Scalable Auction Algorithms for Bipartite Maximum Matching Problems

Quanquan C. Liu

Joint work with

Yiduo Ke

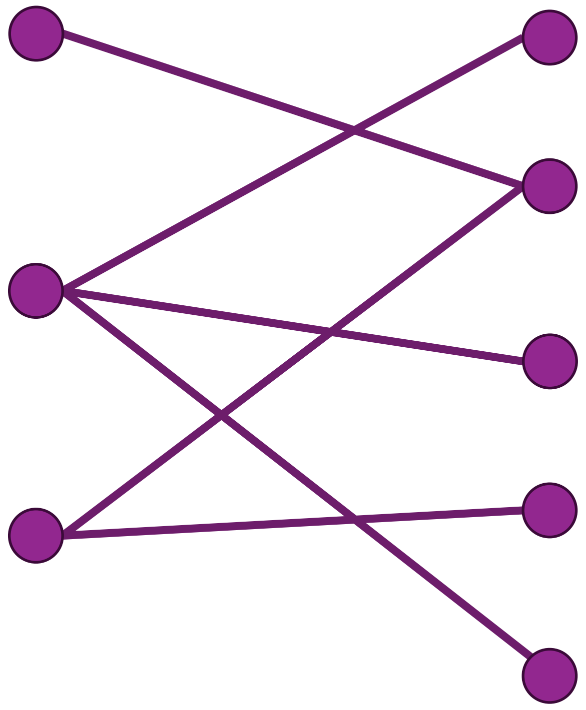
Samir Khuller

Northwestern
Computer Science

SIMONS
INSTITUTE
for the Theory of Computing

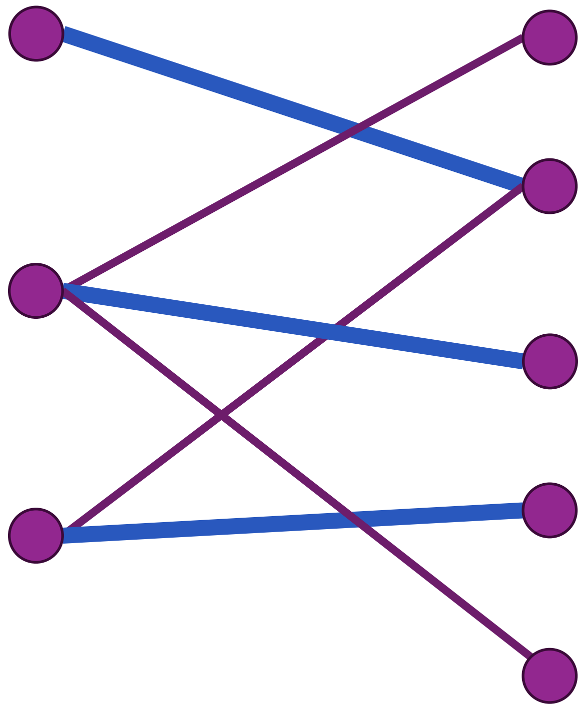
Bipartite Matching Problems

- **Maximum Matching**: return matching of maximum size



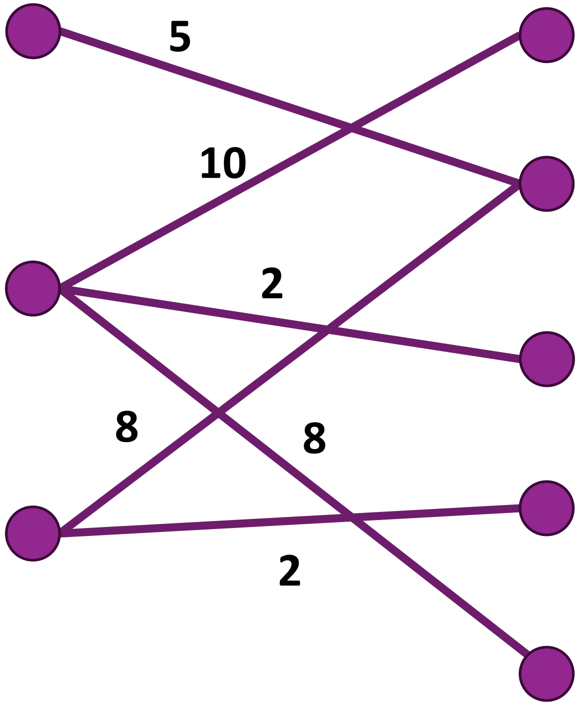
Bipartite Matching Problems

- **Maximum Matching**: return matching of maximum size



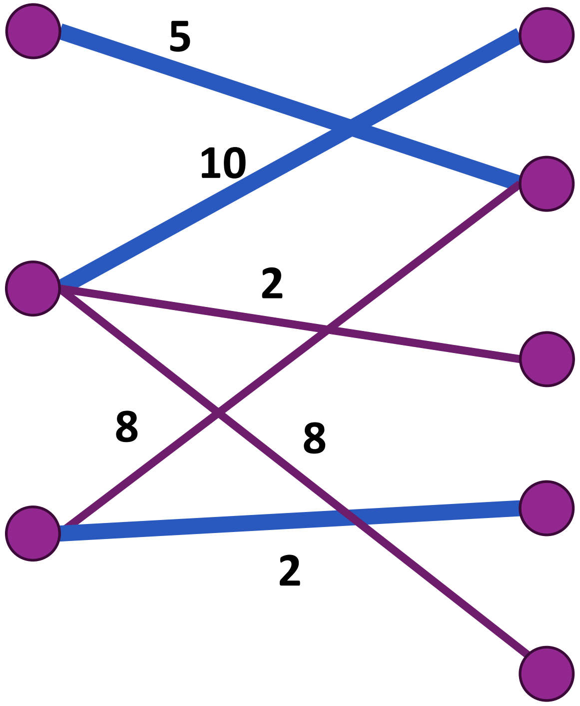
Bipartite Matching Problems

- **Maximum** **Weighted** **Matching**: return matching of maximum weight



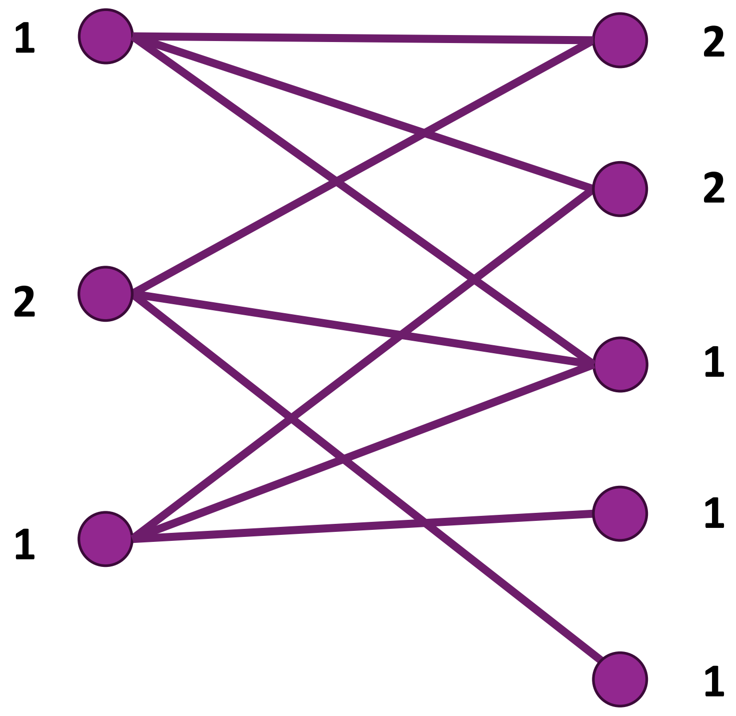
Bipartite Matching Problems

- **Maximum** **Weighted** **Matching**: return matching of maximum weight



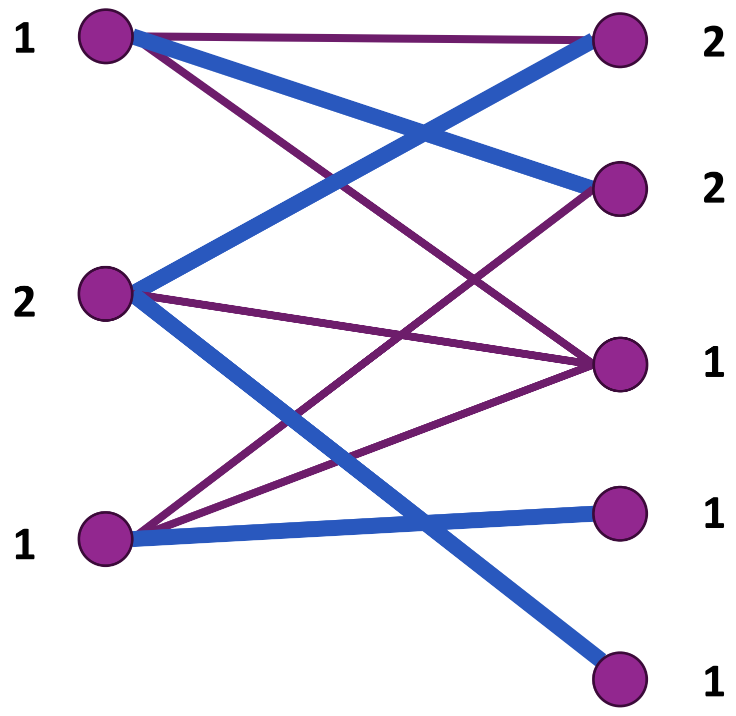
Bipartite Matching Problems

- **Maximum** ***b*-Matching**: return matching of maximum size when each node v can be matched to at most b_v nodes



Bipartite Matching Problems

- **Maximum *b*-Matching**: return matching of maximum size when each node v can be matched to at most b_v nodes



Approximate Bipartite Matching Problems

- Let M^* be the optimum matching solution
- A **$(1-\varepsilon)$-approximate** solution $\widehat{M}$ has value at least:

$$\widehat{M} \geq (1-\varepsilon) \cdot M^*$$

Auction-Based Maximum Matching

- Introduced by Demange, Gale, and Sotomayor '86 and Bertsekas '81 for **exact** (weighted) matching
  - Same runtime as Hungarian method and maxflow

Auction-Based Maximum Matching

- Introduced by Demange, Gale, and Sotomayor ‘86 and Bertsekas ‘81 for **exact** (weighted) matching
  - Same runtime as Hungarian method and maxflow
- Dobzinski, Nisan, and Oren ‘14 extend to **approximation** and **blackboard distributed setting**

Auction-Based Maximum Matching

- Introduced by Demange, Gale, and Sotomayor ‘86 and Bertsekas ‘81 for **exact** (weighted) matching
  - Same runtime as Hungarian method and maxflow
- Dobzinski, Nisan, and Oren ‘14 extend to **approximation** and **blackboard distributed setting**
  - $(1-\varepsilon)$**-approximation** in $O\left(\frac{\log n}{\varepsilon^2}\right)$ **rounds**

Auction-Based Maximum Matching

- Introduced by Demange, Gale, and Sotomayor ‘86 and Bertsekas ‘81 for **exact** (weighted) matching
  - Same runtime as Hungarian method and maxflow
- Dobzinski, Nisan, and Oren ‘14 extend to **approximation** and **blackboard distributed setting**
  - $(1-\varepsilon)$**-approximation** in $O\left(\frac{\log n}{\varepsilon^2}\right)$ **rounds**
- Assadi, Liu, and Tarjan ‘21 extend to **semi-streaming** and **MPC**

Auction-Based Maximum Matching

- Introduced by Demange, Gale, and Sotomayor ‘86 and Bertsekas ‘81 for **exact** (weighted) matching
  - Same runtime as Hungarian method and maxflow
- Dobzinski, Nisan, and Oren ‘14 extend to **approximation** and **blackboard distributed setting**
  - $(1-\varepsilon)$**-approximation** in $O\left(\frac{\log n}{\varepsilon^2}\right)$ **rounds**
- Assadi, Liu, and Tarjan ‘21 extend to **semi-streaming** and **MPC**
  - $O\left(\frac{1}{\varepsilon^2}\right)$ **passes** in streaming, $O\left(n \log\left(\frac{1}{\varepsilon}\right)\right)$ **space**
  - $O\left(\frac{1}{\varepsilon^2} \cdot \log\log n\right)$**-round,** $O(n)$**-memory** algorithm in MPC

Auction-Based Maximum

Zheng and Henzinger '23 extends MWM to **sequential and dynamic models**

- Introduced by Demange, Gale, and Sotomayor '86 and Bertsekas '81 for **exact** (weighted) matching
  - Same runtime as Hungarian method and maxflow
- Dobzinski, Nisan, and Oren '14 extend to **approximation** and **blackboard distributed setting**
  - $(1-\varepsilon)$**-approximation** in $O\left(\frac{\log n}{\varepsilon^2}\right)$ **rounds**
- Assadi, Liu, and Tarjan '21 extend to **semi-streaming** and **MPC**
  - $O\left(\frac{1}{\varepsilon^2}\right)$ **passes** in streaming, $O\left(n \log\left(\frac{1}{\varepsilon}\right)\right)$ **space**
  - $O\left(\frac{1}{\varepsilon^2} \cdot \log\log n\right)$**-round,** $O(n)$**-memory** algorithm in MPC

Our Results

MWM = Maximum Weighted Matching
MCBM = Maximum Cardinality *b*-Matching

Model		Previous Results		Our Results	
Blackboard Distributed	MWM	$\Omega(n \log n)$ (trivial)	[DNO14]	$O\left(\frac{n \log^3(n)}{\varepsilon^8}\right)$	Theorem 3.9
	MCBM	$\Omega(nb \log n)$	trivial	$O\left(\frac{nb \log^2 n}{\varepsilon^2}\right)$	Theorem 4.8
Streaming	MWM	$O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)$ pass $O\left(\frac{n \log n}{\varepsilon^2}\right)$ space	[AG11]	$O\left(\frac{1}{\varepsilon^8}\right)$ pass $O\left(n \cdot \log n \cdot \log(1/\varepsilon)\right)$ space	Theorem 3.11
	MCBM	$O(\log n/\varepsilon^3)$ pass $\widetilde{O}\left(\frac{\sum_{i \in L \cup R} b_i}{\varepsilon^3}\right)$ space	[AG18]	$O\left(\frac{1}{\varepsilon^2}\right)$ pass $O\left(\left(\sum_{i \in L} b_i + \lvert R \rvert\right) \log(1/\varepsilon)\right)$ space	Theorem 4.10
MPC	MWM	$O_\varepsilon(\log \log n)$ rounds $O_\varepsilon(n \operatorname{poly}(\log n))$ space p.m.	[GKMS19] (general)	$O\left(\frac{\log \log n}{\varepsilon^7}\right)$ rounds $O(n \cdot \log_{(1/\varepsilon)}(n))$ space p.m.	Theorem 3.15
Parallel	MWM	$O\left(m \cdot \operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ work* $O\left(\operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ depth*	[HS22] (general)	$O\left(\frac{m \log(n)}{\varepsilon^7}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^7}\right)$ depth	Theorem 3.13
	MCBM	N/A	N/A	$O\left(\frac{m \log n}{\varepsilon^2}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^2}\right)$ depth	Theorem 4.11

"Universal" solution across many different scalable models!

Our Results

MWM = Maximum Weighted Matching
MCBM = Maximum Cardinality *b*-Matching

First results in blackboard distributed

Model		Previous Results		Our Results	
Blackboard Distributed	MWM	$\Omega(n \log n)$ (trivial)	[DNO14]	$O\left(\frac{n \log^3(n)}{\varepsilon^8}\right)$	Theorem 3.9
	MCBM	$\Omega(nb \log n)$	trivial	$O\left(\frac{nb \log^2 n}{\varepsilon^2}\right)$	Theorem 4.8
Streaming	MWM	$O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)$ pass $O\left(\frac{n \log n}{\varepsilon^2}\right)$ space	[AG11]	$O\left(\frac{1}{\varepsilon^8}\right)$ pass $O\left(n \cdot \log n \cdot \log(1/\varepsilon)\right)$ space	Theorem 3.11
	MCBM	$O(\log n/\varepsilon^3)$ pass $\widetilde{O}\left(\frac{\sum_{i \in L \cup R} b_i}{\varepsilon^3}\right)$ space	[AG18]	$O\left(\frac{1}{\varepsilon^2}\right)$ pass $O\left(\left(\sum_{i \in L} b_i + \lvert R \rvert\right) \log(1/\varepsilon)\right)$ space	Theorem 4.10
MPC	MWM	$O_\varepsilon(\log \log n)$ rounds $O_\varepsilon(n\,\mathrm{poly}(\log n))$ space p.m.	[GKMS19] (general)	$O\left(\frac{\log \log n}{\varepsilon^7}\right)$ rounds $O(n \cdot \log_{(1/\varepsilon)}(n))$ space p.m.	Theorem 3.15
Parallel	MWM	$O\left(m \cdot \mathrm{poly}\left(1/\varepsilon, \log n\right)\right)$ work* $O\left(\mathrm{poly}\left(1/\varepsilon, \log n\right)\right)$ depth*	[HS22] (general)	$O\left(\frac{m \log(n)}{\varepsilon^7}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^7}\right)$ depth	Theorem 3.13
	MCBM	N/A	N/A	$O\left(\frac{m \log n}{\varepsilon^2}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^2}\right)$ depth	Theorem 4.11

Our Results

MWM = Maximum Weighted Matching
MCBM = Maximum Cardinality *b*-Matching

Model		Previous Results		Our Results	
Blackboard Distributed	MWM	$\Omega(n \log n)$ (trivial)	[DNO14]	$O\left(\frac{n \log^3(n)}{\varepsilon^8}\right)$	Theorem 3.9
	MCBM	$\Omega(nb \log n)$	trivial	$O\left(\frac{nb \log^2 n}{\varepsilon^2}\right)$	Theorem 4.8
Streaming	MWM	$O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)$ pass $O\left(\frac{n \log n}{\varepsilon^2}\right)$ space	[AG11]	$O\left(\frac{1}{\varepsilon^8}\right)$ pass $O\left(n \cdot \log n \cdot \log(1/\varepsilon)\right)$ space	Theorem 3.11
	MCBM	$O(\log n/\varepsilon^3)$ pass $\widetilde{O}\left(\frac{\sum_{i \in L \cup R} b_i}{\varepsilon^3}\right)$ space	[AG18]	$O\left(\frac{1}{\varepsilon^2}\right)$ pass $O\left(\left(\sum_{i \in L} b_i + \lvert R \rvert\right) \log(1/\varepsilon)\right)$ space	Theorem 4.10
MPC	MWM	$O_\varepsilon(\log \log n)$ rounds $O_\varepsilon(n \operatorname{poly}(\log n))$ space p.m.	[GKMS19] (general)	$O\left(\frac{\log \log n}{\varepsilon^7}\right)$ rounds $O(n \cdot \log_{(1/\varepsilon)}(n))$ space p.m.	Theorem 3.15
Parallel	MWM	$O\left(m \cdot \operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ work* $O\left(\operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ depth*	[HS22] (general)	$O\left(\frac{m \log(n)}{\varepsilon^7}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^7}\right)$ depth	Theorem 3.13
	MCBM	N/A	N/A	$O\left(\frac{m \log n}{\varepsilon^2}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^2}\right)$ depth	Theorem 4.11

Eliminate polynomial dependence in $\left(\frac{1}{\varepsilon}\right)$ in space

Our Results

MWM = Maximum Weighted Matching
MCBM = Maximum Cardinality b-Matching

Model		Previous Results		Our Results	
Blackboard Distributed	MWM	$\Omega(n \log n)$ (trivial)	[DNO14]	$O\left(\frac{n \log^3(n)}{\varepsilon^8}\right)$	Theorem 3.9
	MCBM	$\Omega(nb \log n)$	trivial	$O\left(\frac{nb \log^2 n}{\varepsilon^2}\right)$	Theorem 4.8
Streaming	MWM	$O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)$ pass $O\left(\frac{n \log n}{\varepsilon^2}\right)$ space	[AG11]	$O\left(\frac{1}{\varepsilon^8}\right)$ pass $O\left(n \cdot \log n \cdot \log(1/\varepsilon)\right)$ space	Theorem 3.11
	MCBM	$O(\log n/\varepsilon^3)$ pass $\widetilde{O}\left(\frac{\sum_{i \in L \cup R} b_i}{\varepsilon^3}\right)$ space	[AG18]	$O\left(\frac{1}{\varepsilon^2}\right)$ pass $O\left(\left(\sum_{i \in L} b_i + \lvert R \rvert\right) \log(1/\varepsilon)\right)$ space	Theorem 4.10
MPC	MWM	$O_\varepsilon(\log \log n)$ rounds $O_\varepsilon(n \operatorname{poly}(\log n))$ space p.m.	[GKMS19] (general)	$O\left(\frac{\log \log n}{\varepsilon^7}\right)$ rounds $O(n \cdot \log_{(1/\varepsilon)}(n))$ space p.m.	Theorem 3.15
Parallel	MWM	$O\left(m \cdot \operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ work* $O\left(\operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ depth*	[HS22] (general)	$O\left(\frac{m \log(n)}{\varepsilon^7}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^7}\right)$ depth	Theorem 3.13
	MCBM	N/A	N/A	$O\left(\frac{m \log n}{\varepsilon^2}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^2}\right)$ depth	Theorem 4.11

Eliminate exponential dependence on $\left(\frac{1}{\varepsilon}\right)$

Our Results

MWM = Maximum Weighted Matching
MCBM = Maximum Cardinality *b*-Matching

Model		Previous Results		Our Results	
Blackboard Distributed	MWM	$\Omega(n \log n)$ (trivial)	[DNO14]	$O\left(\frac{n \log^3(n)}{\varepsilon^8}\right)$	Theorem 3.9
	MCBM	$\Omega(nb \log n)$	trivial	$O\left(\frac{nb \log^2 n}{\varepsilon^2}\right)$	Theorem 4.8
Streaming	MWM	$O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)$ pass $O\left(\frac{n \log n}{\varepsilon^2}\right)$ space	[AG11]	$O\left(\frac{1}{\varepsilon^8}\right)$ pass $O\left(n \cdot \log n \cdot \log(1/\varepsilon)\right)$ space	Theorem 3.11
	MCBM	$O(\log n/\varepsilon^3)$ pass $\widetilde{O}\left(\frac{\sum_{i \in L \cup R} b_i}{\varepsilon^3}\right)$ space	[AG18]	$O\left(\frac{1}{\varepsilon^2}\right)$ pass $O\left(\left(\sum_{i \in L} b_i + \lvert R \rvert\right) \log(1/\varepsilon)\right)$ space	Theorem 4.10
MPC	MWM	$O_\varepsilon(\log \log n)$ rounds $O_\varepsilon(n \operatorname{poly}(\log n))$ space p.m.	[GKMS19] (general)	$O\left(\frac{\log \log n}{\varepsilon^7}\right)$ rounds $O(n \cdot \log_{(1/\varepsilon)}(n))$ space p.m.	Theorem 3.15
Parallel	MWM	$O\left(m \cdot \operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ work* $O\left(\operatorname{poly}\left(1/\varepsilon, \log n\right)\right)$ depth*	[HS22] (general)	$O\left(\frac{m \log(n)}{\varepsilon^7}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^7}\right)$ depth	Theorem 3.13
	MCBM	N/A	N/A	$O\left(\frac{m \log n}{\varepsilon^2}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^2}\right)$ depth	Theorem 4.11

Eliminate large dependence on $\left(\frac{1}{\varepsilon}\right)$ and $\log n$

Outline

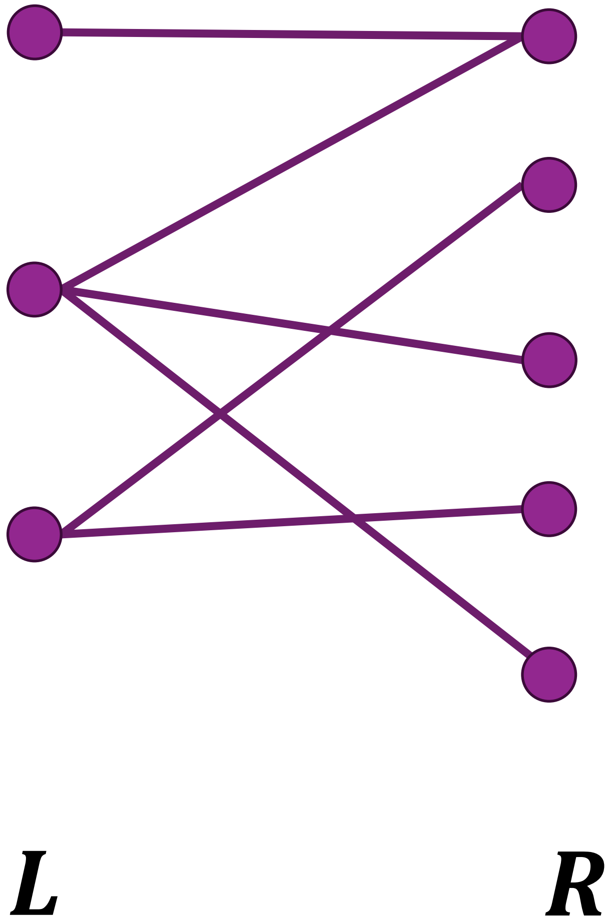
- Auction Algorithm of [ALT21] for maximum cardinality matching
- Our auction algorithm for maximum weighted matching
  - Algorithm description
  - Minimizing dependence on $\log(W)$
- Our auction algorithm for maximum *b*-matching

Outline

- Auction Algorithm of [ALT21] for maximum cardinality matching
- Our auction algorithm for maximum weighted matching
  - Algorithm description
  - Minimizing dependence on $\log(W)$
- Our auction algorithm for maximum *b*-matching

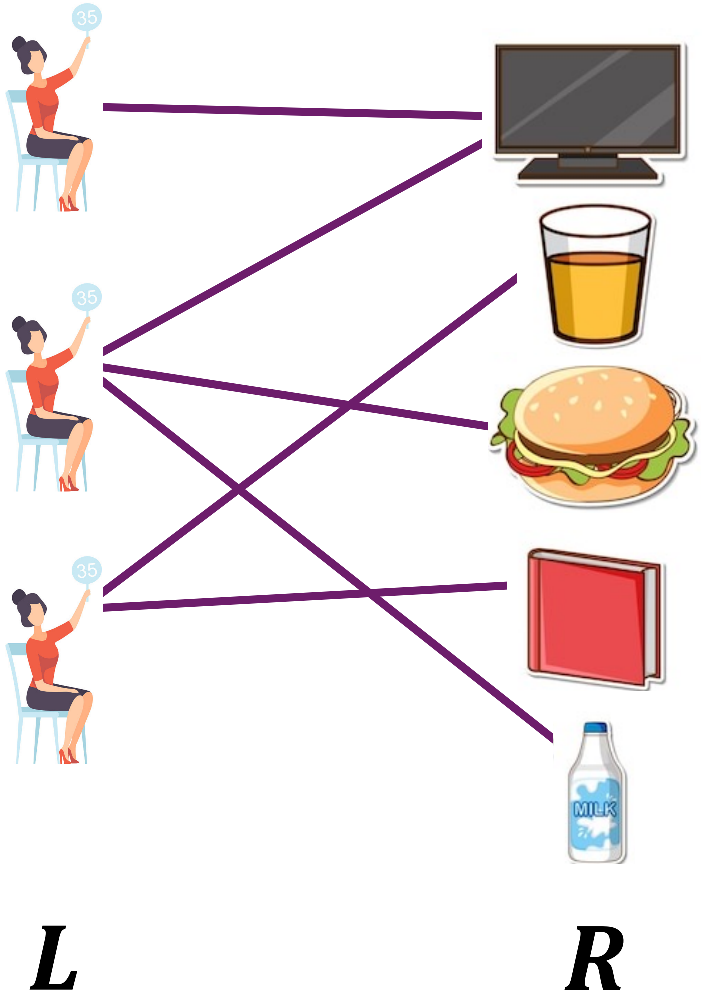
Auction Algorithm of [ALT21]

Left and Right Side of Bipartite Graph



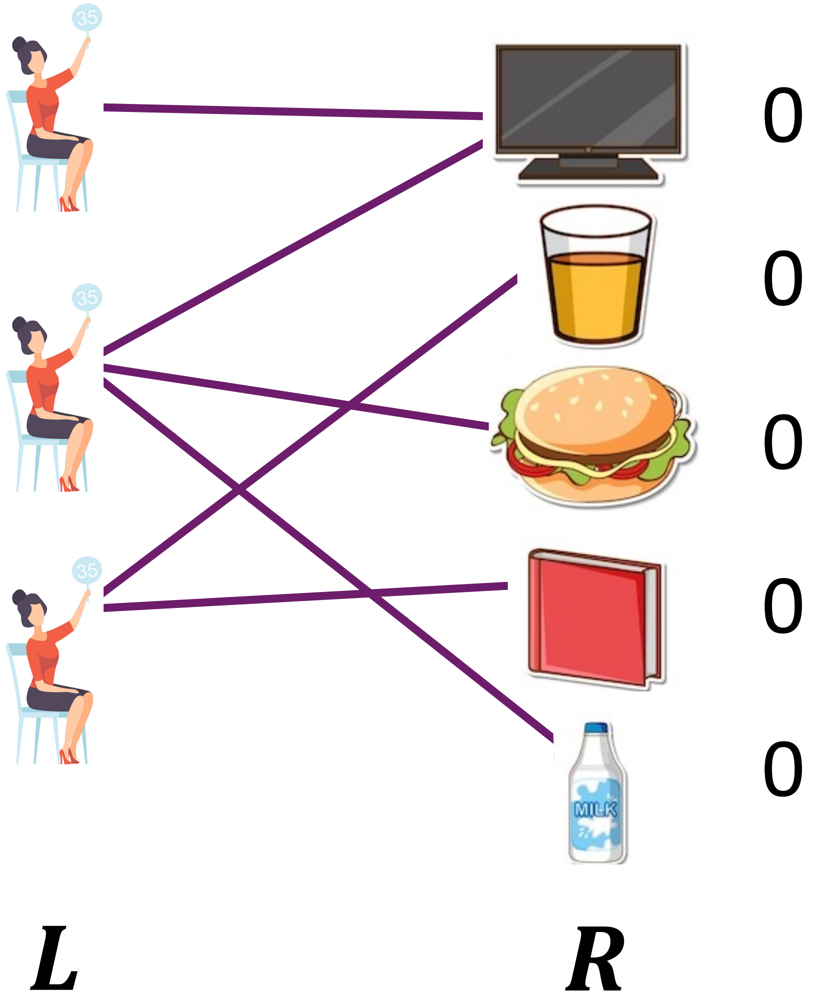
Auction Algorithm of [ALT21]

Left and Right Side of Bipartite Graph



Left Side has **Bidders** and **Right Side** has **Items**

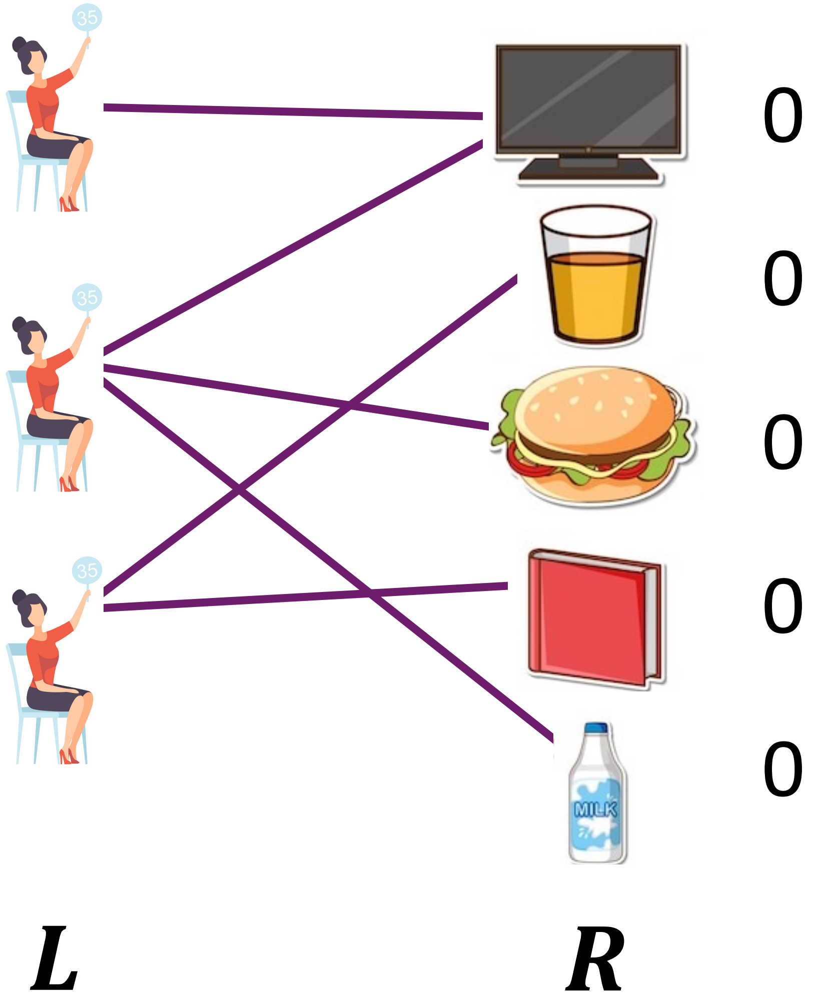
Auction Algorithm of [ALT21]



All items **start with price 0**

Auction Algorithm of [ALT21]

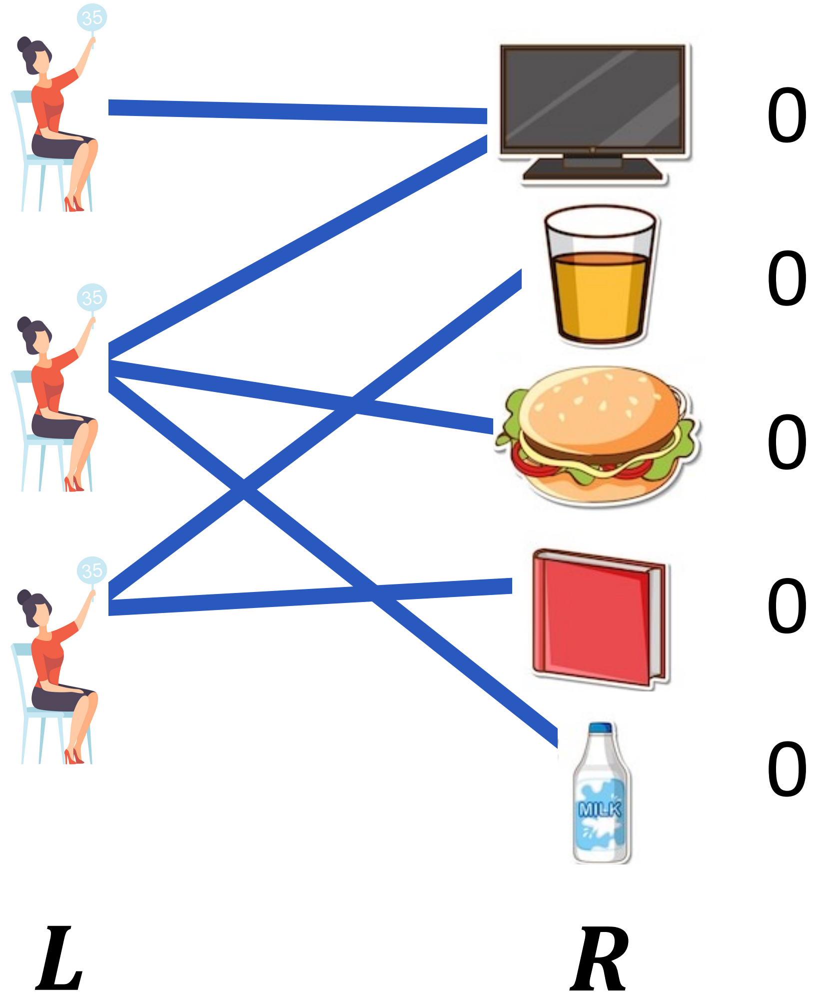
Iteratively, bidders **bid** on all **lowest price adjacent items**



All items **start with price 0**

Auction Algorithm of [ALT21]

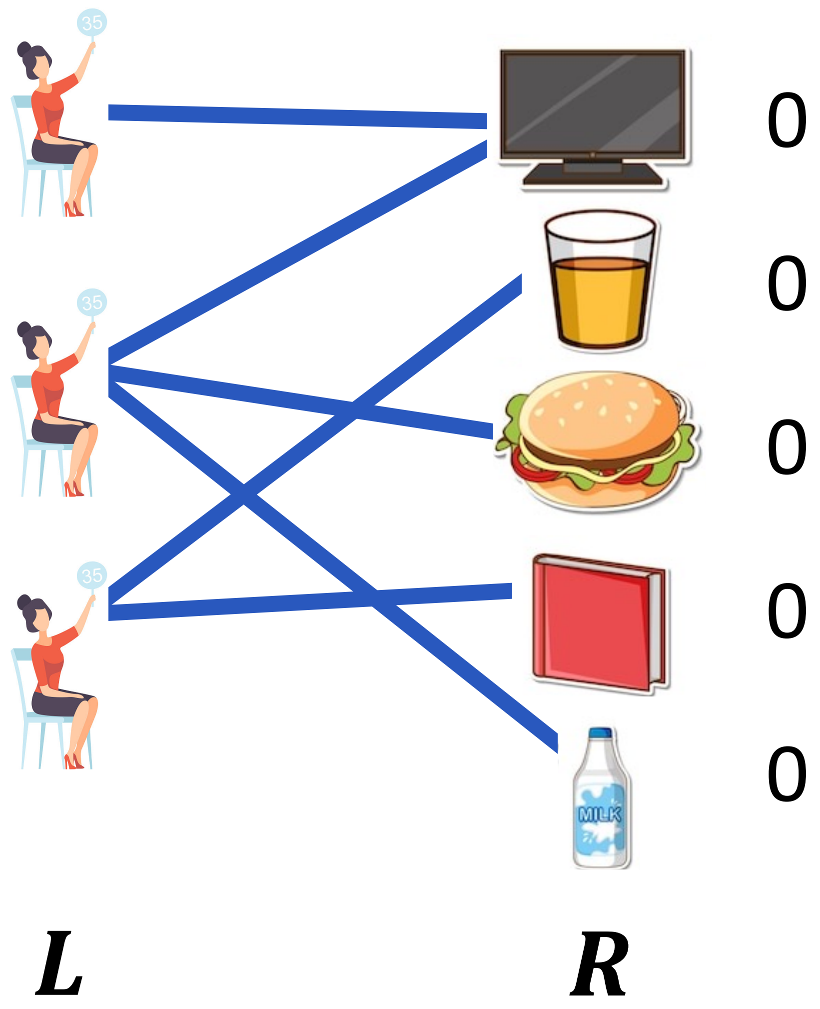
Iteratively, bidders **bid** on all **lowest price adjacent items**



All items **start with price 0**

Auction Algorithm of [ALT21]

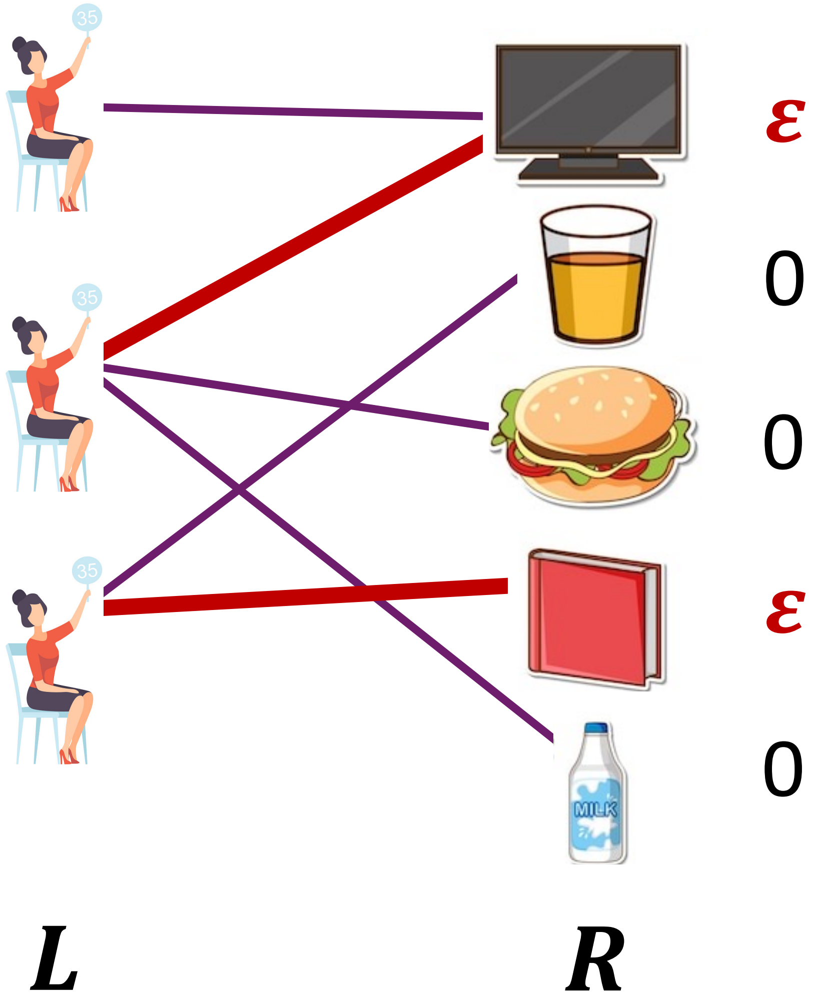
Iteratively, bidders **bid** on all **lowest price adjacent items**



Find **maximal matching** among **induced subgraph of bid items**

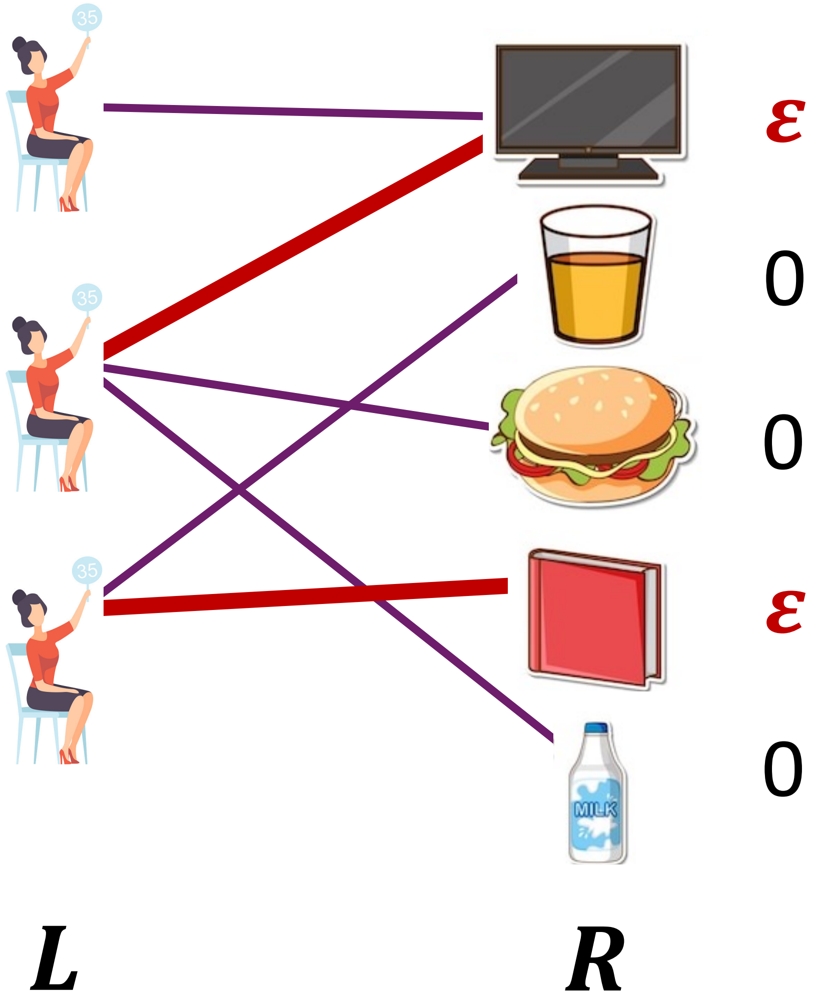
Auction Algorithm of [ALT21]

Increase price of items in matching **by ε** and **maintain current matching**



Auction Algorithm of [ALT21]

Increase price of items in matching **by ε** and **maintain current matching**

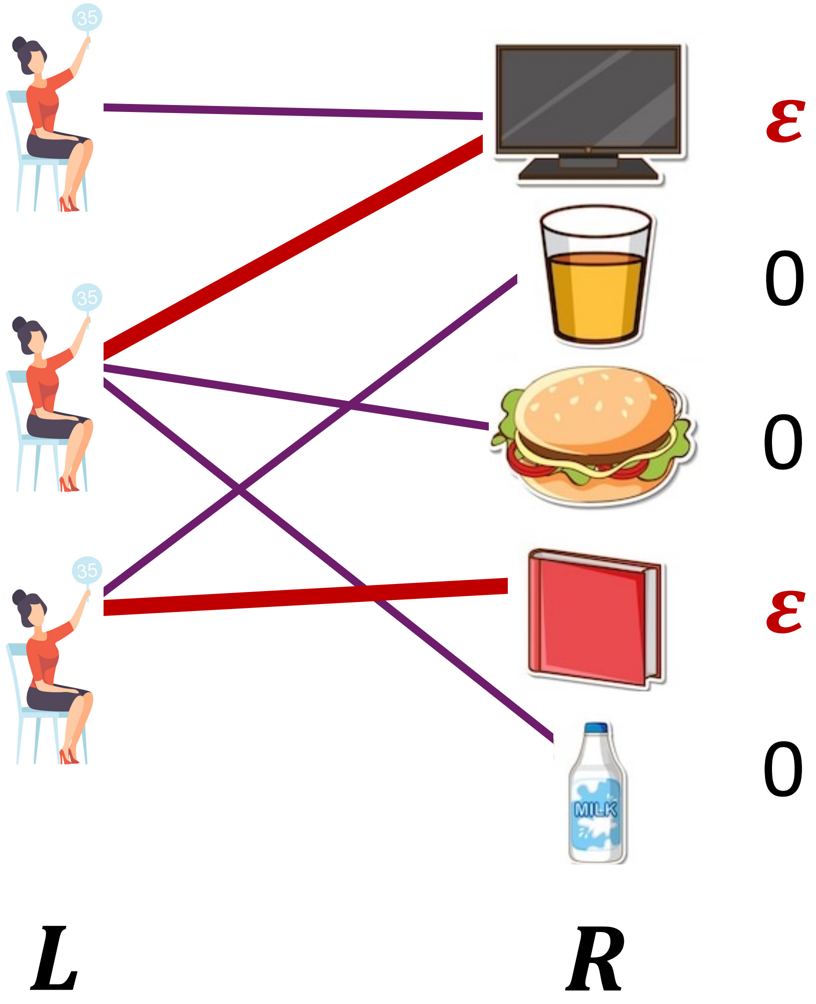


Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

Iterate for

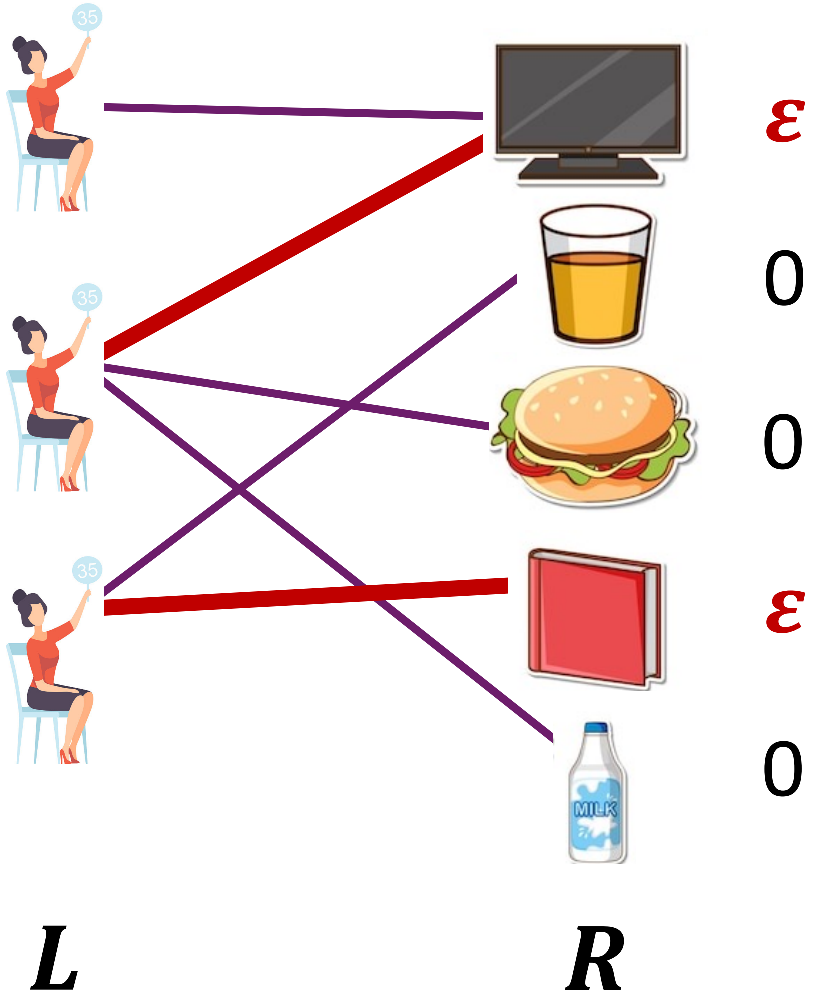
$\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**



Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

Iterate for $\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**

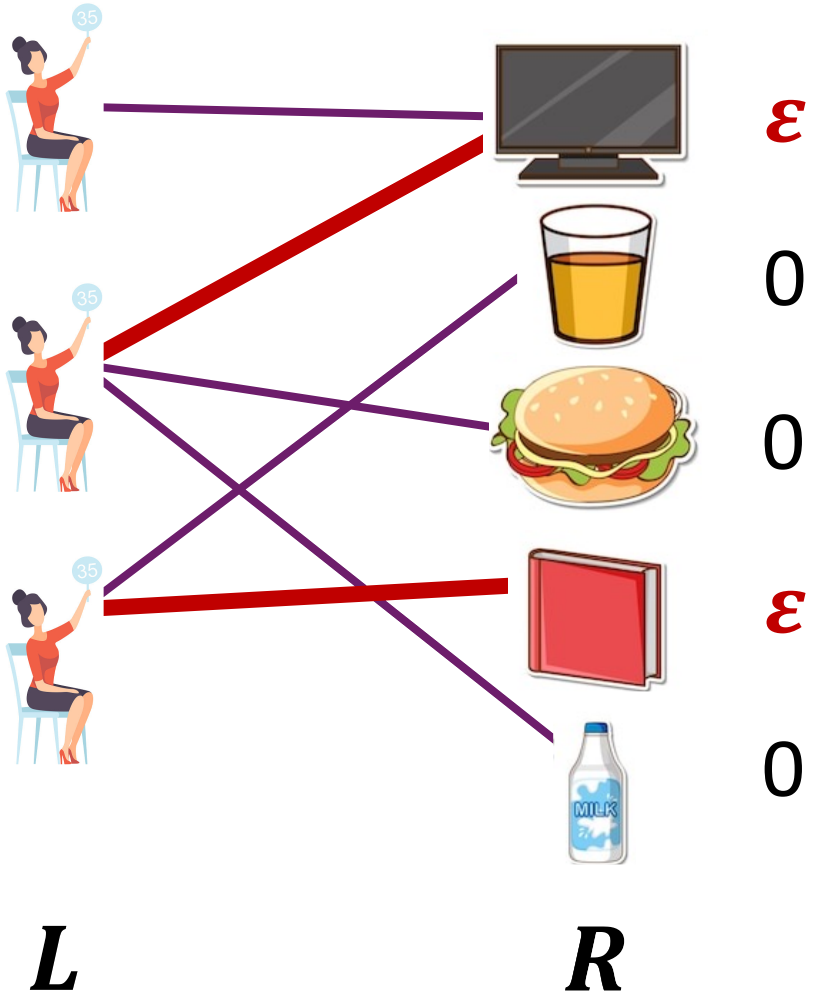


Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

Iterate for $\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**

Each **unmatched** bidder bids



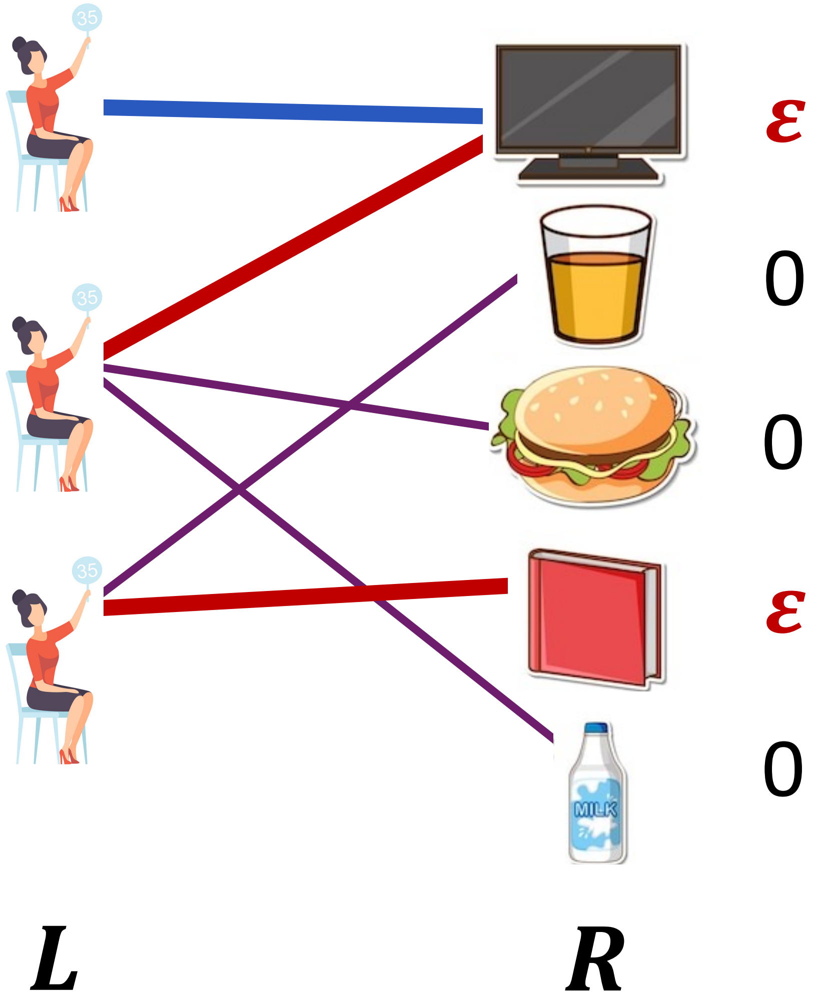
Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

Iterate for

$\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**

Each **unmatched** bidder bids



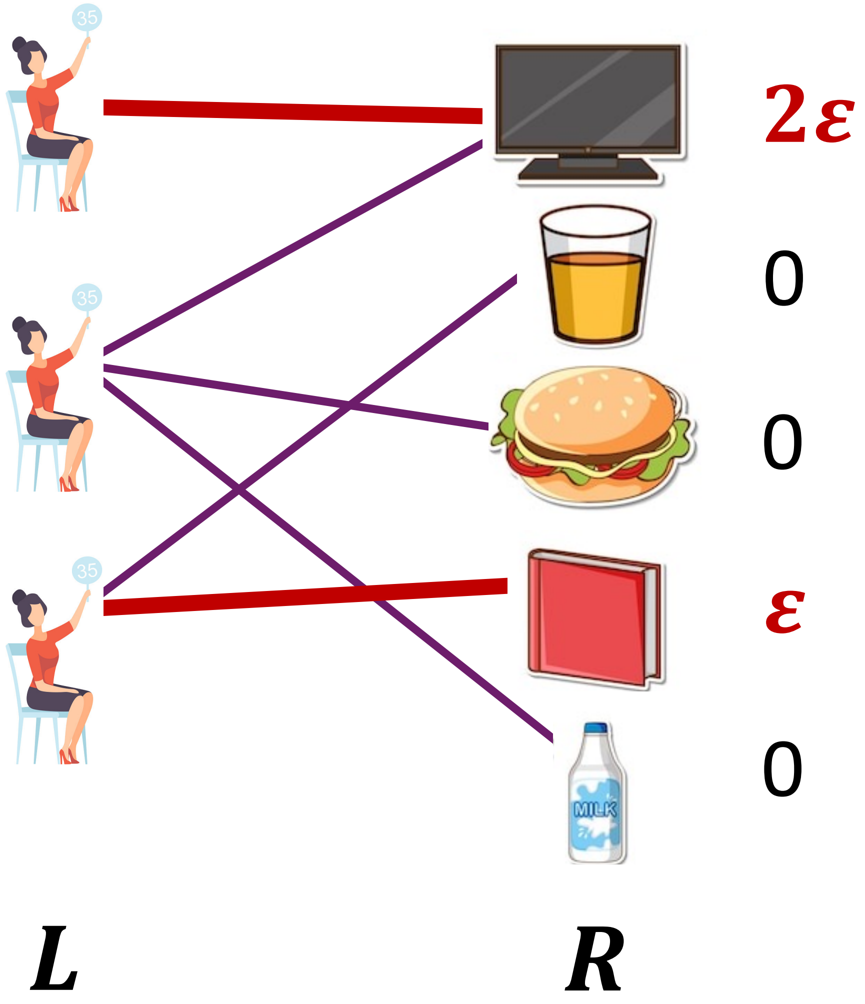
Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

Iterate for

$\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**

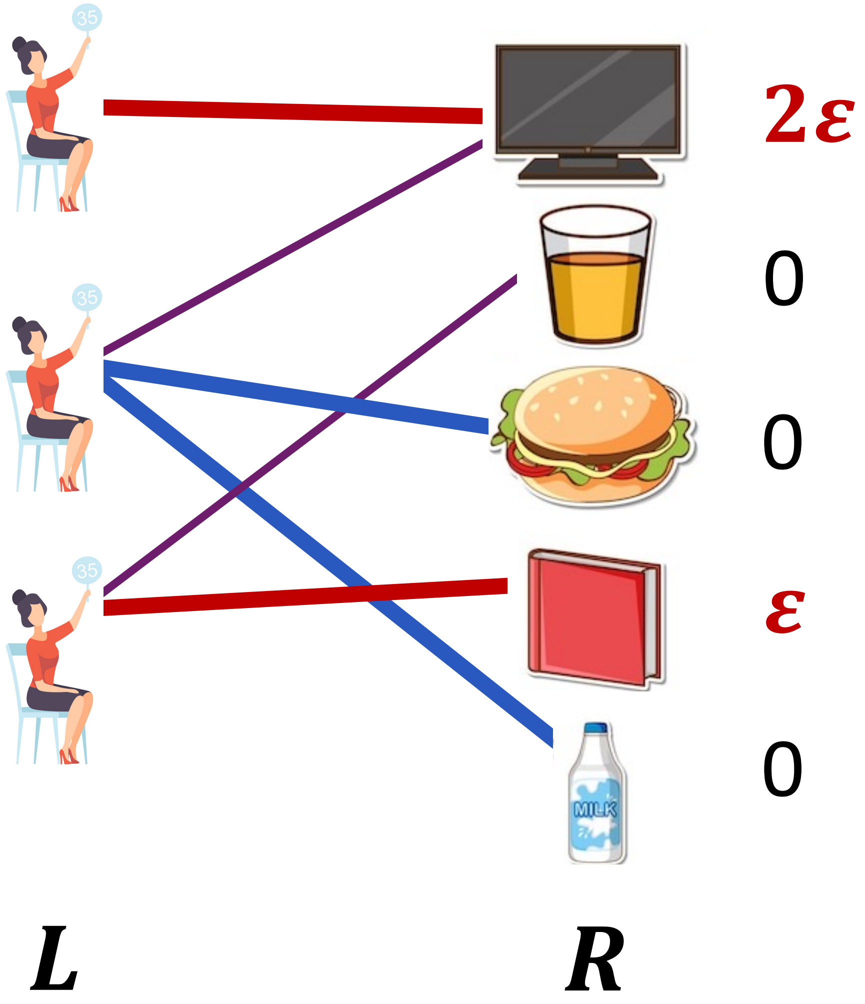
Item goes to new bidder!



Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

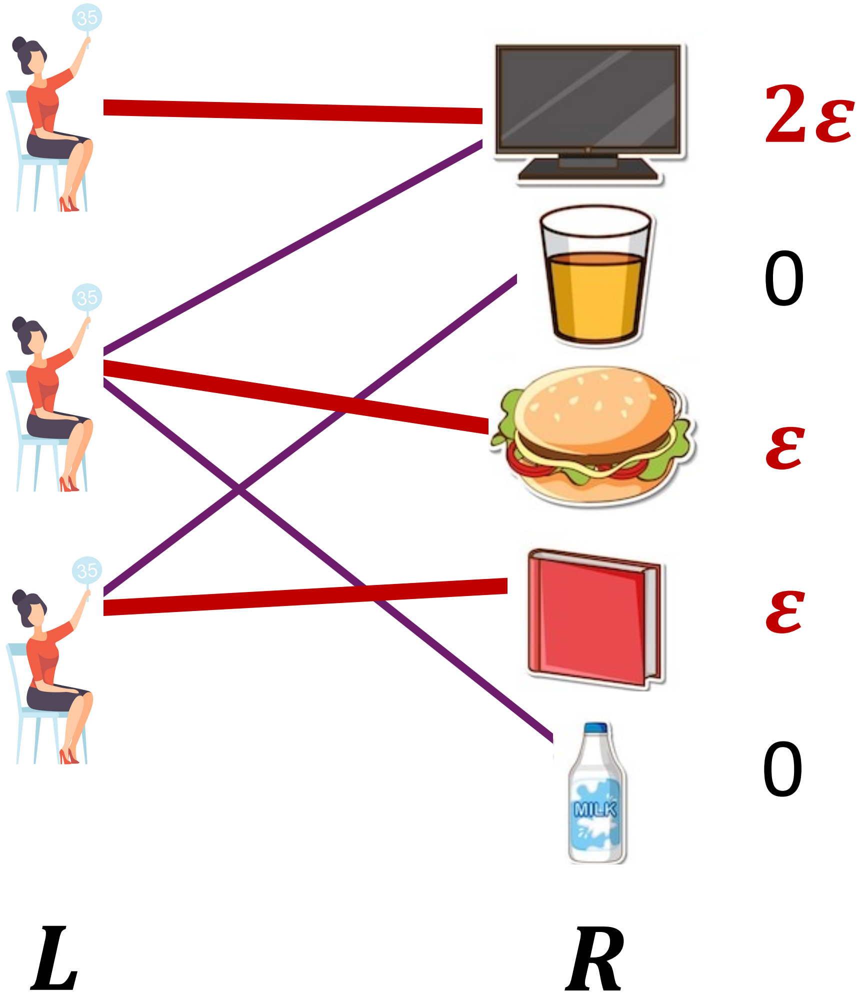
Iterate for $\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**



Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

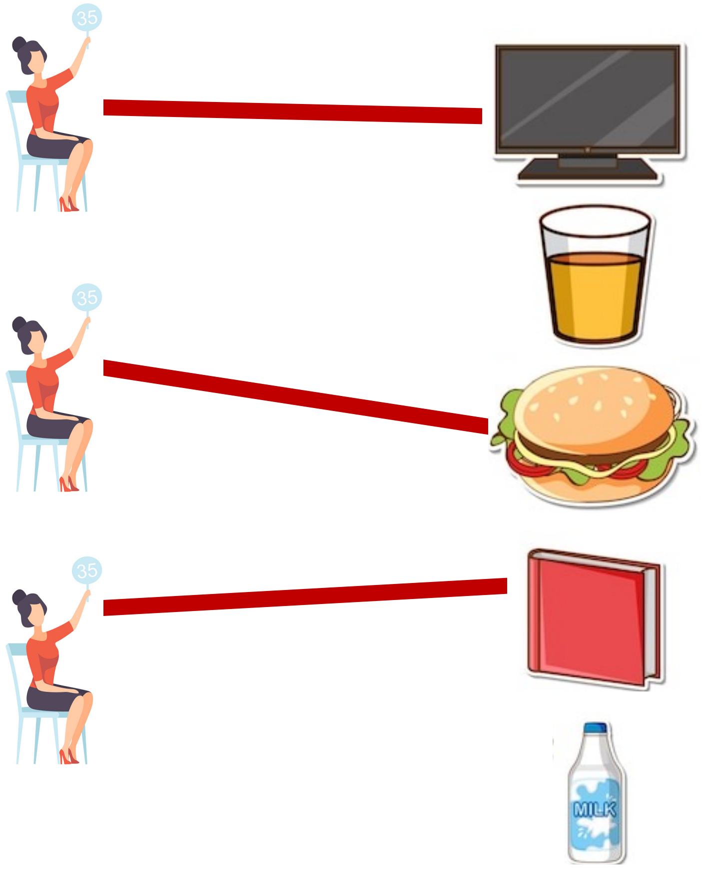
Iterate for $\left\lceil \frac{2}{\varepsilon^2} \right\rceil$ **iterations**



Can bid on item as long as **price < 1**

Auction Algorithm of [ALT21]

Final Matching



L R

Outline

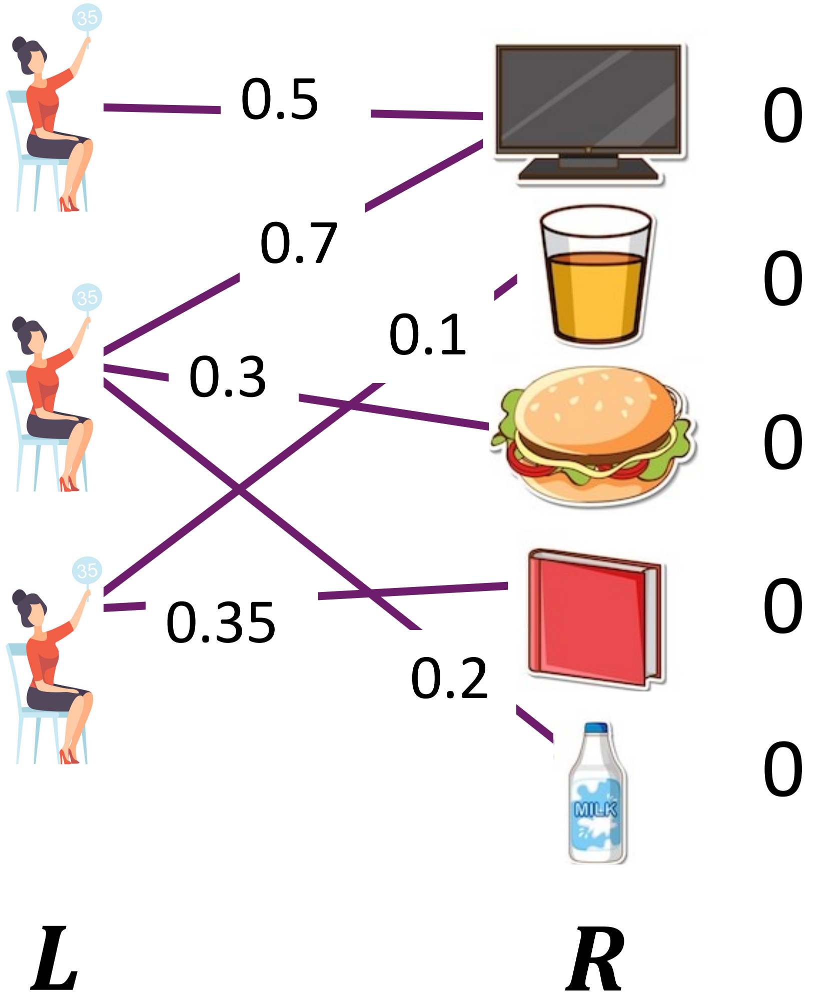
- Auction Algorithm of [ALT21] for maximum cardinality matching
- Our auction algorithm for maximum weighted matching
  - Algorithm description
  - Minimizing dependence on $\log(W)$
- Our auction algorithm for maximum *b*-matching

Our Maximum Weight Auction Algorithm

- **Bucket the edges** using buckets based on the weights of the edges
  - **Rescale weights** to **(0, 1]**
  - Edge with weight $\boldsymbol{w} \in (\mathbf{0}, \mathbf{1}]$ is in bucket ***b*** if

$$\boldsymbol{\varepsilon}^{\boldsymbol{b}-\mathbf{1}} \leq \boldsymbol{w} < \boldsymbol{\varepsilon}^{\boldsymbol{b}-\mathbf{2}}$$

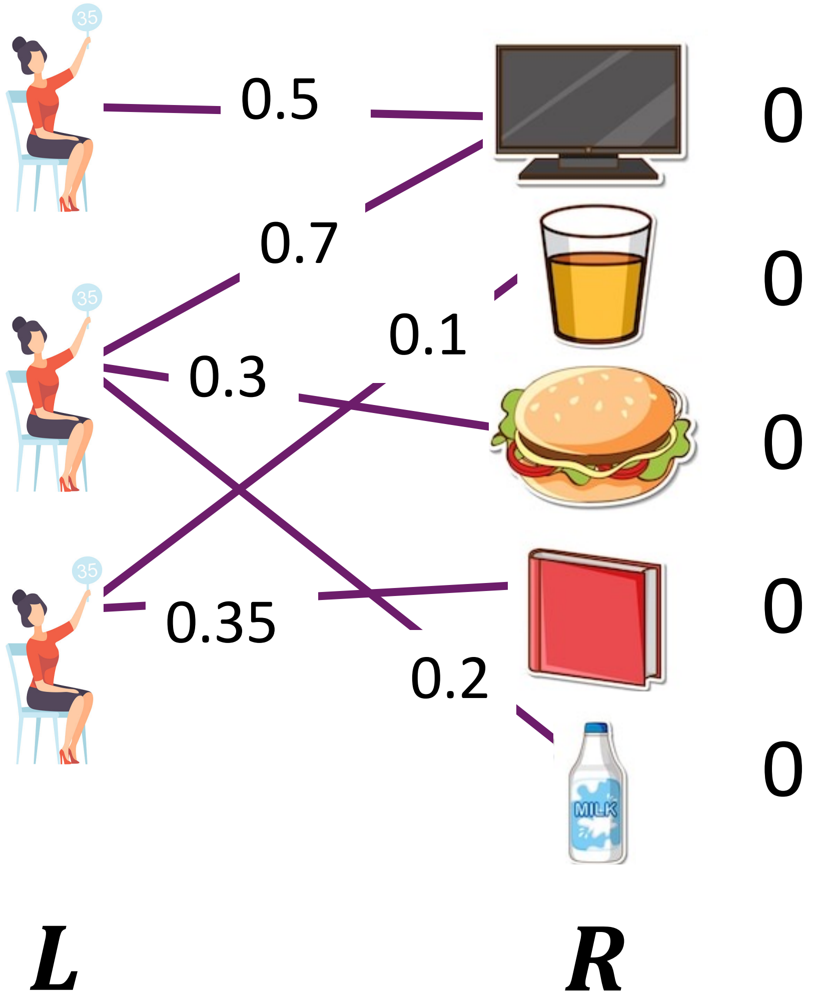
Our **Simplified** Maximum Weight Auction Algorithm



All items **start with price 0**

Our **Simplified** Maximum Weight Auction Algorithm

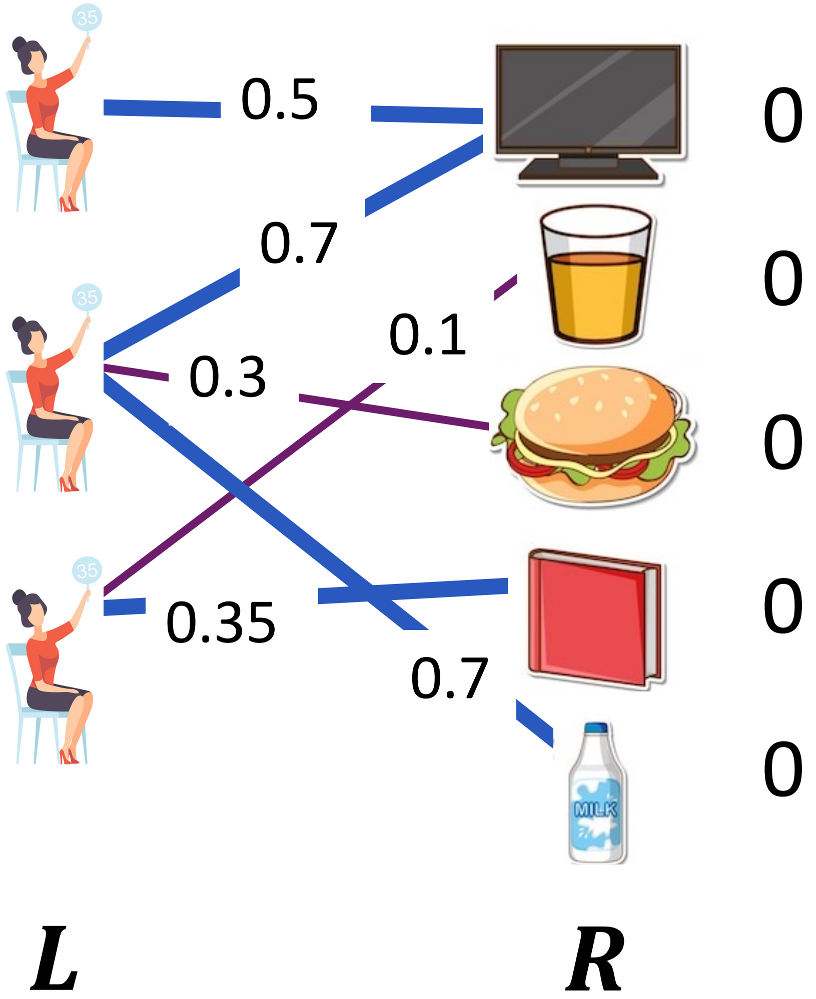
Iteratively, bidders **bid** on all **highest (value – price)** **items**



All items **start with price 0**

Our **Simplified** Maximum Weight Auction Algorithm

Iteratively, bidders **bid** on all **highest (value – price)** **items**

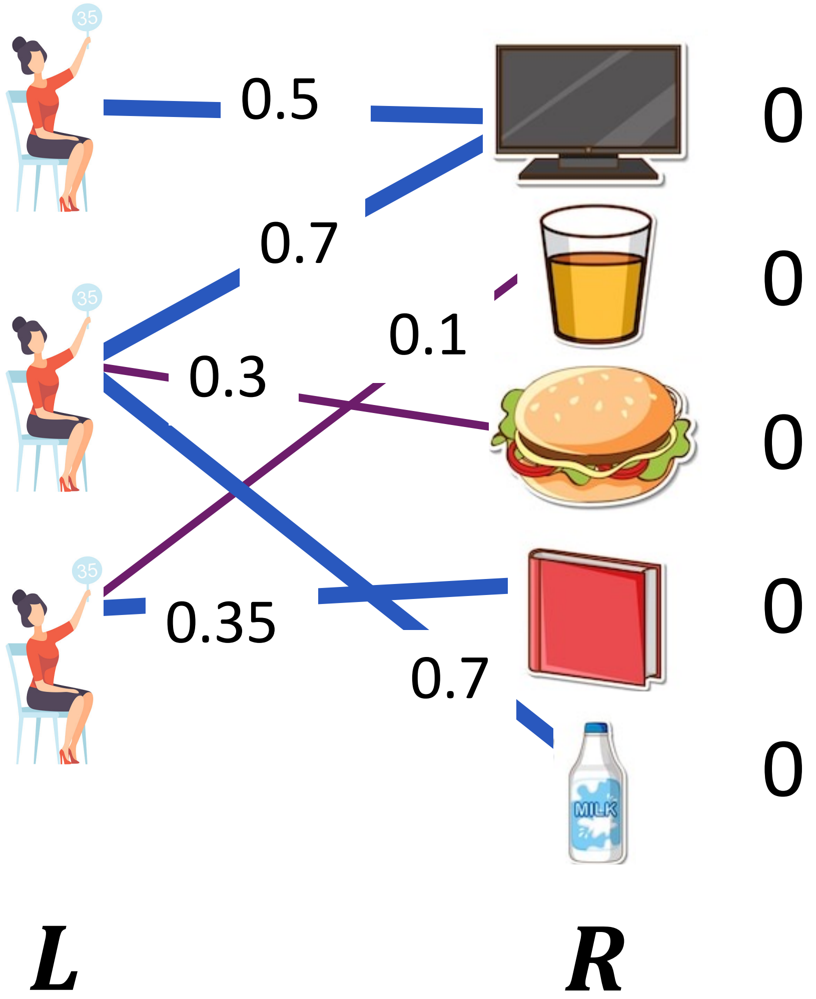


All items **start with price 0**

Our **Simplified** Maximum Weight Auction Algorithm

Iteratively, bidders **bid** on all **highest (value – price)** **items**

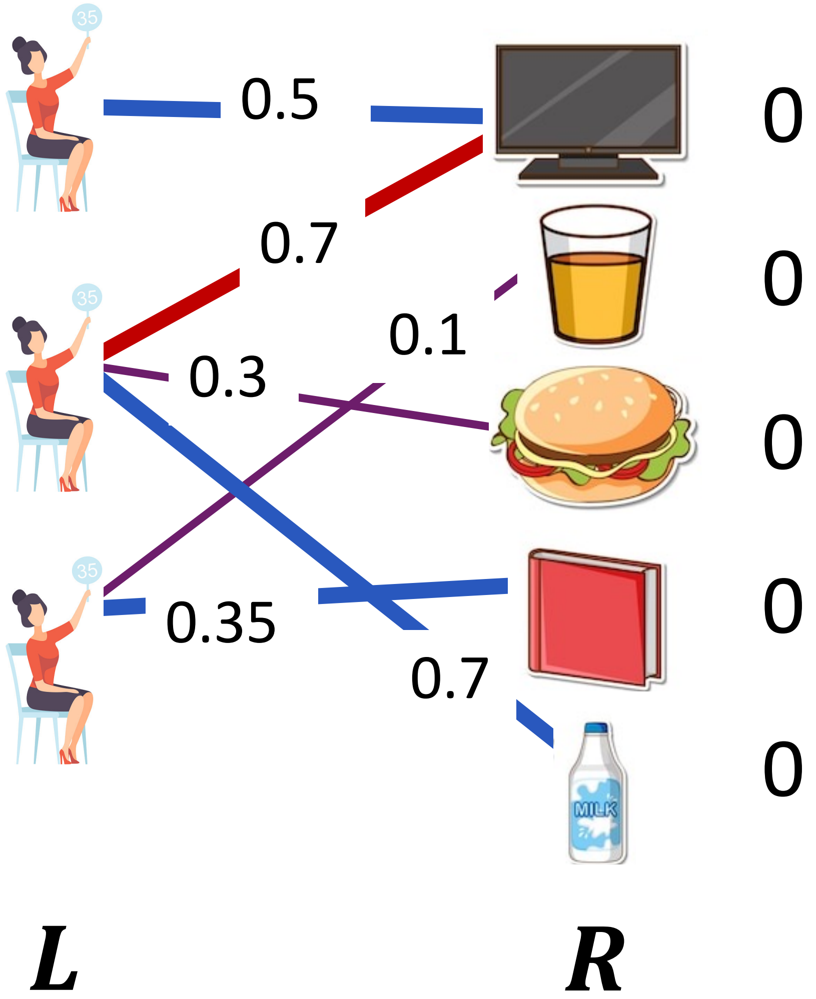
Reason: items **under contention** should be **won by edges with larger weights**



Find **maximal matching** among **induced subgraph of bid items from highest bucket down**

Our **Simplified** Maximum Weight Auction Algorithm

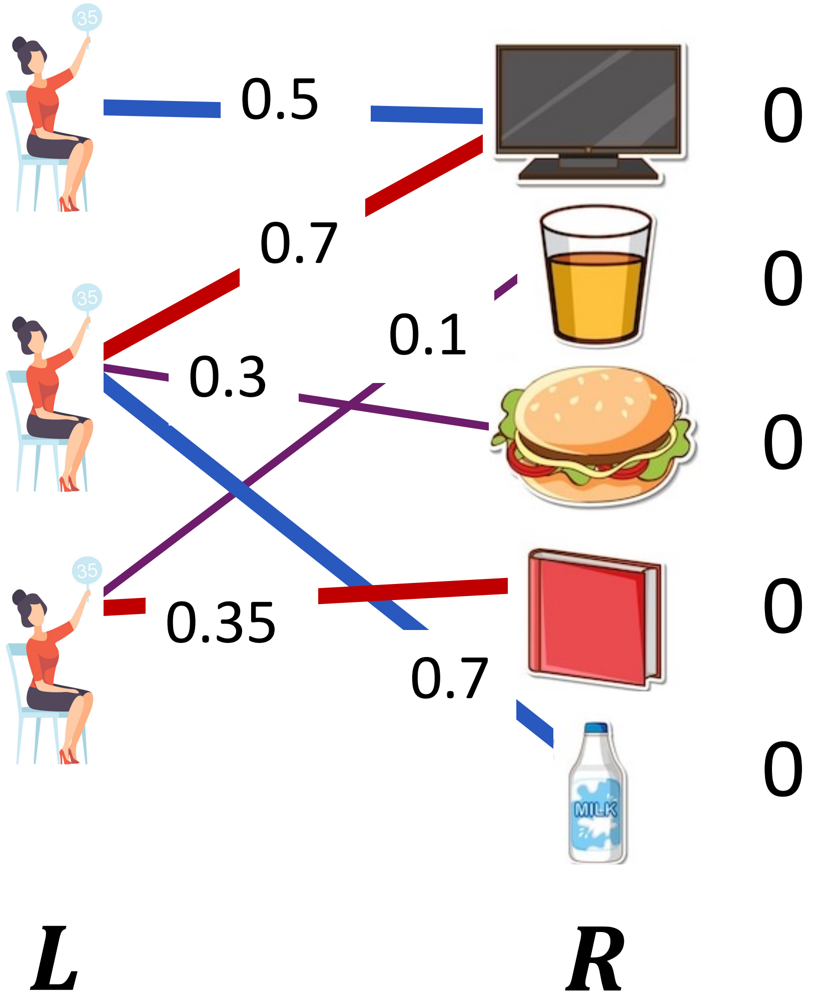
Iteratively, bidders **bid** on all **highest (value – price)** **items**



Find **maximal matching** among **induced subgraph of bid items from highest bucket down**

Our **Simplified** Maximum Weight Auction Algorithm

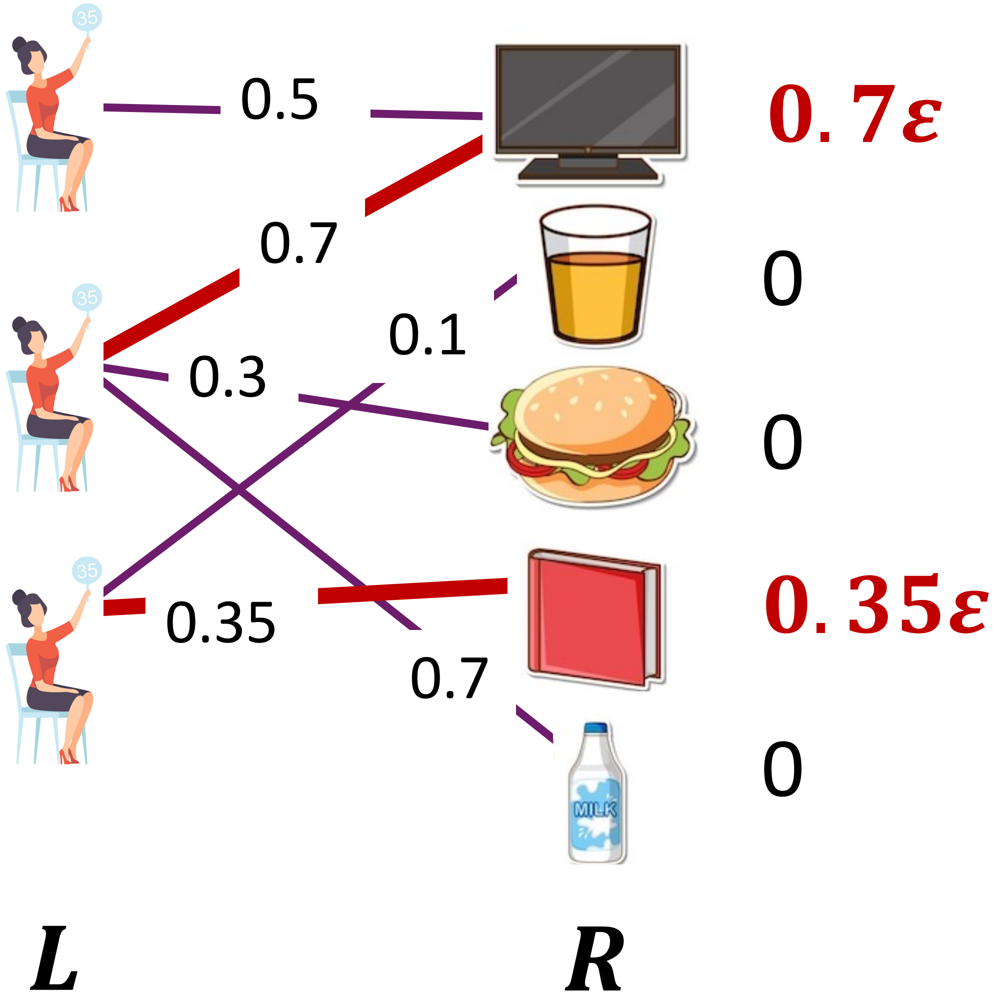
Iteratively, bidders **bid** on all **highest (value – price)** **items**



Find **maximal matching** among **induced subgraph of bid items from highest bucket down**

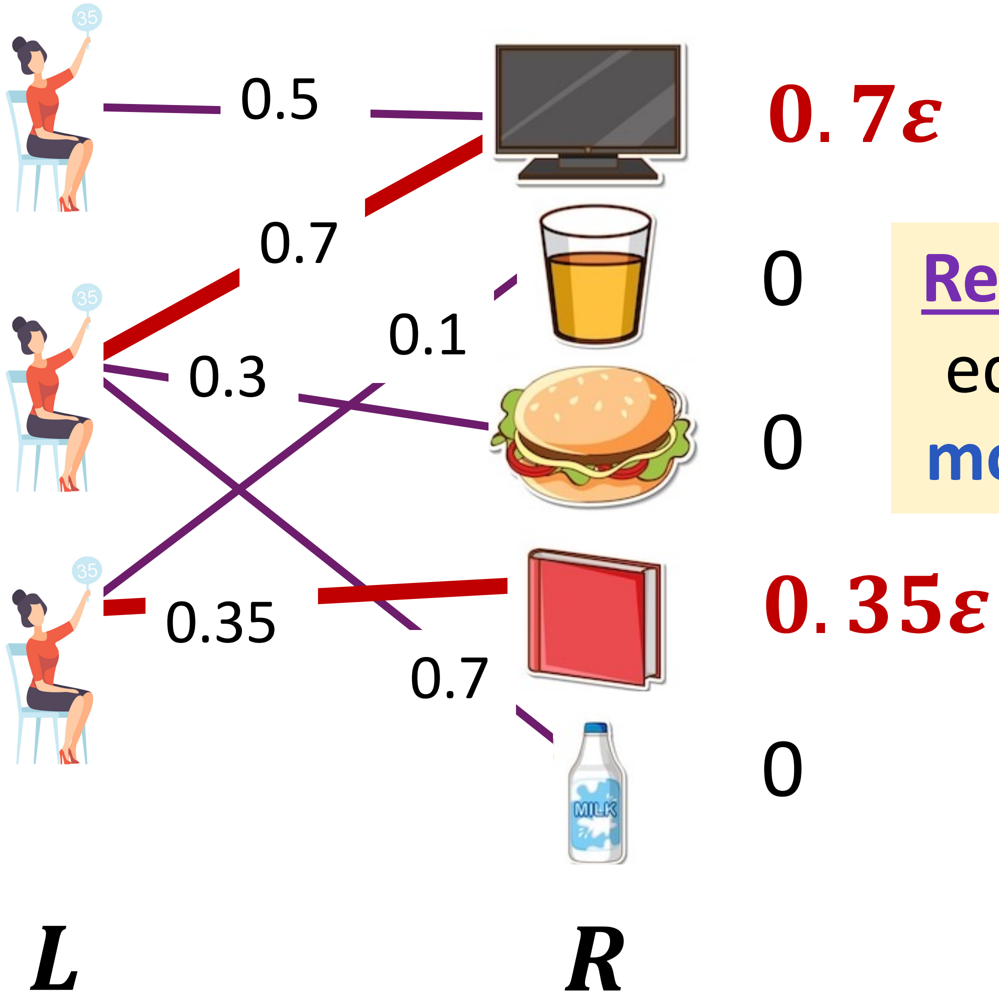
Our **Simplified** Maximum Weight Auction Algorithm

Increase price of items in matching **by $\varepsilon \cdot w$** and **maintain current matching**



Our **Simplified** Maximum Weight Auction Algorithm

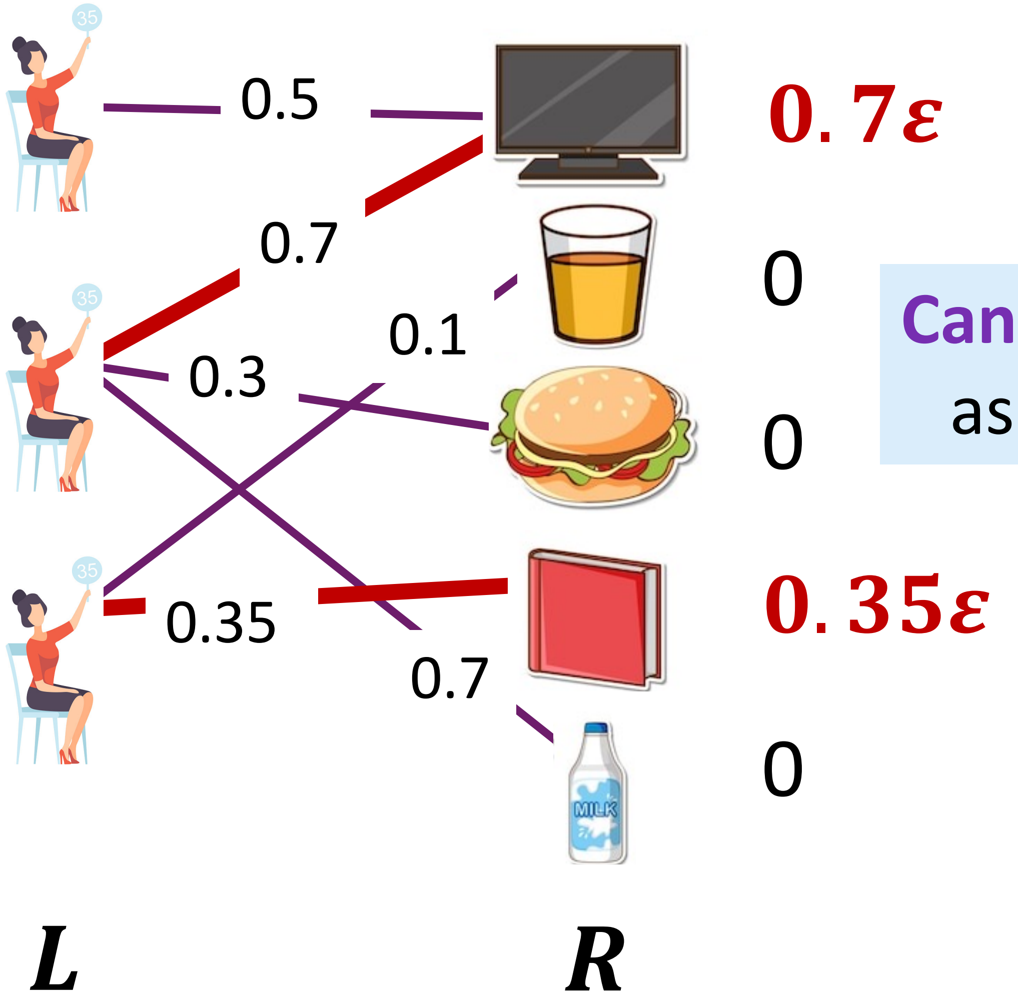
Increase price of items in matching **by $\varepsilon \cdot w$** and **maintain current matching**



Reason: higher weight edges will contribute **more to the matching**

Our **Simplified** Maximum Weight Auction Algorithm

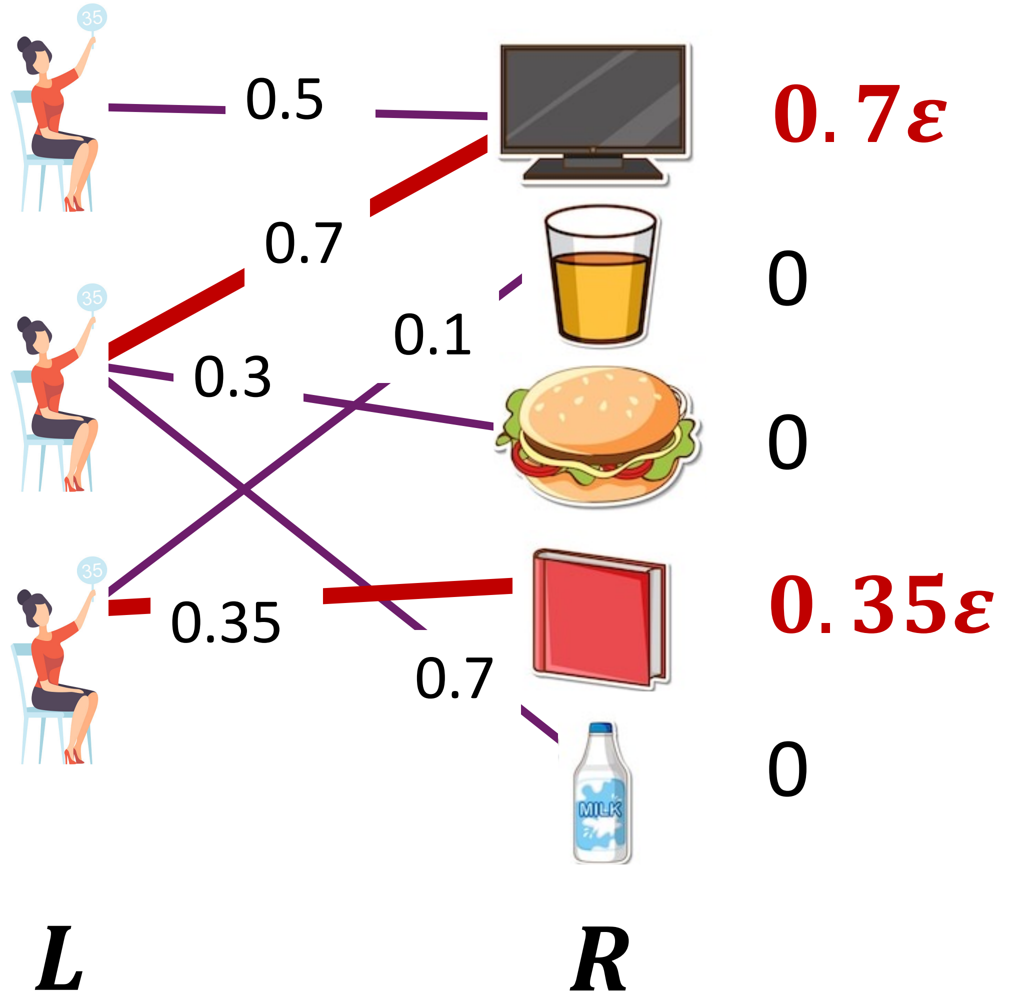
Increase price of items in matching **by $\varepsilon \cdot w$** and **maintain current matching**



Can bid on item as long as **weight - price > 0**

Our **Simplified** Maximum Weight Auction Algorithm

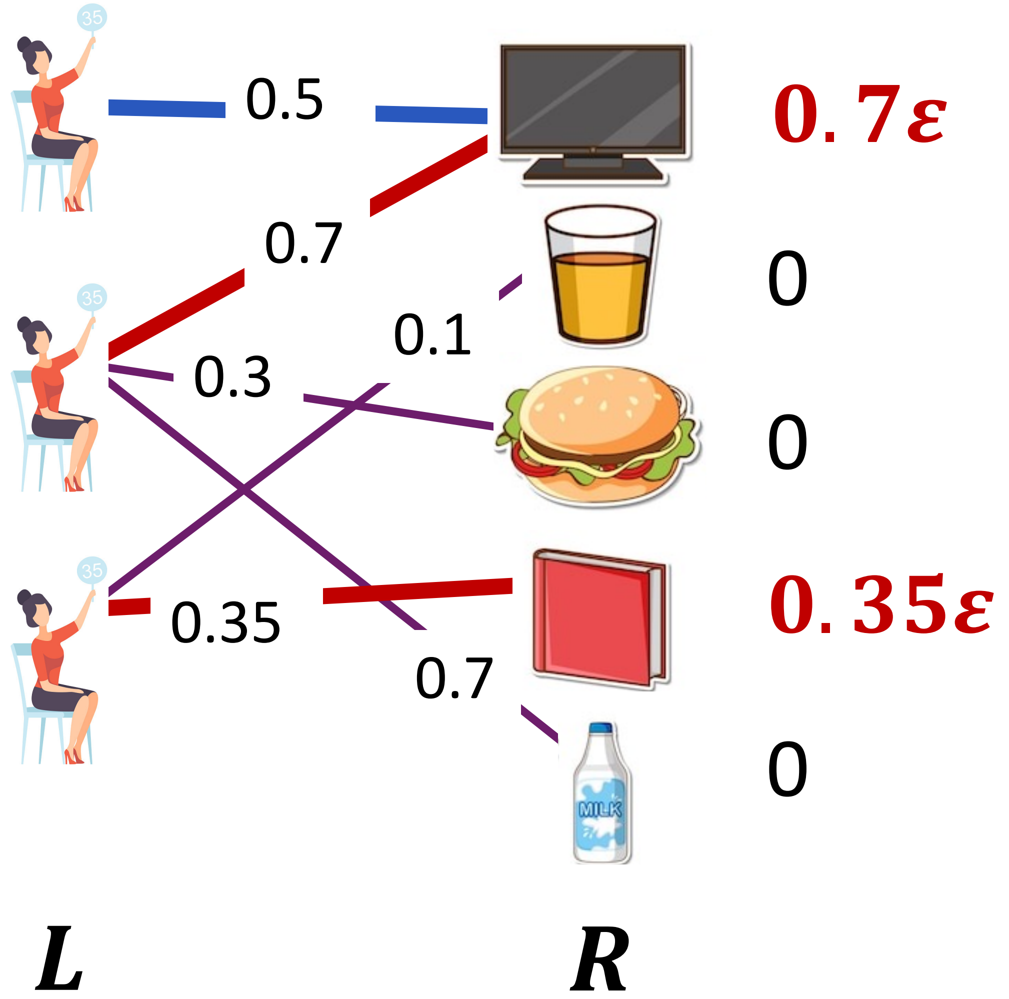
Iterate for $\left\lceil \frac{\log^2(n)}{\varepsilon^4} \right\rceil$ **iterations**



Our **Simplified** Maximum Weight Auction Algorithm

Iterate for $\left\lceil \frac{\log^2(n)}{\varepsilon^4} \right\rceil$ **iterations**

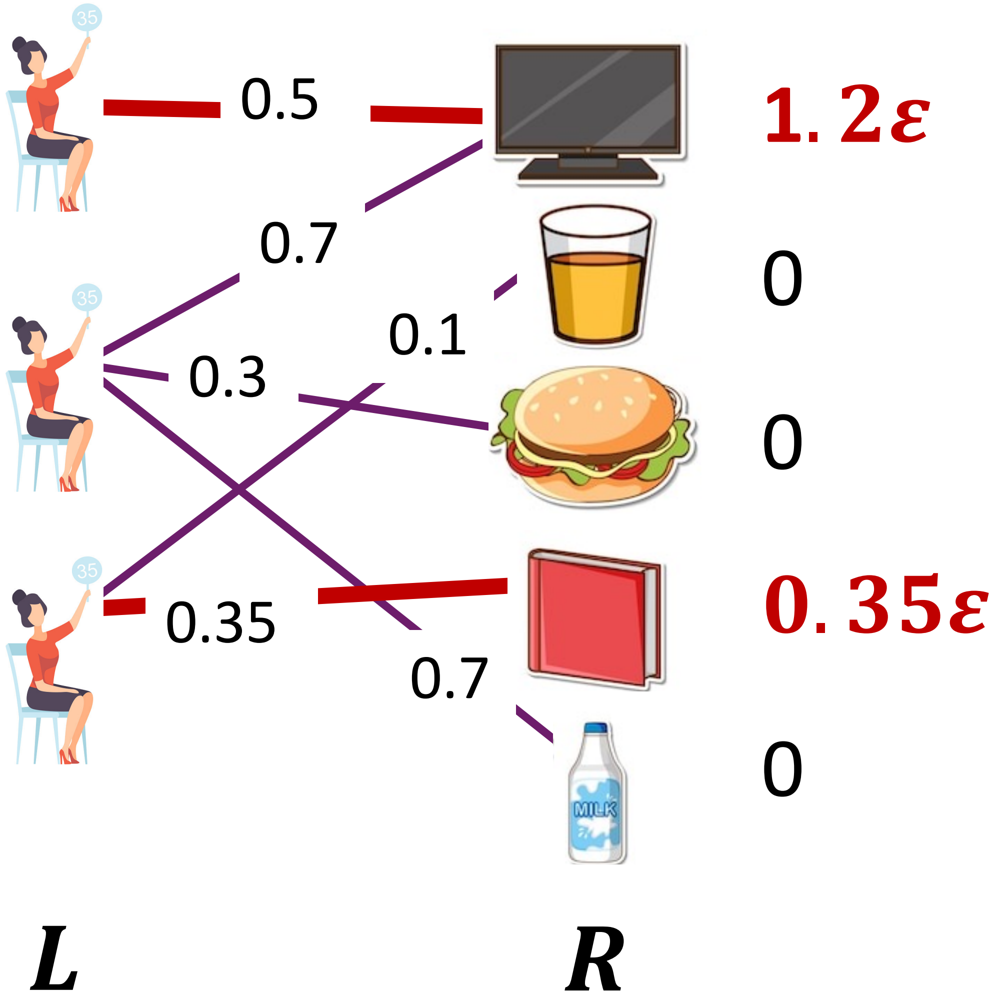
Each **unmatched** bidder bids



Our **Simplified** Maximum Weight Auction Algorithm

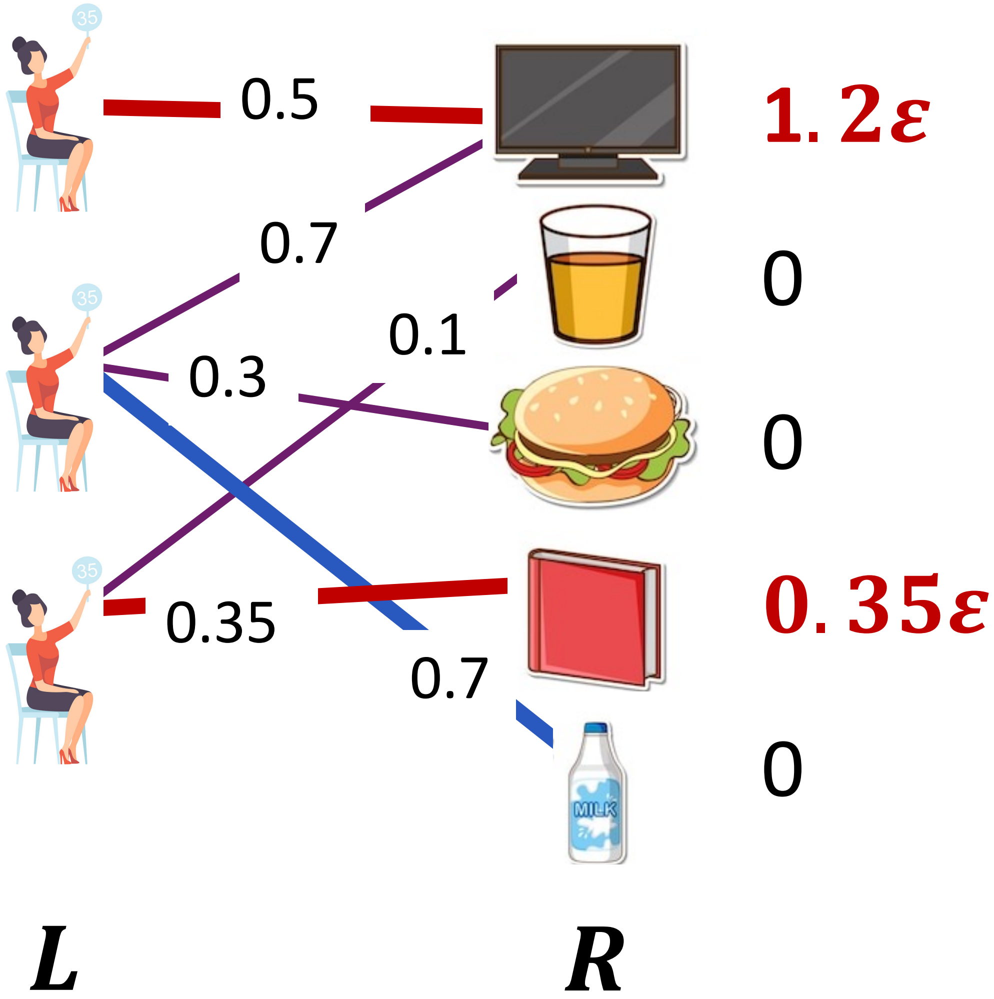
Iterate for $\left\lceil \frac{\log^2(n)}{\varepsilon^4} \right\rceil$ **iterations**

Item goes to new bidder!



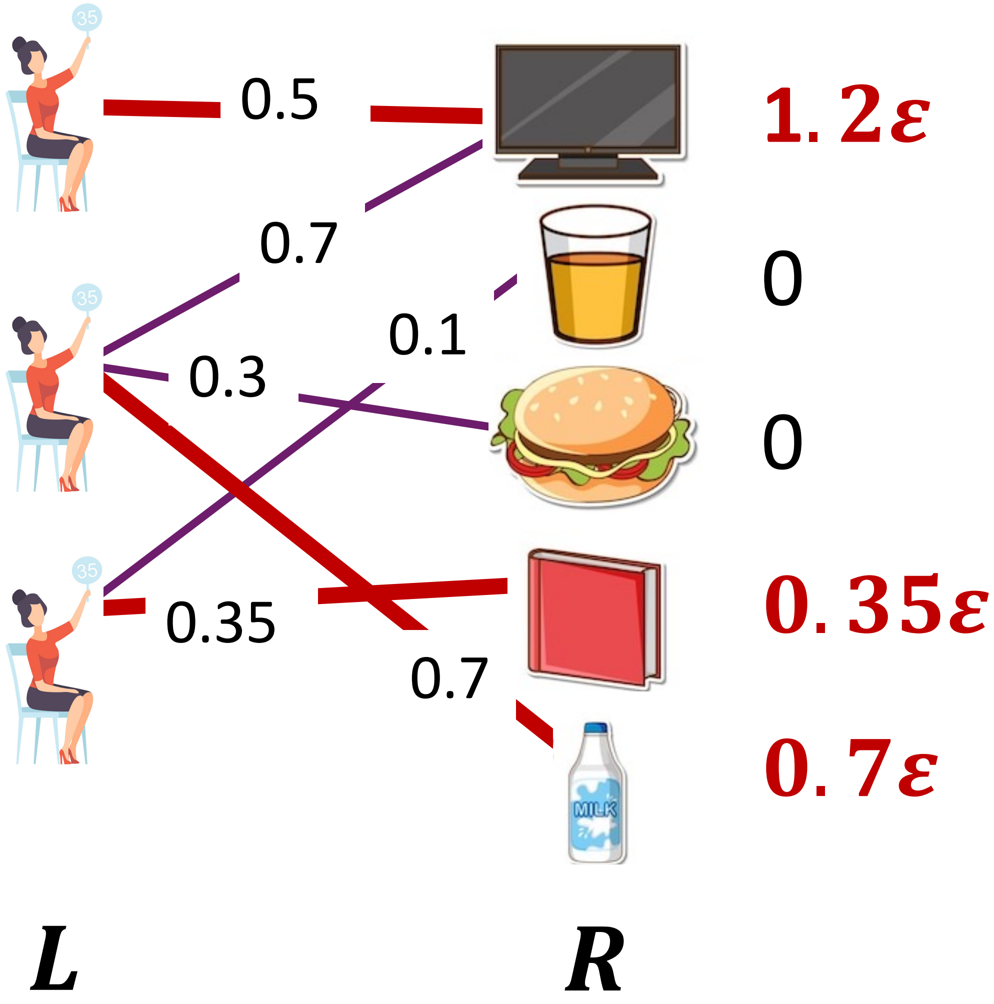
Our **Simplified** Maximum Weight Auction Algorithm

Iterate for $\left\lceil \frac{\log^2(n)}{\varepsilon^4} \right\rceil$ **iterations**



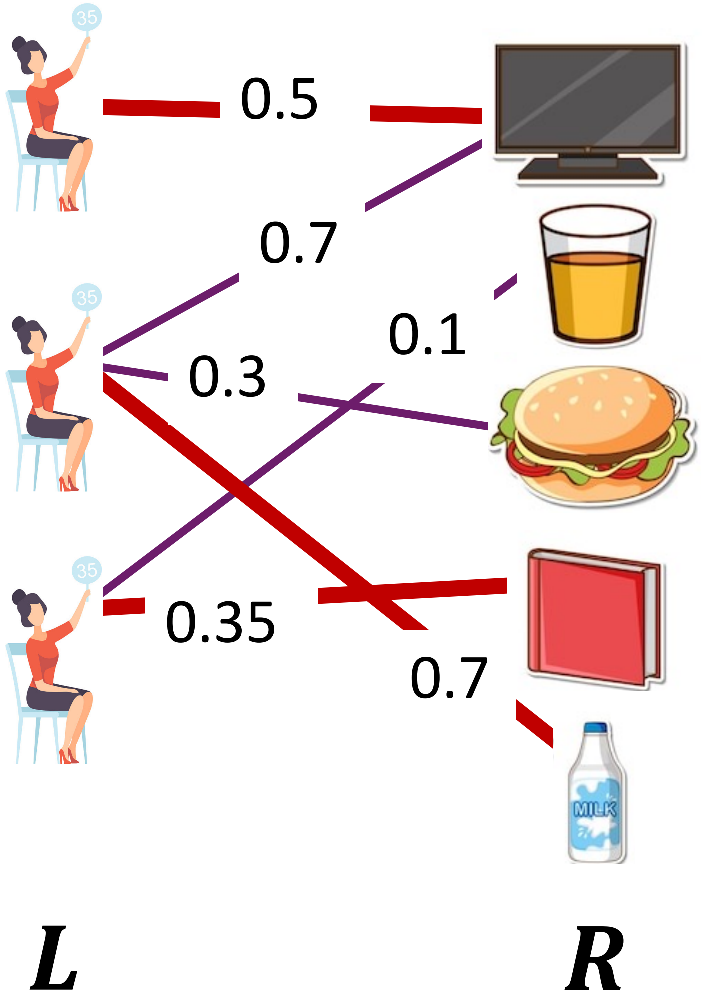
Our **Simplified** Maximum Weight Auction Algorithm

Iterate for $\left\lceil \frac{\log^2(n)}{\varepsilon^4} \right\rceil$ **iterations**



Our **Simplified** Maximum Weight Auction Algorithm

Final Matching



Minimizing Dependence on $\log(W)$

- Modified Gupta-Peng '13 transformation
  - Partition edges into levels based on edge weight
  - Each level contains multiple buckets
  - Omit certain buckets to prevent too large ratio in weights
- **Ratio of weights in each level is bounded by $\varepsilon^{-O\left(\frac{1}{\varepsilon}\right)}$**

Minimizing Dependence on $\log(W)$

- Modified Gupta-Peng '13 transformation
  - Partition edges into levels based on edge weight
  - Each level contains multiple buckets
  - Omit certain buckets to prevent too large ratio in weights
- **Ratio of weights in each level is bounded by $\varepsilon^{-O\left(\frac{1}{\varepsilon}\right)}$**

Iterate for $\left\lceil \frac{\log^2(n)}{\varepsilon^4} \right\rceil$ **iterations** → Iterate for $\left\lceil \frac{\log(n)}{\varepsilon^7} \right\rceil$ **iterations**

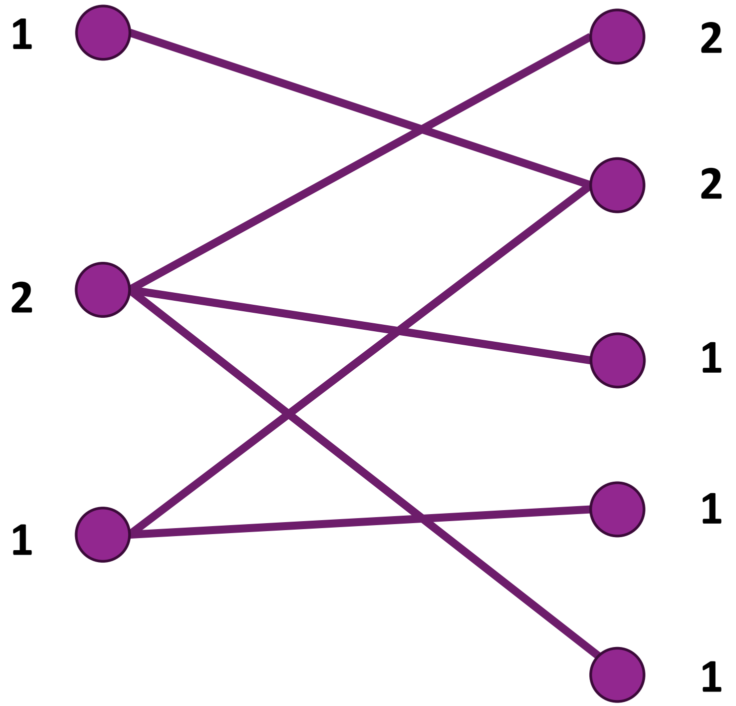
Outline

- Auction Algorithm of [ALT21] for maximum cardinality matching
- Our auction algorithm for maximum weighted matching
  - Algorithm description
  - Minimizing dependence on $\log(W)$
- Our auction algorithm for maximum *b*-matching

Very brief!

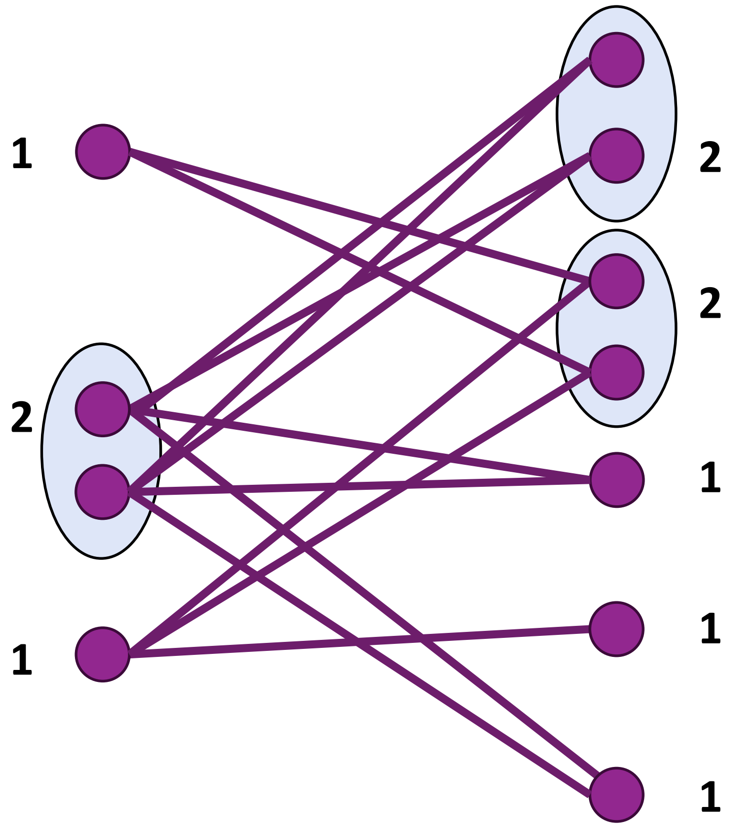
Very Simplified Maximum *b*-Matching Algorithm

Create a copy for each bidder and item equal to their *b* value



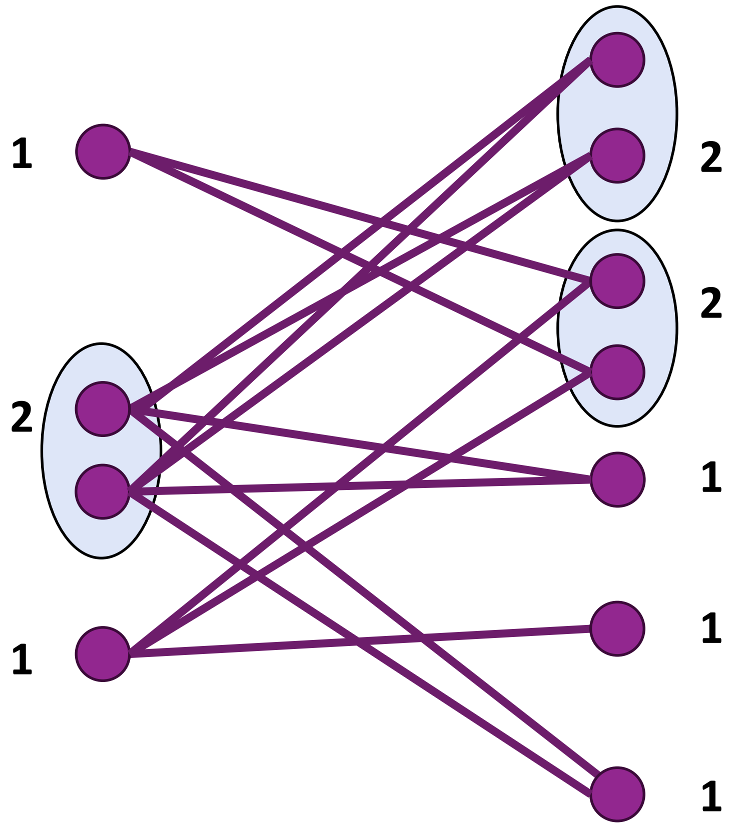
Very Simplified Maximum *b*-Matching Algorithm

Create a biclique between copies representing bidder and item



Very Simplified Maximum *b*-Matching Algorithm

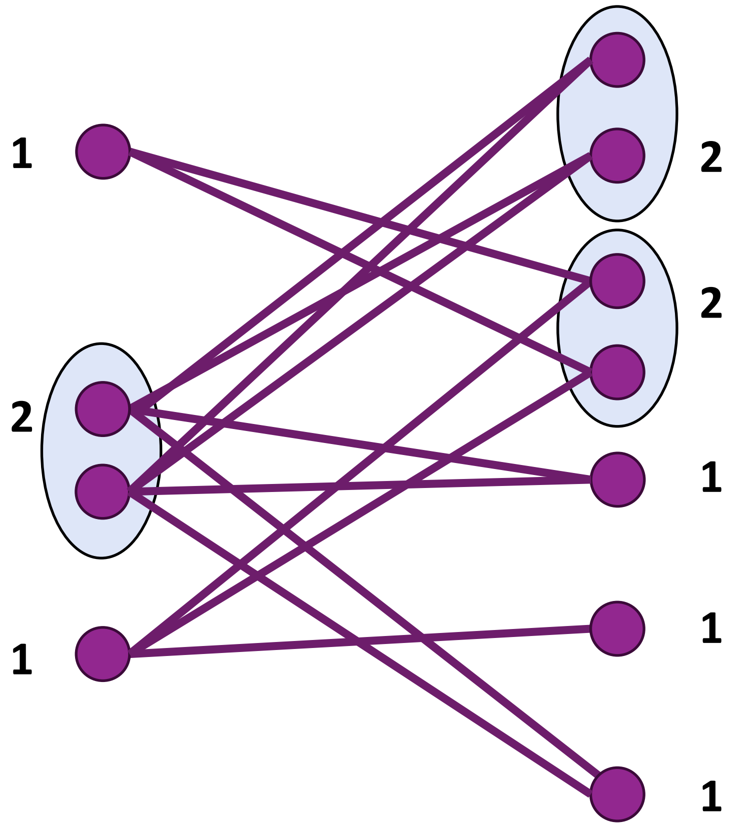
Create a biclique between copies representing bidder and item



Make sure match only one copy!

Very Simplified Maximum *b*-Matching Algorithm

Create a biclique between copies representing bidder and item



Make sure match only one copy!

Solution: each time price increases, increase the **lowest possible bidding price** for **each unmatched bidder**

Our Results

MWM = Maximum Weighted Matching
MCBM = Maximum Cardinality *b*-Matching

Model		Previous Results		Our Results	
Blackboard Distributed	MWM	$\Omega(n \log n)$ (trivial)	[DNO14]	$O\left(\frac{n\log^3(n)}{\varepsilon^8}\right)$	Theorem 3.9
	MCBM	$\Omega(nb \log n)$	trivial	$O\left(\frac{nb\log^2 n}{\varepsilon^2}\right)$	Theorem 4.8
Streaming	MWM	$O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)$ pass $O\left(\frac{n\log n}{\varepsilon^2}\right)$ space	[AG11]	$O\left(\frac{1}{\varepsilon^8}\right)$ pass $O(n \cdot \log n \cdot \log(1/\varepsilon))$ space	Theorem 3.11
	MCBM	$O(\log n/\varepsilon^3)$ pass $\widetilde{O}\left(\frac{\sum_{i\in L\cup R} b_i}{\varepsilon^3}\right)$ space	[AG18]	$O\left(\frac{1}{\varepsilon^2}\right)$ pass $O\left(\left(\sum_{i\in L} b_i + \lvert R\rvert\right)\log(1/\varepsilon)\right)$ space	Theorem 4.10
MPC	MWM	$O_\varepsilon(\log\log n)$ rounds $O_\varepsilon(n \operatorname{poly}(\log n))$ space p.m.	[GKMS19] (general)	$O\left(\frac{\log\log n}{\varepsilon^7}\right)$ rounds $O(n \cdot \log_{(1/\varepsilon)}(n))$ space p.m.	Theorem 3.15
Parallel	MWM	$O(m \cdot \operatorname{poly}(1/\varepsilon, \log n))$ work* $O(\operatorname{poly}(1/\varepsilon, \log n))$ depth*	[HS22] (general)	$O\left(\frac{m\log(n)}{\varepsilon^7}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^7}\right)$ depth	Theorem 3.13
	MCBM	N/A	N/A	$O\left(\frac{m\log n}{\varepsilon^2}\right)$ work $O\left(\frac{\log^3 n}{\varepsilon^2}\right)$ depth	Theorem 4.11

"Universal" solution across many different scalable models!