

Efficient Algorithms for Online Subadditive Combinatorial Allocations

Rian Neogi

joint work with Calum MacRury, Kanstantsin Pashkovich, Sahil Singla, Siddarth Sundaram, Chaitanya Swamy

University of Waterloo

Online combinatorial auctions

- ▶ A set N of **agents** and a set M of **items**. Items need to be allocated to agents.

Online combinatorial auctions

- ▶ A set N of **agents** and a set M of **items**. Items need to be allocated to agents.
- ▶ Agent i has a distribution $\mathcal{F}_i$ over monotone, subadditive valuations

Online combinatorial auctions

- ▶ A set N of **agents** and a set M of **items**. Items need to be allocated to agents.
- ▶ Agent i has a distribution $\mathcal{F}_i$ over monotone, subadditive valuations
- ▶ Agents arrive in adversarial order. On arrival, $v_i \sim \mathcal{F}_i$ is revealed and we must irrevocably allocate a set A_i of remaining items.

Online combinatorial auctions

- ▶ A set N of **agents** and a set M of **items**. Items need to be allocated to agents.
- ▶ Agent i has a distribution $\mathcal{F}_i$ over monotone, subadditive valuations
- ▶ Agents arrive in adversarial order. On arrival, $v_i \sim \mathcal{F}_i$ is revealed and we must irrevocably allocate a set A_i of remaining items.

$$v_i : 2^M \longrightarrow \mathbb{R}_+, \quad v_i(S \cup T) \leq v_i(S) + v_i(T).$$

Online combinatorial auctions

- ▶ A set N of **agents** and a set M of **items**. Items need to be allocated to agents.
- ▶ Agent i has a distribution $\mathcal{F}_i$ over monotone, subadditive valuations
- ▶ Agents arrive in adversarial order. On arrival, $v_i \sim \mathcal{F}_i$ is revealed and we must irrevocably allocate a set A_i of remaining items.

$$v_i : 2^M \longrightarrow \mathbb{R}_+, \quad v_i(S \cup T) \leq v_i(S) + v_i(T).$$

- ▶ Each item can be allocated at most once.

Online combinatorial auctions

- ▶ A set N of **agents** and a set M of **items**. Items need to be allocated to agents.
- ▶ Agent i has a distribution $\mathcal{F}_i$ over monotone, subadditive valuations
- ▶ Agents arrive in adversarial order. On arrival, $v_i \sim \mathcal{F}_i$ is revealed and we must irrevocably allocate a set A_i of remaining items.

$$v_i : 2^M \longrightarrow \mathbb{R}_+, \quad v_i(S \cup T) \leq v_i(S) + v_i(T).$$

- ▶ Each item can be allocated at most once.

Objective

Maximize social welfare:

$$SW(A_1, \dots, A_n) = \sum_{i \in N} v_i(A_i).$$

Main result

Theorem

For every $\varepsilon > 0$, online subadditive combinatorial auctions admit a polynomial-time $(6 + \varepsilon)$ -factor prophet inequality.

Main result

Theorem

For every $\varepsilon > 0$, online subadditive combinatorial auctions admit a polynomial-time $(6 + \varepsilon)$ -factor prophet inequality.

Equivalently, the algorithm constructs an online allocation satisfying

$$\mathbb{E}_{\substack{v_i \sim \mathcal{F}_i \\ \forall i}} \left[\sum_{i \in N} v_i(A_i) \right] \geq \frac{1}{6 + \varepsilon} \mathbb{E}_{\substack{v_i \sim \mathcal{F}_i \\ \forall i}} \left[\sum_{i \in N} v_i(\text{OPT}_i(\mathbf{v})) \right].$$

Main result

Theorem

For every $\varepsilon > 0$, online subadditive combinatorial auctions admit a polynomial-time $(6 + \varepsilon)$ -factor prophet inequality.

Equivalently, the algorithm constructs an online allocation satisfying

$$\mathbb{E}_{\substack{v_i \sim \mathcal{F}_i \\ \forall i}} \left[\sum_{i \in N} v_i(A_i) \right] \geq \frac{1}{6 + \varepsilon} \mathbb{E}_{\substack{v_i \sim \mathcal{F}_i \\ \forall i}} \left[\sum_{i \in N} v_i(\text{OPT}_i(\mathbf{v})) \right].$$

Polynomial time with sample access to $\mathcal{F}_i$ and demand-oracle access to v_i .

Prior Work

- ▶ Feige (2006): Gets a $\frac{1}{2+\epsilon}$ -approximation for the offline setting.

Prior Work

- ▶ Feige (2006): Gets a $\frac{1}{2+\epsilon}$ -approximation for the offline setting.
 - ▶ Valuations $v_1, \dots, v_n$ are known. Need to output a social welfare maximizing allocation.

Prior Work

- ▶ Feige (2006): Gets a $\frac{1}{2+\epsilon}$ -approximation for the offline setting.
 - ▶ Valuations $v_1, \dots, v_n$ are known. Need to output a social welfare maximizing allocation.
- ▶ Dutting-Kesselheim-Lucier (2020): Get a $\frac{1}{O(\log \log m)}$ -prophet inequality for the online setting.

Prior Work

- ▶ Feige (2006): Gets a $\frac{1}{2+\epsilon}$ -approximation for the offline setting.
 - ▶ Valuations $v_1, \dots, v_n$ are known. Need to output a social welfare maximizing allocation.
- ▶ Dutting-Kesselheim-Lucier (2020): Get a $\frac{1}{O(\log \log m)}$ -prophet inequality for the online setting.
 - ▶ Runs in polytime.

Prior Work

- ▶ Feige (2006): Gets a $\frac{1}{2+\epsilon}$ -approximation for the offline setting.
 - ▶ Valuations $v_1, \dots, v_n$ are known. Need to output a social welfare maximizing allocation.
- ▶ Dutting-Kesselheim-Lucier (2020): Get a $\frac{1}{O(\log \log m)}$ -prophet inequality for the online setting.
 - ▶ Runs in polytime.
- ▶ Correa-Cristi (2023): Get a $\frac{1}{6+\epsilon}$ -prophet inequality.

Prior Work

- ▶ Feige (2006): Gets a $\frac{1}{2+\epsilon}$ -approximation for the offline setting.
 - ▶ Valuations $v_1, \dots, v_n$ are known. Need to output a social welfare maximizing allocation.
- ▶ Dutting-Kesselheim-Lucier (2020): Get a $\frac{1}{O(\log \log m)}$ -prophet inequality for the online setting.
 - ▶ Runs in polytime.
- ▶ Correa-Cristi (2023): Get a $\frac{1}{6+\epsilon}$ -prophet inequality.
 - ▶ Existential result (via a fixed-point theorem).

First: the offline problem

- ▶ All valuations $(v_1, \dots, v_n)$ are known in advance.

First: the offline problem

- ▶ All valuations $(v_1, \dots, v_n)$ are known in advance.
- ▶ Choose pairwise-disjoint sets $(A_1, \dots, A_n)$.

First: the offline problem

- ▶ All valuations $(v_1, \dots, v_n)$ are known in advance.
- ▶ Choose pairwise-disjoint sets $(A_1, \dots, A_n)$.
- ▶ The goal is

$$\max_{\substack{A_1, \dots, A_n \subseteq M \\ A_i \cap A_\ell = \emptyset \ (i \neq \ell)}} \sum_{i \in N} v_i(A_i).$$

The configuration LP

$$\begin{aligned} \max \quad & \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) \\ \text{s.t.} \quad & \sum_{S \subseteq M} x_{i,S} = 1 && \forall i \in N, \\ & \sum_{i \in N} \sum_{\substack{S \subseteq M \\ j \in S}} x_{i,S} \leq 1 && \forall j \in M, \\ & x_{i,S} \in \{0, 1\}. \end{aligned}$$

The configuration LP

$$\begin{aligned} \max \quad & \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) \\ \text{s.t.} \quad & \sum_{S \subseteq M} x_{i,S} = 1 && \forall i \in N, \\ & \sum_{i \in N} \sum_{\substack{S \subseteq M \\ j \in S}} x_{i,S} \leq 1 && \forall j \in M, \\ & x_{i,S} \geq 0. \end{aligned}$$

The configuration LP

$$\begin{aligned} \max \quad & \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) \\ \text{s.t.} \quad & \sum_{S \subseteq M} x_{i,S} = 1 && \forall i \in N, \\ & \sum_{i \in N} \sum_{\substack{S \subseteq M \\ j \in S}} x_{i,S} \leq 1 && \forall j \in M, \\ & x_{i,S} \geq 0. \end{aligned}$$

Distributional view

For each agent i , x_i is a distribution over sets $W_i \subseteq M$.

For every item j ,

$$\sum_{i \in N} \Pr_{W_i \sim x_i} [j \in W_i] \leq 1.$$

In words: the total fractional demand for an item is at most one.

The configuration LP

$$\begin{aligned} \max \quad & \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) \\ \text{s.t.} \quad & \sum_{S \subseteq M} x_{i,S} = 1 && \forall i \in N, \\ & \sum_{i \in N} \sum_{\substack{S \subseteq M \\ j \in S}} x_{i,S} \leq 1 && \forall j \in M, \\ & x_{i,S} \geq 0. \end{aligned}$$

Distributional view

For each agent i , x_i is a distribution over sets $W_i \subseteq M$.

For every item j ,

$$\sum_{i \in N} \Pr_{W_i \sim x_i} [j \in W_i] \leq 1.$$

In words: the total fractional demand for an item is at most one.

Solvable in polynomial time given demand oracles.

The configuration LP

$$\begin{aligned} \max \quad & \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) \\ \text{s.t.} \quad & \sum_{S \subseteq M} x_{i,S} = 1 & \forall i \in N, \\ & \sum_{i \in N} \sum_{\substack{S \subseteq M \\ j \in S}} x_{i,S} \leq 1 & \forall j \in M, \\ & x_{i,S} \geq 0. \end{aligned}$$

Distributional view

For each agent i , x_i is a distribution over sets $W_i \subseteq M$.

For every item j ,

$$\sum_{i \in N} \Pr_{W_i \sim x_i} [j \in W_i] \leq 1.$$

In words: the total fractional demand for an item is at most one.

Solvable in polynomial time given demand oracles.

Can we round a fractional solution into a high-welfare allocation?

A simultaneous item-bidding auction

Each agent i submits

$$b_i = (b_{ij})_{j \in M} \in \mathbb{R}_+^M.$$

Each item goes to its unique highest bidder:

$$A_i = \left\{ j \in M : b_{ij} > \max_{\ell \in N - \{i\}} b_{\ell j} \right\}.$$

- The sets $(A_i)_{i \in N}$ are automatically disjoint.

A simultaneous item-bidding auction

Each agent i submits

$$b_i = (b_{ij})_{j \in M} \in \mathbb{R}_+^M.$$

Each item goes to its unique highest bidder:

$$A_i = \left\{ j \in M : b_{ij} > \max_{\ell \in N - \{i\}} b_{\ell j} \right\}.$$

- The sets $(A_i)_{i \in N}$ are automatically disjoint.

A simultaneous item-bidding auction

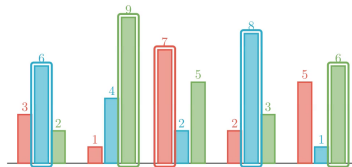
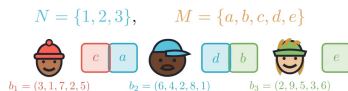
Each agent i submits

$$b_i = (b_{ij})_{j \in M} \in \mathbb{R}_+^M.$$

Each item goes to its unique highest bidder:

$$A_i = \left\{ j \in M : b_{ij} > \max_{\ell \in N - \{i\}} b_{\ell j} \right\}.$$

- The sets $(A_i)_{i \in N}$ are automatically disjoint.



A simultaneous item-bidding auction

- ▶ More generally, agent i may randomize using a distribution $\mathcal{D}_i$.

A simultaneous item-bidding auction

- ▶ More generally, agent i may randomize using a distribution $\mathcal{D}_i$.
- ▶ So each agent i samples $b_i \sim \mathcal{D}_i$.
- ▶ Assign to each agent the set

$$A_i = \left\{ j \in M : b_{ij} > \max_{\ell \in N - \{i\}} b_{\ell j} \right\}.$$

A simultaneous item-bidding auction

- ▶ More generally, agent i may randomize using a distribution $\mathcal{D}_i$.
- ▶ So each agent i samples $b_i \sim \mathcal{D}_i$.
- ▶ Assign to each agent the set

$$A_i = \left\{ j \in M : b_{ij} > \max_{\ell \in N - \{i\}} b_{\ell j} \right\}.$$

How might the agents strategize so that the final allocation $(A_1, \dots, A_n)$ has high social welfare?

A mirroring trick

Let $S \subseteq M$.

A mirroring trick

Let $S \subseteq M$.

Let $p, q \in \mathbb{Z}_+^S$ be independent and identically distributed. Define

$$A = \{j \in S : p_j \geq q_j\}, \quad B = \{j \in S : q_j \geq p_j\}.$$

A mirroring trick

Let $S \subseteq M$.

Let $p, q \in \mathbb{Z}_+^S$ be independent and identically distributed. Define

$$A = \{j \in S : p_j \geq q_j\}, \quad B = \{j \in S : q_j \geq p_j\}.$$

Subadditivity

Since $A \cup B = S$,

$$v(A) + v(B) \geq v(S).$$

A mirroring trick

Let $S \subseteq M$.

Let $p, q \in \mathbb{Z}_+^S$ be independent and identically distributed. Define

$$A = \{j \in S : p_j \geq q_j\}, \quad B = \{j \in S : q_j \geq p_j\}.$$

Subadditivity

Since $A \cup B = S$,

$$v(A) + v(B) \geq v(S).$$

Symmetry

Since p and q are i.i.d.,

$$\mathbb{E}[v(A)] = \mathbb{E}[v(B)].$$

A mirroring trick

Let $S \subseteq M$.

Let $p, q \in \mathbb{Z}_+^S$ be independent and identically distributed. Define

$$A = \{j \in S : p_j \geq q_j\}, \quad B = \{j \in S : q_j \geq p_j\}.$$

Subadditivity

Since $A \cup B = S$,

$$v(A) + v(B) \geq v(S).$$

Symmetry

Since p and q are i.i.d.,

$$\mathbb{E}[v(A)] = \mathbb{E}[v(B)].$$

$$2 \mathbb{E}[v(A)] = \mathbb{E}[v(A) + v(B)] \geq v(S)$$

A mirroring trick

Let $S \subseteq M$.

Let $p, q \in \mathbb{Z}_+^S$ be independent and identically distributed. Define

$$A = \{j \in S : p_j \geq q_j\}, \quad B = \{j \in S : q_j \geq p_j\}.$$

Subadditivity

Since $A \cup B = S$,

$$v(A) + v(B) \geq v(S).$$

Symmetry

Since p and q are i.i.d.,

$$\mathbb{E}[v(A)] = \mathbb{E}[v(B)].$$

$$2 \mathbb{E}[v(A)] = \mathbb{E}[v(A) + v(B)] \geq v(S) \implies \mathbb{E}_{p,q}[v(\{j \in S : p_j \geq q_j\})] \geq \frac{1}{2} v(S)$$

The mirroring strategy

Fix agent i and $\mathcal{D}_{-i}$. The strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Samples an interest set
 $W \sim (x_{i,s})_{s \subseteq M}$.

The mirroring strategy

Fix agent i and $\mathcal{D}_{-i}$. The strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Samples an interest set $W \sim (x_{i,s})_{s \subseteq M}$.
2. Samples $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.

The mirroring strategy

Fix agent i and $\mathcal{D}_{-i}$. The strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Samples an interest set $W \sim (x_{i,s})_{s \subseteq M}$.
2. Samples $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Returns

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

The mirroring strategy

Fix agent i and $\mathcal{D}_{-i}$. The strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Samples an interest set $W \sim (x_{i,s})_{s \subseteq M}$.
2. Samples $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Returns

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

The mirroring strategy

Fix agent i and $\mathcal{D}_{-i}$. The strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Samples an interest set $W \sim (x_{i,s})_{s \subseteq M}$.
2. Samples $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Returns

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

On W , bid according to a fresh sample of the competition.

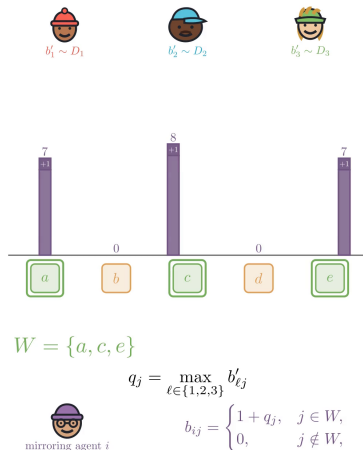
The mirroring strategy

Fix agent i and $\mathcal{D}_{-i}$. The strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Samples an interest set $W \sim (x_{i,S})_{S \subseteq M}$.
2. Samples $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Returns

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

On W , bid according to a fresh sample of the competition.



Mirroring captures half of the LP value

Let $b_\ell \sim \mathcal{D}_\ell$ be the opponents' actual bids and $b'_\ell \sim \mathcal{D}_\ell$ the fresh samples used by the mirror. Conditional on W ,

$$\begin{aligned} A_i &= \left\{ j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j} \right\} \\ &= \left\{ j \in W : \underbrace{\max_{\ell \neq i} b'_{\ell j}}_{p_j} \geq \underbrace{\max_{\ell \neq i} b_{\ell j}}_{q_j} \right\}. \end{aligned}$$

Mirroring strategy

To sample from $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Sample $W \sim x_i$.
2. Independently sample $b'_\ell \sim \mathcal{D}_\ell$ for every $\ell \neq i$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Mirroring captures half of the LP value

Let $b_\ell \sim \mathcal{D}_\ell$ be the opponents' actual bids and $b'_\ell \sim \mathcal{D}_\ell$ the fresh samples used by the mirror. Conditional on W ,

$$\begin{aligned} A_i &= \left\{ j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j} \right\} \\ &= \left\{ j \in W : \underbrace{\max_{\ell \neq i} b'_{\ell j}}_{p_j} \geq \underbrace{\max_{\ell \neq i} b_{\ell j}}_{q_j} \right\}. \end{aligned}$$

The two maxima are i.i.d. on A_i . The previous claim gives

$$\mathbb{E}[v_i(A_i) \mid W] \geq \frac{1}{2} v_i(W).$$

Mirroring strategy

To sample from $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Sample $W \sim x_i$.
2. Independently sample $b'_\ell \sim \mathcal{D}_\ell$ for every $\ell \neq i$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Mirroring captures half of the LP value

Let $b_\ell \sim \mathcal{D}_\ell$ be the opponents' actual bids and $b'_\ell \sim \mathcal{D}_\ell$ the fresh samples used by the mirror. Conditional on W ,

$$\begin{aligned} A_i &= \left\{ j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j} \right\} \\ &= \left\{ j \in W : \underbrace{\max_{\ell \neq i} b'_{\ell j}}_{p_j} \geq \underbrace{\max_{\ell \neq i} b_{\ell j}}_{q_j} \right\}. \end{aligned}$$

The two maxima are i.i.d. on A_i . The previous claim gives

$$\mathbb{E}[v_i(A_i) \mid W] \geq \frac{1}{2} v_i(W).$$

Averaging over $W \sim x_i$,

$$\mathbb{E}[v_i(A_i)] \geq \frac{1}{2} \sum_{S \subseteq M} x_{i,S} v_i(S)$$

Mirroring strategy

To sample from $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Sample $W \sim x_i$.
2. Independently sample $b'_\ell \sim \mathcal{D}_\ell$ for every $\ell \neq i$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

What if every agent mirrors?

Definition

Bid distributions $(\mathcal{D}_1, \dots, \mathcal{D}_n)$ are **mutually mirroring** if

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$$

What if every agent mirrors?

Definition

Bid distributions $(\mathcal{D}_1, \dots, \mathcal{D}_n)$ are **mutually mirroring** if

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$$

If all agents use mutually mirroring distributions, then

$$\mathbb{E} \left[\sum_{i \in N} v_i(A_i) \right] \geq \frac{1}{2} \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) = \frac{1}{2} \text{LP}.$$

What if every agent mirrors?

Definition

Bid distributions $(\mathcal{D}_1, \dots, \mathcal{D}_n)$ are **mutually mirroring** if

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$$

If all agents use mutually mirroring distributions, then

$$\mathbb{E} \left[\sum_{i \in N} v_i(A_i) \right] \geq \frac{1}{2} \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) = \frac{1}{2} \text{LP}.$$

Existence

The mutually mirroring equations form a fixed-point problem. The fixed-point can be shown to exist by Brouwer's fixed-point theorem.

What if every agent mirrors?

Definition

Bid distributions $(\mathcal{D}_1, \dots, \mathcal{D}_n)$ are **mutually mirroring** if

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$$

If all agents use mutually mirroring distributions, then

$$\mathbb{E} \left[\sum_{i \in N} v_i(A_i) \right] \geq \frac{1}{2} \sum_{i \in N} \sum_{S \subseteq M} x_{i,S} v_i(S) = \frac{1}{2} \text{LP}.$$

Existence

The mutually mirroring equations form a fixed-point problem. The fixed-point can be shown to exist by Brouwer's fixed-point theorem.

Can we sample mutually mirroring bids in polynomial time?

Unrolling the fixed point

Mutually mirroring distributions satisfy $\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i})?$$



Unrolling the fixed point

Mutually mirroring distributions satisfy $\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N$.

- If I had samples $b'_\ell \sim \mathcal{D}_\ell$, for $\ell \neq i$, I can sample $W \sim x_i$, then set

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i})?$$



Unrolling the fixed point

Mutually mirroring distributions satisfy $\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N$.

- If I had samples $b'_\ell \sim \mathcal{D}_\ell$, for $\ell \neq i$, I can sample $W \sim x_i$, then set

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

- How can I simulate samples $b'_\ell \sim \mathcal{D}_\ell$?

$$\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i})?$$



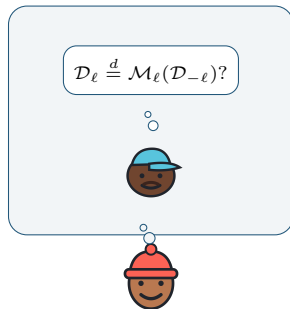
Unrolling the fixed point

Mutually mirroring distributions satisfy $\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$

- ▶ If I had samples $b'_\ell \sim \mathcal{D}_\ell$, for $\ell \neq i$, I can sample $W \sim x_i$, then set

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

- ▶ How can I simulate samples $b'_\ell \sim \mathcal{D}_\ell$?
- ▶ If agent ℓ had samples $b''_k \sim \mathcal{D}_k$ for $k \neq \ell$, then he does the same mirroring thing. . .



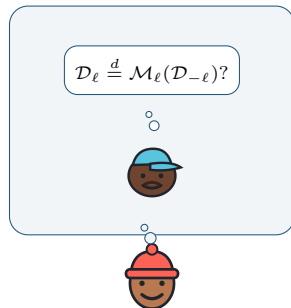
Unrolling the fixed point

Mutually mirroring distributions satisfy $\mathcal{D}_i \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_{-i}) \quad \forall i \in N.$

- ▶ If I had samples $b'_\ell \sim \mathcal{D}_\ell$, for $\ell \neq i$, I can sample $W \sim x_i$, then set

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

- ▶ How can I simulate samples $b'_\ell \sim \mathcal{D}_\ell$?
- ▶ If agent ℓ had samples $b''_k \sim \mathcal{D}_k$ for $k \neq \ell$, then he does the same mirroring thing...
- ▶ And so on...



The recursive algorithm

SimBids(i, S)

1. If $S = \emptyset$, return the all-zero vector.
2. Sample an interest set $W \sim x_i$.
3. For every $\ell \neq i$, recursively compute

$$b'_\ell \leftarrow \text{SimBids}(\ell, S \cap W).$$

4. Return $b_i \in \mathbb{Z}_+^S$ with

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in S \cap W, \\ 0, & j \notin S \cap W. \end{cases}$$

Mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Sample an interest set $W \sim x_i$.
2. Sample $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Intended Meaning: SimBids(i, S) returns a bid sampled from $\mathcal{D}_i$, restricted to items in S .

The recursive algorithm

SimBids(i, S)

1. If $S = \emptyset$, return the all-zero vector.
2. Sample an interest set $W \sim x_i$.
3. For every $\ell \neq i$, recursively compute

$$b'_\ell \leftarrow \text{SimBids}(\ell, S \cap W).$$

4. Return $b_i \in \mathbb{Z}_+^S$ with

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in S \cap W, \\ 0, & j \notin S \cap W. \end{cases}$$

Mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Sample an interest set $W \sim x_i$.
2. Sample $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Intended Meaning: SimBids(i, S) returns a bid sampled from $\mathcal{D}_i$, restricted to items in S .

Lemma

If $\mathcal{D}_i$ is the distribution returned by SimBids(i, M), then $(\mathcal{D}_1, \dots, \mathcal{D}_n)$ are mutually mirroring.

The recursive algorithm

SimBids(i, S)

1. If $S = \emptyset$, return the all-zero vector.
2. Sample an interest set $W \sim x_i$.
3. For every $\ell \neq i$, recursively compute

$$b'_\ell \leftarrow \text{SimBids}(\ell, S \cap W).$$

4. Return $b_i \in \mathbb{Z}_+^S$ with

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in S \cap W, \\ 0, & j \notin S \cap W. \end{cases}$$

Mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$:

1. Sample an interest set $W \sim x_i$.
2. Sample $b'_\ell \sim \mathcal{D}_\ell$ independently for every $\ell \neq i$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Intended Meaning: SimBids(i, S) returns a bid sampled from $\mathcal{D}_i$, restricted to items in S .

Lemma

If $\mathcal{D}_i$ is the distribution returned by SimBids(i, M), then $(\mathcal{D}_1, \dots, \mathcal{D}_n)$ are mutually mirroring.

Proof: By induction.

Unrolling the recursion

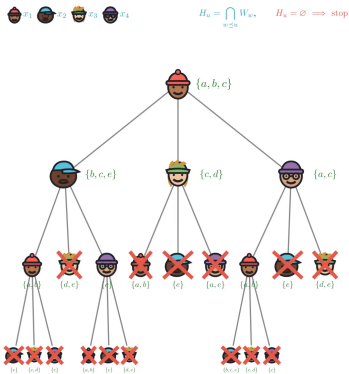
SimBids(i, S)

1. If $S = \emptyset$, return the all-zero vector.
2. Sample an interest set $W \sim x_i$.
3. For every $\ell \neq i$, recursively compute

$$b'_\ell \leftarrow \text{SimBids}(\ell, S \cap W).$$

4. Return $b_i \in \mathbb{Z}_+^S$ with

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in S \cap W, \\ 0, & j \notin S \cap W. \end{cases}$$



Unrolling the recursion

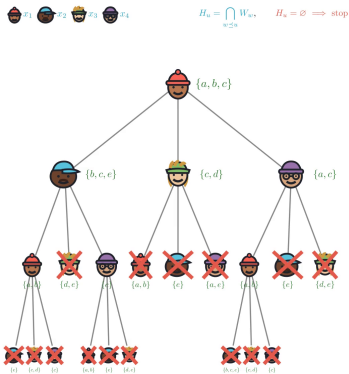
SimBids(i, S)

1. If $S = \emptyset$, return the all-zero vector.
2. Sample an interest set $W \sim x_i$.
3. For every $\ell \neq i$, recursively compute

$$b'_\ell \leftarrow \text{SimBids}(\ell, S \cap W).$$

4. Return $b_i \in \mathbb{Z}_+^S$ with

$$b_{ij} = \begin{cases} 1 + \max_{\ell \neq i} b'_{\ell j}, & j \in S \cap W, \\ 0, & j \notin S \cap W. \end{cases}$$



Stop when intersection of interest sets on root-to-leaf path becomes $\emptyset$

Why should the recursion terminate?

At first glance, the recursion can simulate an agent again:

$$\text{SimBids}(i, M) \longrightarrow \text{SimBids}(\ell, W) \longrightarrow \text{SimBids}(i, W \cap W_\ell).$$

Why should the recursion terminate?

At first glance, the recursion can simulate an agent again:

$$\text{SimBids}(i, M) \longrightarrow \text{SimBids}(\ell, W) \longrightarrow \text{SimBids}(i, W \cap W_\ell).$$

The progress is that bids are needed only on the smaller set $W \cap W_\ell$.

Why should the recursion terminate?

At first glance, the recursion can simulate an agent again:

$$\text{SimBids}(i, M) \longrightarrow \text{SimBids}(\ell, W) \longrightarrow \text{SimBids}(i, W \cap W_\ell).$$

The progress is that bids are needed only on the smaller set $W \cap W_\ell$.

Branching

Each internal node creates up to $n - 1$ new recursive calls.

Why should the recursion terminate?

At first glance, the recursion can simulate an agent again:

$$\text{SimBids}(i, M) \longrightarrow \text{SimBids}(\ell, W) \longrightarrow \text{SimBids}(i, W \cap W_\ell).$$

The progress is that bids are needed only on the smaller set $W \cap W_\ell$.

Branching

Each internal node creates up to $n - 1$ new recursive calls.

Repeated Intersections

Repeated intersections remove items and eventually reach $\emptyset$.

Why should the recursion terminate?

At first glance, the recursion can simulate an agent again:

$$\text{SimBids}(i, M) \longrightarrow \text{SimBids}(\ell, W) \longrightarrow \text{SimBids}(i, W \cap W_\ell).$$

The progress is that bids are needed only on the smaller set $W \cap W_\ell$.

Branching

Each internal node creates up to $n - 1$ new recursive calls.

Repeated Intersections

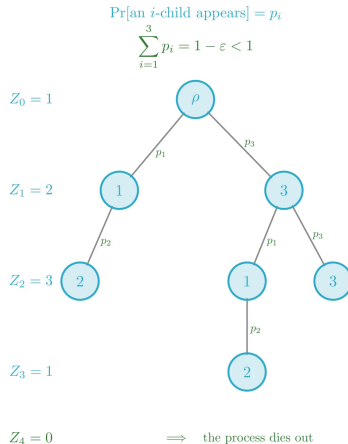
Repeated intersections remove items and eventually reach $\emptyset$.

Intersections dominate when distributions $(x_1, \dots, x_n)$ come from configuration LP.

$$\sum_{i \in N} \sum_{S: j \in S} x_{i,S} \leq 1, \forall j \in M$$

Quick detour: branching processes

A branching process $\mathcal{B}$ begins with a root.
Every node independently creates child ℓ with probability p_ℓ .

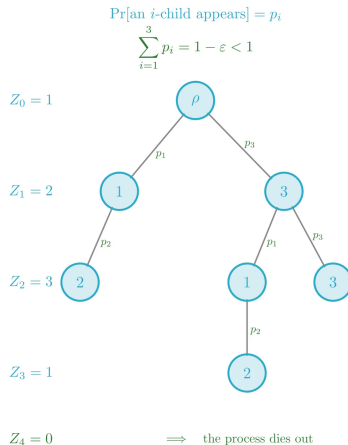


Quick detour: branching processes

A branching process $\mathcal{B}$ begins with a root. Every node independently creates child ℓ with probability p_ℓ .

Let Z_t be the number of nodes at depth t . If

$$\sum_{\ell \in N} p_\ell \leq 1 - \varepsilon,$$



Quick detour: branching processes

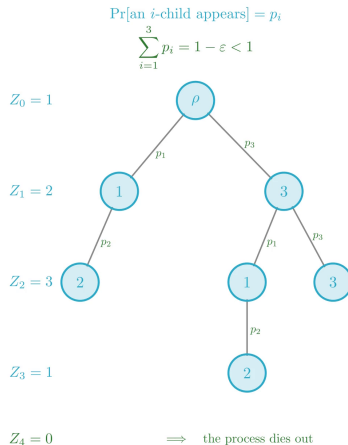
A branching process $\mathcal{B}$ begins with a root. Every node independently creates child ℓ with probability p_ℓ .

Let Z_t be the number of nodes at depth t . If

$$\sum_{\ell \in N} p_\ell \leq 1 - \varepsilon,$$

then

$$\begin{aligned} \mathbb{E}[Z_t \mid Z_{t-1}] &\leq (1 - \varepsilon)Z_{t-1}, \\ \implies \mathbb{E}[Z_t] &\leq (1 - \varepsilon)^t. \end{aligned}$$



Quick detour: branching processes

A branching process $\mathcal{B}$ begins with a root. Every node independently creates child ℓ with probability p_ℓ .

Let Z_t be the number of nodes at depth t . If

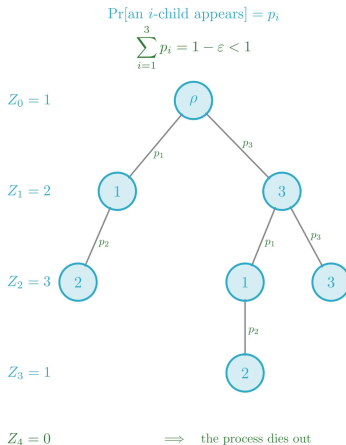
$$\sum_{\ell \in N} p_\ell \leq 1 - \varepsilon,$$

then

$$\begin{aligned} \mathbb{E}[Z_t \mid Z_{t-1}] &\leq (1 - \varepsilon)Z_{t-1}, \\ \implies \mathbb{E}[Z_t] &\leq (1 - \varepsilon)^t. \end{aligned}$$

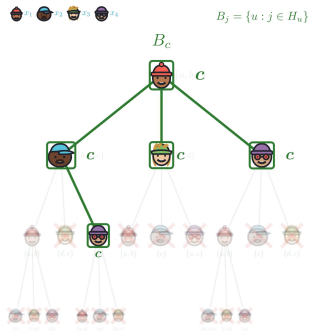
Therefore

$$\begin{aligned} \mathbb{E}[|V(\mathcal{B})|] &= \sum_{t \geq 0} \mathbb{E}[Z_t] \\ &\leq \sum_{t \geq 0} (1 - \varepsilon)^t = \frac{1}{\varepsilon}. \end{aligned}$$



Proof of termination

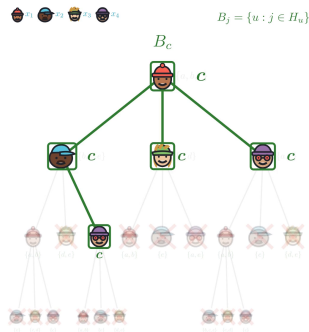
For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.



Proof of termination

For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.

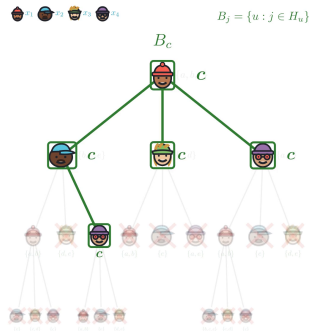
- **Obs 1.** Every internal node belongs to at least one B_j .



Proof of termination

For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.

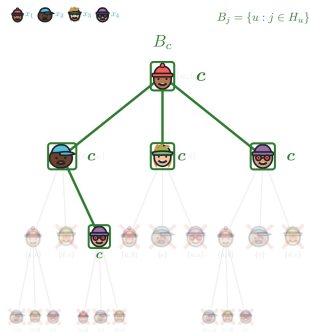
- **Obs 1.** Every internal node belongs to at least one B_j .
- Let $\kappa_{ij} = \Pr_{W \sim x_i}[j \in W] = \sum_{S \ni j} x_{i,S}$



Proof of termination

For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.

- **Obs 1.** Every internal node belongs to at least one B_j .
- Let $\kappa_{ij} = \Pr_{W \sim x_i}[j \in W] = \sum_{S \ni j} x_{i,S}$
- **Obs 2.** Each B_j is distributed as per a branching process with parameters $(\kappa_{1j}, \dots, \kappa_{nj})$.

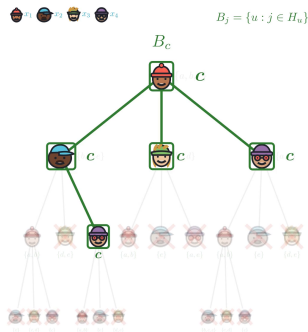


Proof of termination

For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.

- **Obs 1.** Every internal node belongs to at least one B_j .
- Let $\kappa_{ij} = \Pr_{W \sim x_i}[j \in W] = \sum_{S \ni j} x_{i,S}$
- **Obs 2.** Each B_j is distributed as per a branching process with parameters $(\kappa_{1j}, \dots, \kappa_{nj})$.
- Config LP constraint:

$$\sum_{i \in N} \sum_{S \ni j} x_{i,S} = \sum_{i \in N} \kappa_{ij} \leq 1$$

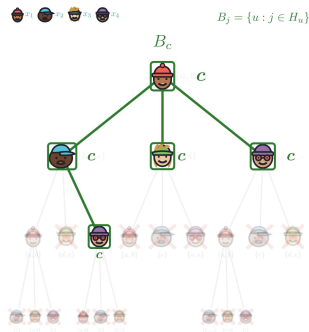


Proof of termination

For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.

- **Obs 1.** Every internal node belongs to at least one B_j .
- Let $\kappa_{ij} = \Pr_{W \sim x_i}[j \in W] = \sum_{S \ni j} x_{i,S}$
- **Obs 2.** Each B_j is distributed as per a branching process with parameters $(\kappa_{1j}, \dots, \kappa_{nj})$.
- Config LP constraint:

$$\sum_{i \in N} \sum_{S \ni j} x_{i,S} = \sum_{i \in N} \kappa_{ij} \leq 1 - \epsilon, \quad \forall j \in M$$



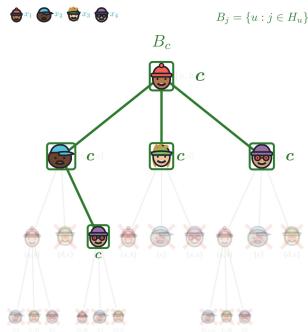
Proof of termination

For $j \in M$, let $B_j :=$ subtree containing nodes u for which every s in the root-to- u path satisfies $j \in W_s$.

- **Obs 1.** Every internal node belongs to at least one B_j .
- Let $\kappa_{ij} = \Pr_{W \sim x_i}[j \in W] = \sum_{S \ni j} x_{i,S}$
- **Obs 2.** Each B_j is distributed as per a branching process with parameters $(\kappa_{1j}, \dots, \kappa_{nj})$.
- Config LP constraint:

$$\sum_{i \in N} \sum_{S \ni j} x_{i,S} = \sum_{i \in N} \kappa_{ij} \leq 1 - \epsilon, \quad \forall j \in M$$

$$\text{Hence } \mathbb{E}[|V(B_j)|] \leq \frac{1}{\epsilon} \implies \mathbb{E}[|V(\mathcal{T})|] \leq \frac{nm}{\epsilon}$$



What have we achieved?

1. We looked at the offline setting

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.
5. This can be cast as a fixed-point: Existence shown by Brouwer's theorem.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.
5. This can be cast as a fixed-point: Existence shown by Brouwer's theorem.
6. Unrolling that fixed point yields a recursive algorithm.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.
5. This can be cast as a fixed-point: Existence shown by Brouwer's theorem.
6. Unrolling that fixed point yields a recursive algorithm.
7. Branching process style analysis shows that the algorithm terminates in polytime.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.
5. This can be cast as a fixed-point: Existence shown by Brouwer's theorem.
6. Unrolling that fixed point yields a recursive algorithm.
7. Branching process style analysis shows that the algorithm terminates in polytime.
8. Final allocation is a polytime $(2 + \varepsilon)$ -approximation to the configuration LP.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.
5. This can be cast as a fixed-point: Existence shown by Brouwer's theorem.
6. Unrolling that fixed point yields a recursive algorithm.
7. Branching process style analysis shows that the algorithm terminates in polytime.
8. Final allocation is a polytime $(2 + \varepsilon)$ -approximation to the configuration LP.

What have we achieved?

1. We looked at the offline setting
2. Simple auction: $b_i \sim \mathcal{D}_i$ for all $i \in N$, and

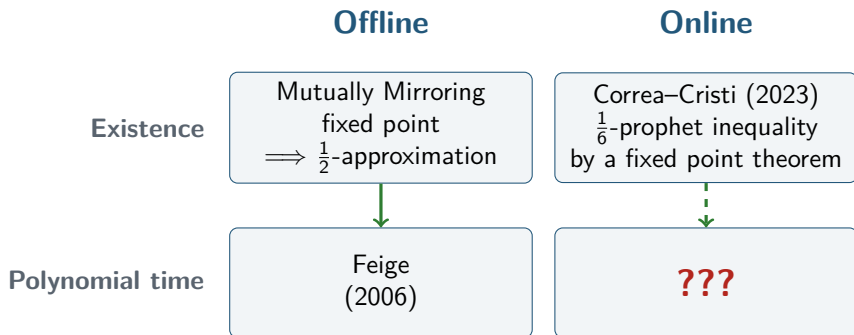
$$A_i = \{j \in M : b_{ij} > \max_{\ell \neq i} b_{\ell j}\}$$

3. Agent i plays according to mirroring strategy $\mathcal{M}_i(\mathcal{D}_{-i})$: gets $\frac{1}{2}$ of her LP value.
4. **Mutually Mirroring.** All players simultaneously mirror each other.
5. This can be cast as a fixed-point: Existence shown by Brouwer's theorem.
6. Unrolling that fixed point yields a recursive algorithm.
7. Branching process style analysis shows that the algorithm terminates in polytime.
8. Final allocation is a polytime $(2 + \varepsilon)$ -approximation to the configuration LP.

But this offline result was already known

Feige (2006) shows how to round the configuration LP with an approximation of $2 + \epsilon$.

The emerging picture



Recall the online setting

- ▶ Each agent i has a known distribution $\mathcal{F}_i$ over monotone subadditive valuations.

Recall the online setting

- ▶ Each agent i has a known distribution $\mathcal{F}_i$ over monotone subadditive valuations.
- ▶ Agents arrive in adversarial order.

Recall the online setting

- ▶ Each agent i has a known distribution $\mathcal{F}_i$ over monotone subadditive valuations.
- ▶ Agents arrive in adversarial order.
- ▶ On arrival, $v_i \sim \mathcal{F}_i$ is realized and observed.

Recall the online setting

- ▶ Each agent i has a known distribution $\mathcal{F}_i$ over monotone subadditive valuations.
- ▶ Agents arrive in adversarial order.
- ▶ On arrival, $v_i \sim \mathcal{F}_i$ is realized and observed.
- ▶ The allocation A_i must be chosen irrevocably from the remaining items.

Recall the online setting

- ▶ Each agent i has a known distribution $\mathcal{F}_i$ over monotone subadditive valuations.
- ▶ Agents arrive in adversarial order.
- ▶ On arrival, $v_i \sim \mathcal{F}_i$ is realized and observed.
- ▶ The allocation A_i must be chosen irrevocably from the remaining items.

Recall the online setting

- ▶ Each agent i has a known distribution $\mathcal{F}_i$ over monotone subadditive valuations.
- ▶ Agents arrive in adversarial order.
- ▶ On arrival, $v_i \sim \mathcal{F}_i$ is realized and observed.
- ▶ The allocation A_i must be chosen irrevocably from the remaining items.

Prophet benchmark

Compare the online welfare with the expected offline optimum:

$$\mathbb{E}_{\mathbf{v} \sim \mathcal{F}} \left[\sum_i v_i(A_i) \right] \quad \text{versus} \quad \mathbb{E}_{\mathbf{v} \sim \mathcal{F}} \left[\sum_i v_i(\text{OPT}_i(\mathbf{v})) \right].$$

The Correa–Cristi online auction

Now, agent's strategy is a **function** from possible valuations to distributions over bids.

$$\mathcal{D}_i : \text{supp}(\mathcal{F}_i) \longrightarrow \Delta(\mathbb{Z}_+^M).$$

The Correa–Cristi online auction

Now, agent's strategy is a **function** from possible valuations to distributions over bids.

$$\mathcal{D}_i : \text{supp}(\mathcal{F}_i) \longrightarrow \Delta(\mathbb{Z}_+^M).$$

Correa-Cristi Online Auction

1. Before arrivals, independently sample

$$v'_\ell \sim \mathcal{F}_\ell, \quad b'_\ell \sim \mathcal{D}_\ell(v'_\ell) \quad \forall \ell \in N.$$

The Correa–Cristi online auction

Now, agent's strategy is a **function** from possible valuations to distributions over bids.

$$\mathcal{D}_i : \text{supp}(\mathcal{F}_i) \longrightarrow \Delta(\mathbb{Z}_+^M).$$

Correa-Cristi Online Auction

1. Before arrivals, independently sample

$$v'_\ell \sim \mathcal{F}_\ell, \quad b'_\ell \sim \mathcal{D}_\ell(v'_\ell) \quad \forall \ell \in N.$$

2. When i arrives, observe v_i and draw $b_i \sim \mathcal{D}_i(v_i)$.

The Correa–Cristi online auction

Now, agent's strategy is a **function** from possible valuations to distributions over bids.

$$\mathcal{D}_i : \text{supp}(\mathcal{F}_i) \longrightarrow \Delta(\mathbb{Z}_+^M).$$

Correa-Cristi Online Auction

1. Before arrivals, independently sample

$$v'_\ell \sim \mathcal{F}_\ell, \quad b'_\ell \sim \mathcal{D}_\ell(v'_\ell) \quad \forall \ell \in N.$$

2. When i arrives, observe v_i and draw $b_i \sim \mathcal{D}_i(v_i)$.
3. Allocate

$$A_i = \left\{ j \in M : j \text{ is available, and } b_{ij} > \max_{\ell \in N} b'_{\ell j} \right\}.$$

Online mirroring

Fix the realized valuation v_i . To sample from $\mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i)$:

Online mirroring

Fix the realized valuation v_i . To sample from $\mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i)$:

1. Sample an interest set W from an appropriate LP.

Online mirroring

Fix the realized valuation v_i . To sample from $\mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i)$:

1. Sample an interest set W from an appropriate LP.
2. Sample $\hat{v}_\ell \sim \mathcal{F}_\ell$, and $\hat{b}_\ell \sim \mathcal{D}_\ell(\hat{v}_\ell)$ for all $\ell \in N$.

Online mirroring

Fix the realized valuation v_i . To sample from $\mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i)$:

1. Sample an interest set W from an appropriate LP.
2. Sample $\hat{v}_\ell \sim \mathcal{F}_\ell$, and $\hat{b}_\ell \sim \mathcal{D}_\ell(\hat{v}_\ell)$ for all $\ell \in N$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \in N} \hat{b}_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Online mirroring

Fix the realized valuation v_i . To sample from $\mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i)$:

1. Sample an interest set W from an appropriate LP.
2. Sample $\hat{v}_\ell \sim \mathcal{F}_\ell$, and $\hat{b}_\ell \sim \mathcal{D}_\ell(\hat{v}_\ell)$ for all $\ell \in N$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \in N} \hat{b}_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Online mirroring

Fix the realized valuation v_i . To sample from $\mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i)$:

1. Sample an interest set W from an appropriate LP.
2. Sample $\hat{v}_\ell \sim \mathcal{F}_\ell$, and $\hat{b}_\ell \sim \mathcal{D}_\ell(\hat{v}_\ell)$ for all $\ell \in N$.
3. Return

$$b_{ij} = \begin{cases} 1 + \max_{\ell \in N} \hat{b}_{\ell j}, & j \in W, \\ 0, & j \notin W. \end{cases}$$

Lemma

An agent using this mirroring strategy receives at least $\frac{1}{6}$ of her LP value.

Mutually mirroring online strategies

Online strategies are mutually mirroring if

$$\mathcal{D}_i(v_i) \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i) \quad \forall i, \forall v_i \in \text{supp}(\mathcal{F}_i).$$

Mutually mirroring online strategies

Online strategies are mutually mirroring if

$$\mathcal{D}_i(v_i) \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i) \quad \forall i, \forall v_i \in \text{supp}(\mathcal{F}_i).$$

Welfare guarantee

If all players use mutually mirroring strategies, then

$$\mathbb{E} \left[\sum_i v_i(A_i) \right] \geq \frac{1 - \varepsilon}{6} \mathbb{E} \left[\sum_i v_i(\text{OPT}_i(\mathbf{v})) \right].$$

Mutually mirroring online strategies

Online strategies are mutually mirroring if

$$\mathcal{D}_i(v_i) \stackrel{d}{=} \mathcal{M}_i(\mathcal{D}_1, \dots, \mathcal{D}_n)(v_i) \quad \forall i, \forall v_i \in \text{supp}(\mathcal{F}_i).$$

Welfare guarantee

If all players use mutually mirroring strategies, then

$$\mathbb{E} \left[\sum_i v_i(A_i) \right] \geq \frac{1 - \varepsilon}{6} \mathbb{E} \left[\sum_i v_i(\text{OPT}_i(\mathbf{v})) \right].$$

Efficient sampling

The online fixed point can also be unrolled. This results in a recursive sampling procedure that makes at most nm/ε recursive calls in expectation.

Open directions

1. Can we obtain posted prices?

A posted-price mechanism fixes item prices p . When agent i arrives, she gets

$$A_i = \operatorname{argmax} \left\{ v_i(S) - \sum_{j \in S} p_j \right\}$$

Open directions

1. Can we obtain posted prices?

A posted-price mechanism fixes item prices p . When agent i arrives, she gets

$$A_i = \operatorname{argmax} \left\{ v_i(S) - \sum_{j \in S} p_j \right\}$$

2. What are the limits of mirroring?

Mirroring handles both offline and online combinatorial auctions. Could it apply to other problems with subadditive objectives? What about general setting where one has multiple copies of each item?

Open directions

1. Can we obtain posted prices?

A posted-price mechanism fixes item prices p . When agent i arrives, she gets

$$A_i = \operatorname{argmax} \left\{ v_i(S) - \sum_{j \in S} p_j \right\}$$

2. What are the limits of mirroring?

Mirroring handles both offline and online combinatorial auctions. Could it apply to other problems with subadditive objectives? What about general setting where one has multiple copies of each item?

3. A general framework for subadditivity?

submodular	$\implies$	greedy methods
XOS	$\implies$	supporting additive functions
subadditive	$\implies$	mirroring?

Thank you!

Questions?