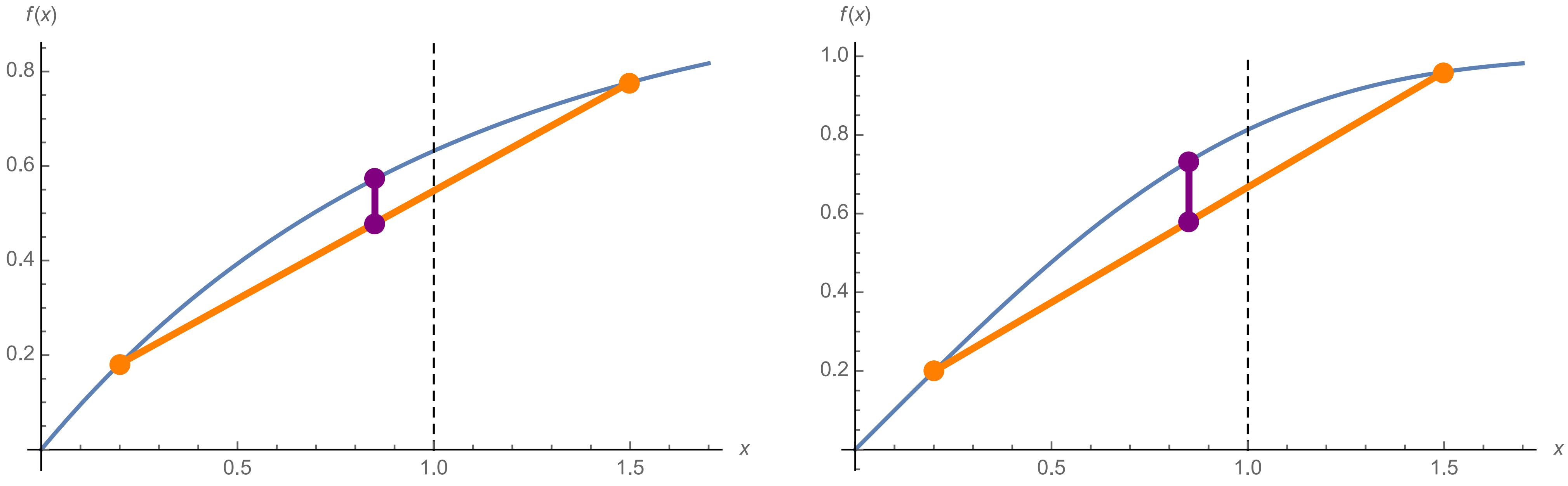




$$\mathrm{E}[\, f(y_u)\,] \ll f(\mathrm{E}[\, y_u\,]) \leq \mathrm{E}[y_u]$$

Main Difficulty

Same issue for other two functions.



We must bound the **spread** of y_u !

Introduction

Our framework

Algorithms / Analysis

IID arrival

Non-IID arrival

IID Arrival: Bounding Variance

Bound the **spread** of a random variable

→

Bound the **variance** of a random variable

IID Arrival: Bounding Variance

Bound the **spread** of a random variable

Bound the **variance** of a random variable

Lemma.

For any random variable y_{u} with variance σ and any (concave) function f, there exists a constant $\mu(\sigma, f)$ satisfying $\mathrm{E}[\, f(y_u)\,] \geq \mu(\sigma, f) \cdot \mathrm{E}[\, y_u \,]$

IID Arrival: Tradeoff

Our Goal.

Design Unbiased Estimators $y_u = \sum_j y_{u,j}$ with minimal variance.

IID Arrival: Estimators

Our Goal.

Design Unbiased Estimators $y_u = \sum_j y_{u,j}$ with minimal variance.

Independent Estimators. $\mathrm{y}_{u,j}(t_j) = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$

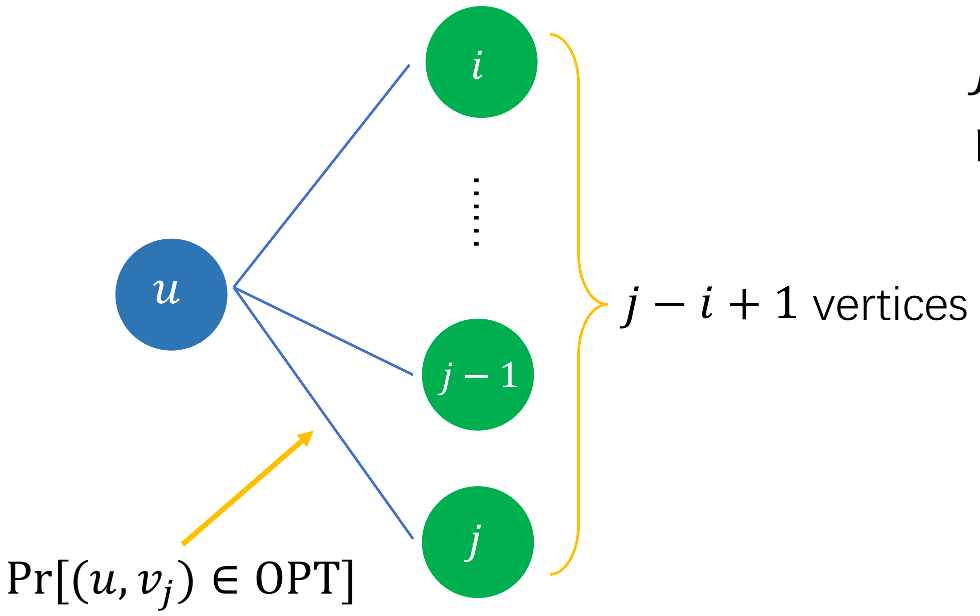
IID Arrival: Estimators

Our Goal.
Design Unbiased Estimators $y_u = \sum_j y_{u,j}$ with minimal variance.

Independent Estimators. $\mathrm{y}_{u,j}(t_j) = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$

Fully-Correlated Estimators. $\mathrm{y}_{u,j}\left(t_1, t_2, \ldots, t_j\right) = \Pr[(u,j) \in \mathrm{OPT} \mid t_1, t_2, \ldots, t_j]$

IID Arrival: Windowed Estimators



Windowed Estimators. Fix the types of the last $j - i + 1$ arrived vertices and resample the remaining types.

$$y_{u,j}^{[i]}(t_i, t_{i+1}, \dots, t_j) = \Pr[(u, j) \in \mathrm{OPT} \mid t_i, t_{i+1}, \dots, t_j]$$

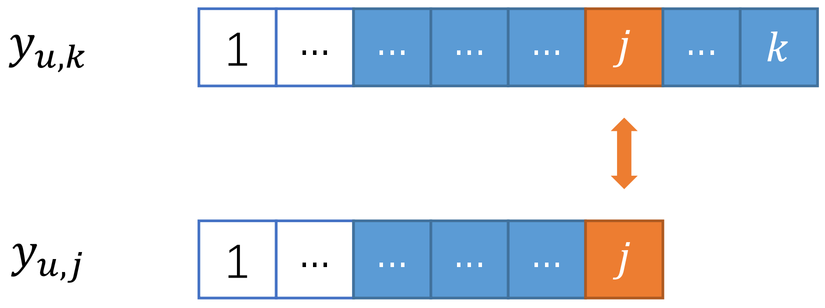
IID Arrival: Tradeoff

$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\leq j,k}}\left[y_{u,j} \cdot y_{u,k}\right]$$

IID Arrival: Tradeoff

$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\le j,k}}\left[y_{u,j} \cdot y_{u,k}\right]$$

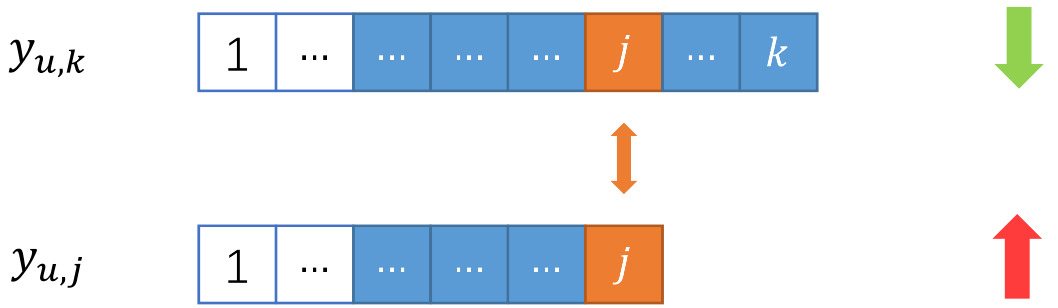
$y_{u,k}$ conditions on t_j



IID Arrival: Tradeoff

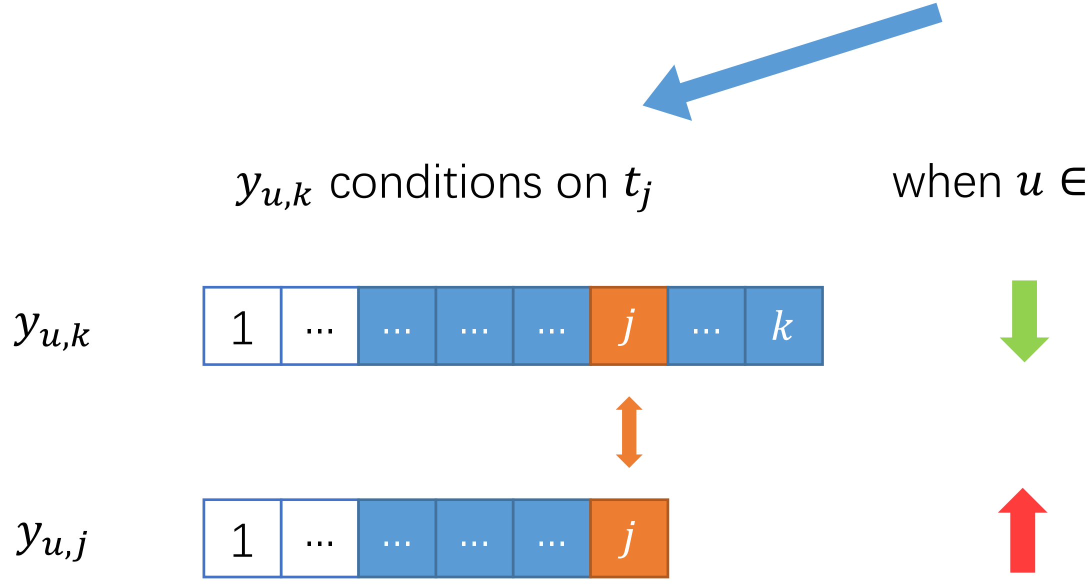
$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\leq j,k}}[y_{u,j} \cdot y_{u,k}]$$

$y_{u,k}$ conditions on t_j when $u \in t_j$



IID Arrival: Tradeoff

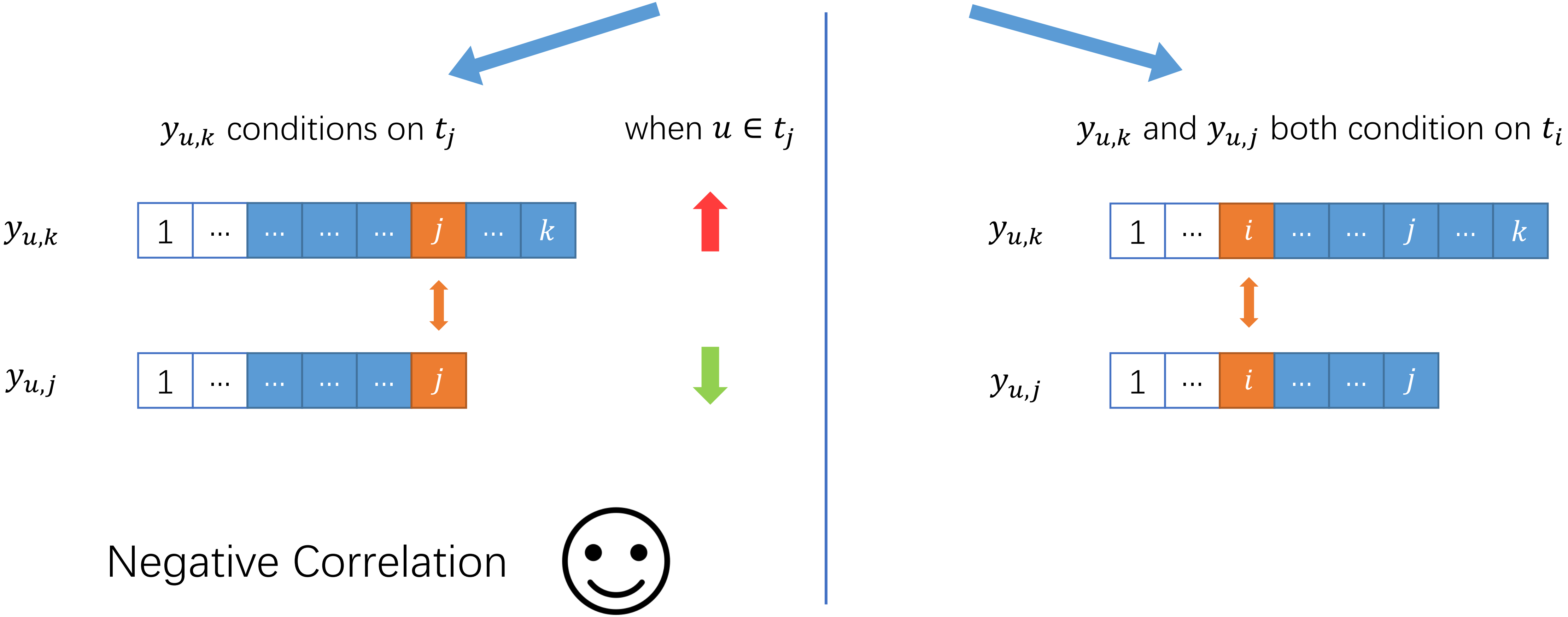
$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\leq j,k}}[y_{u,j} \cdot y_{u,k}]$$



Negative Correlation

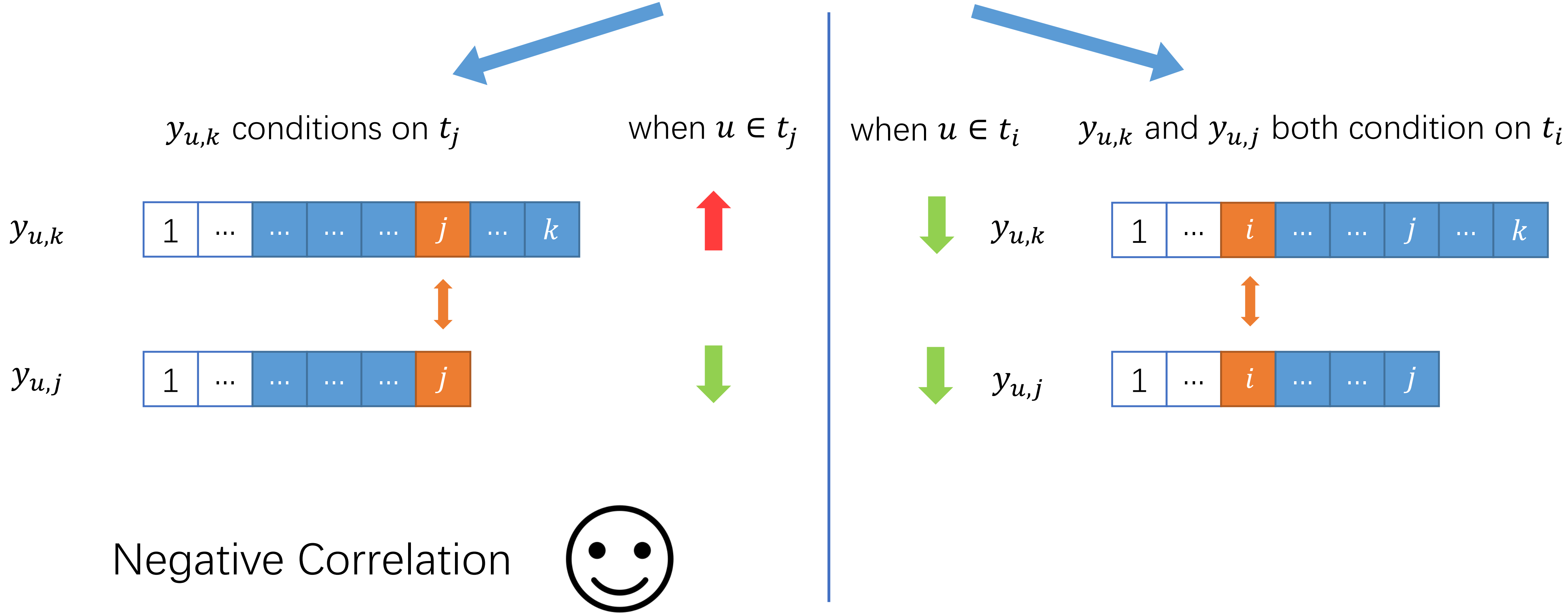
IID Arrival: Tradeoff

$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\le j,k}}[y_{u,j} \cdot y_{u,k}]$$



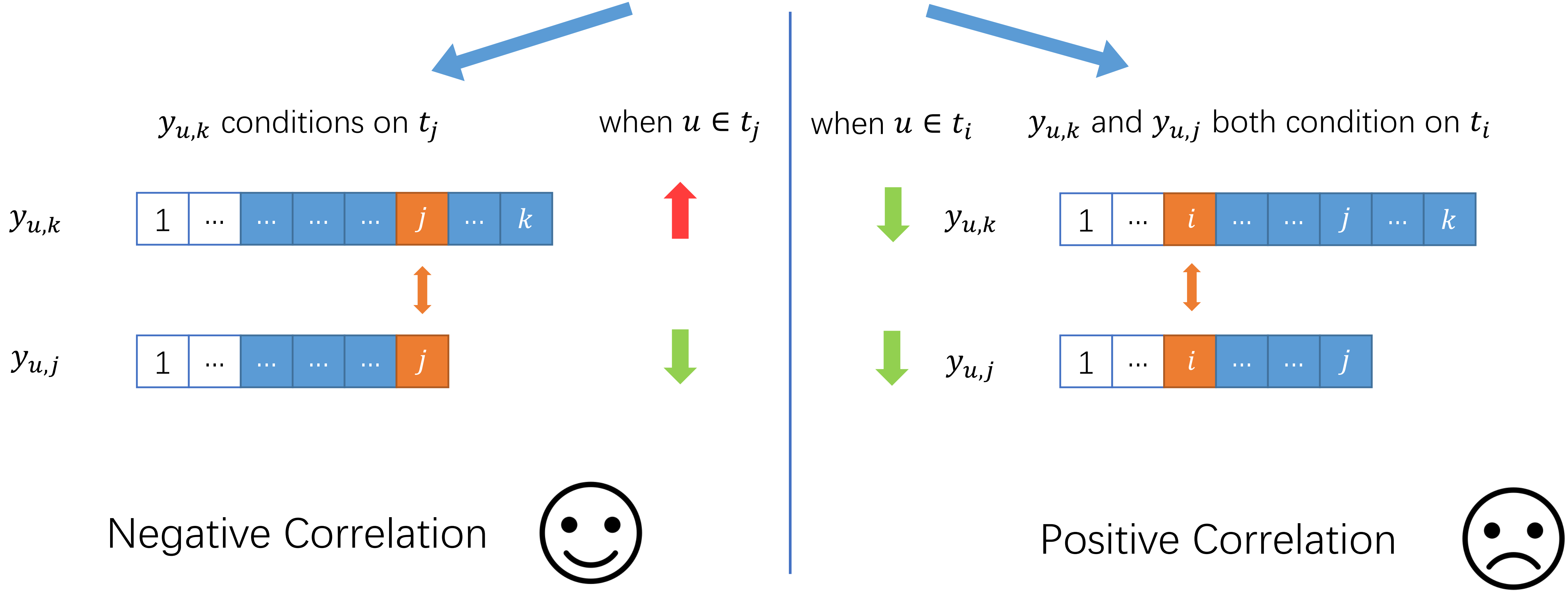
IID Arrival: Tradeoff

$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\leq j,k}}[y_{u,j} \cdot y_{u,k}]$$

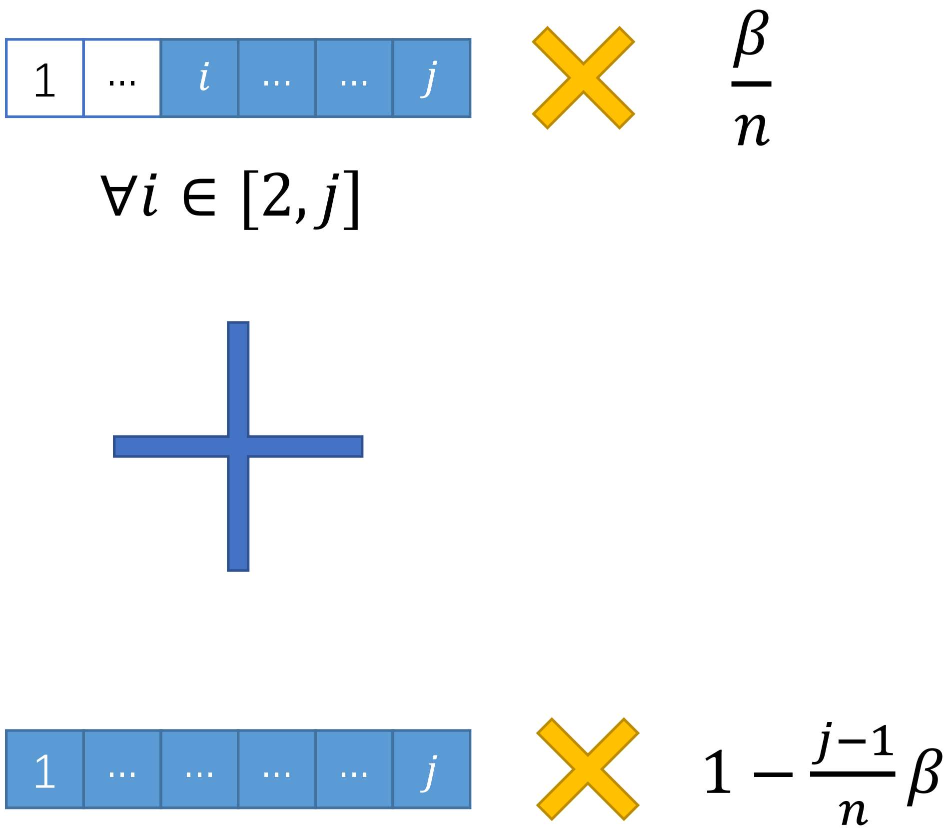


IID Arrival: Tradeoff

$$\mathrm{E}_{\mathbf{t}}[y_u^2] = \sum_{j,k} \mathrm{E}_{\mathbf{t}_{\leq j,k}} [y_{u,j} \cdot y_{u,k}]$$



IID Arrival: Mixing of Windowed Estimators



Our Estimators. The estimators we used is a linear combination of windowed estimators:

$$y_{u,j} = \frac{\beta}{n}\sum_{i=2}^{j} y_{u,j}^{[i]} + \left(1 - \frac{j-1}{n}\beta\right) y_{u,j}^{[1]}$$

where $\beta = 0.79$ is a optimized constant.

Introduction

Our framework

Algorithms / Analysis

IID arrival

Non-IID arrival

Non-IID: Independent Estimator is Optimal

Independent Estimators. $y_{u,j} = \Pr[(u,j) \in \mathrm{OPT} \mid t_j]$.

Proof Sketch.

1. For any fixed mean $\mathrm{E}[\, y_u \,]$, we characterize the **worst-case** distribution that minimizes $\mathrm{E}[\, f(y_u) \,]$.

2. Any unbiased estimator has the same performance under the worst-case distribution.

Thanks!