Σχεδιασμός Μηχανισμών για Συνδυαστικές Δημοπρασίες με Αξιοποίηση Προβλέψεων Μηχανικής Μάθησης

Παρουσίαση Διπλωματικής Εργασίας

Θεόδωρος Τσιλιβής

9/9/2022

How do we sort an array?

Should we use Mergesort or Quicksort?

1. Theoretical Computer Science says Mergesort (better complexity).
2. In practice we tend to use Quicksort.

What are we missing?

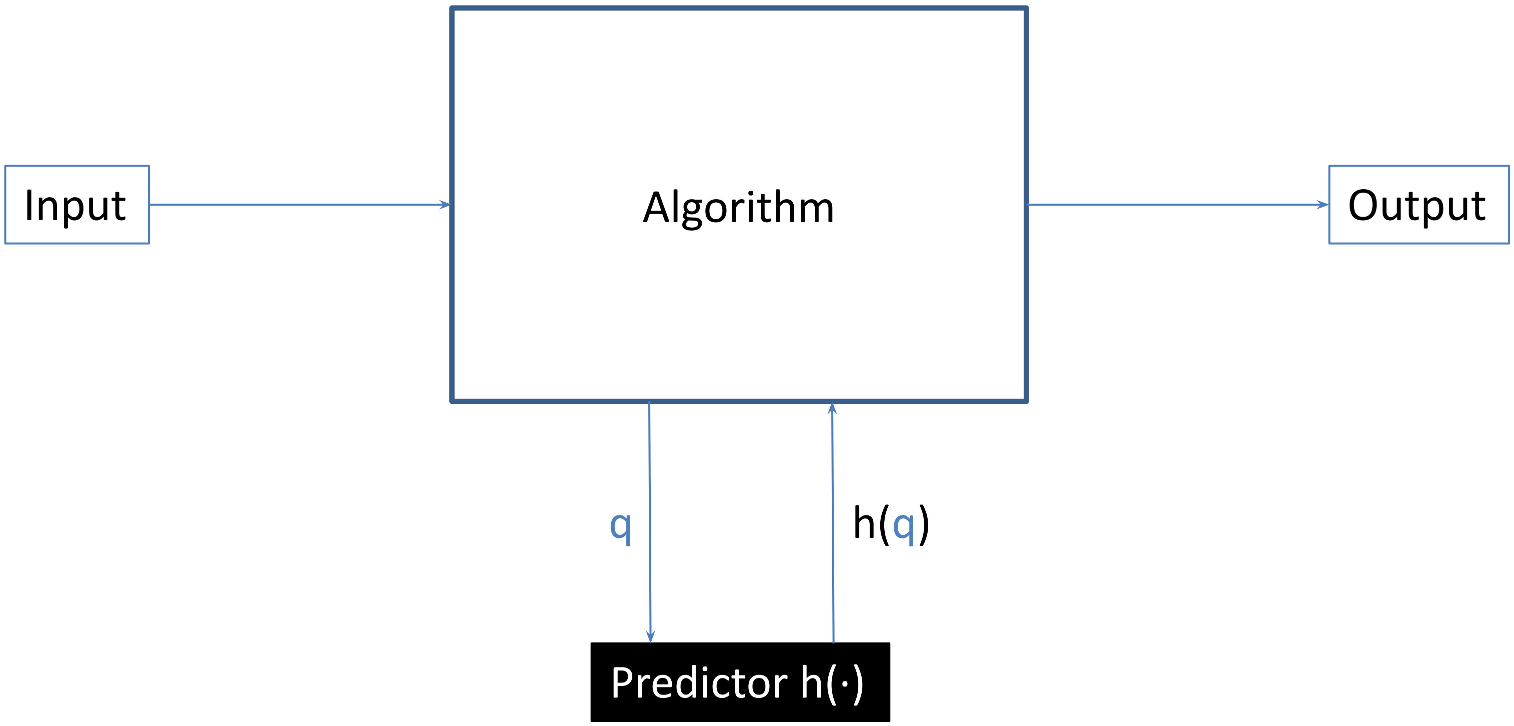
- Some easy probabilistic ideas.
- Worst Case analysis.

Beyond Worst Case

Worst case analysis is always pessimistic. Relaxing it results in:

1. Parameterized Algorithms
2. Smoothed analysis.
3. Algorithms with predictions.
4. More.

Algorithms with predictions



What are we interested in?

-
- Find the appropriate prediction!
- Quantify the error (L_1, L_2, L_∞, something else.).

The usual metrics are:

- **Consistency** = How well does the algorithm work if the prediction is perfect.
- **Robustness** = How well does the algorithm work with the worst (?) possible prediction.

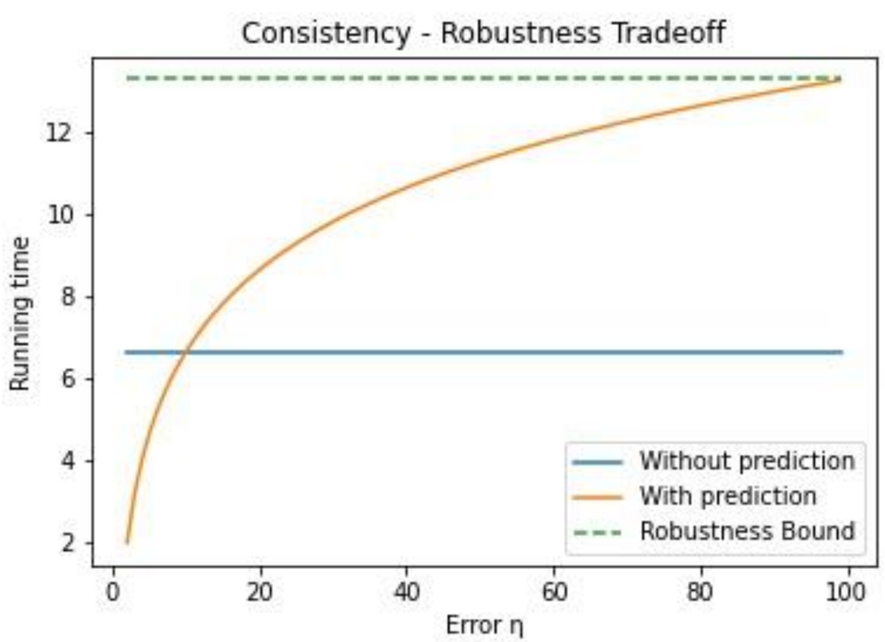
Definition 3.0.1 (consistency). *An algorithm with a prediction black box $h(\cdot)$ that observes error η is b-consistent for some error function $a(\eta)$ if it admits an approximation ratio of $O(b)$, where $b = a(0)$*

Definition 3.0.2 (robustness). *An algorithm with a prediction black box $h(\cdot)$ that observes error η is a-robust for some error function $a(\eta)$ if it admits an approximation ratio of $O\left(\max_{\eta} a(\eta)\right)$.*

Consistency – Robustness Tradeoff

-
- **Learning augmented** Binary search has running time $2\ log\ \eta$.

- **Classic** Binary search has running time $log\ n$, where n is the size of the array.



Combinatorial Auctions – The problem

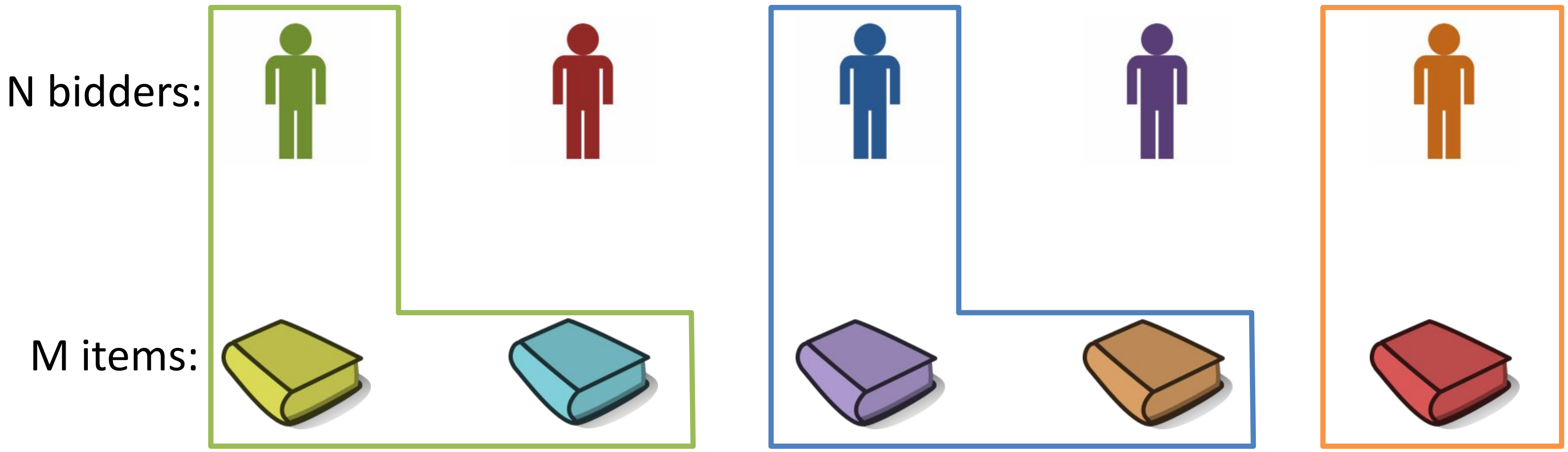
N bidders:

M items:

One objective – The Social Welfare

- The objective is to calculate an **allocation** of the items $A = (A_1, \ldots, A_n)$ that maximizes the **social welfare**.
- Players have **valuation functions** v_i over the possible subsets of M.

Social Welfare = $\sum_{i \in N} v_i(A_i)$



In this case the allocation is $A = (\{1,2\}, \emptyset, \{3,4\}, \emptyset, \{5\})$

Valuations – Bidder 1

N bidders:

M items:

v {1} = 2 v {2} = 3 v {3} = 5 v {4} = 2 v {5} = 2

Valuations – Bidder 1

N bidders:

M items:

v {1} = 2 v {2} = 3 v {3} = 5 v {4} = 2 v {5} = 2

v {1,2} = 5

Valuations – Bidder 1

N bidders:

M items:

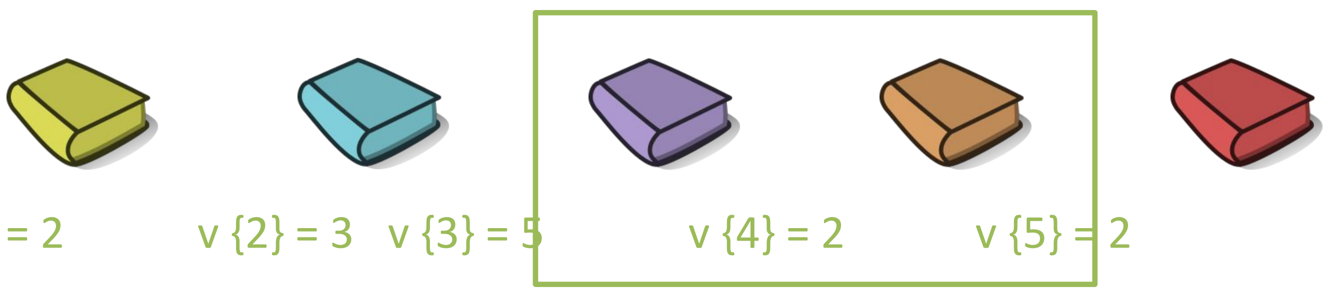
v {1} = 2 v {2} = 3 v {3} = 5 v {4} = 2 v {5} = 2

v {1,2} = 5 **v {2,3} = 6**

Valuations – Bidder 1

N bidders:

M items:



v {1} = 2   v {2} = 3   v {3} = 5   v {4} = 2   v {5} = 2

v {1,2} = 5   v {2,3} = 8   **v {3,4} = 5**

Valuations – Bidder 1

N bidders:

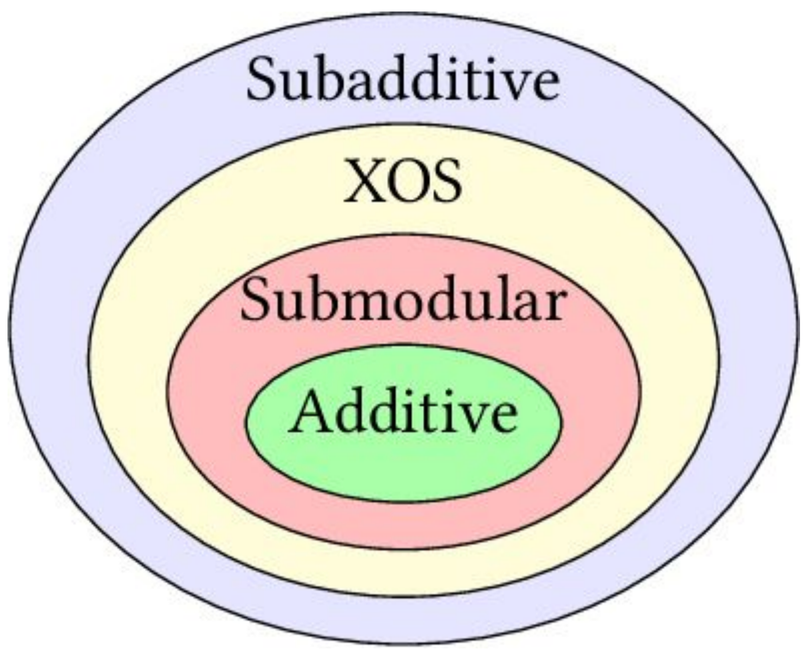
M items:

v {1} = 2   v {2} = 3   v {3} = 5   v {4} = 2   v {5} = 2

v {1,2} = 5   v {2,3} = 8   v {3,4} = 6   **v {4,5} = 3**

Valuation function classes

- **Additive:** $v(S) + v(T) = v(S \cup T) + v(S \cap T)\ \forall (S, T)$
- **Submodular:** $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)\ \forall (S, T)$
- **Subadditive:** $v(S) + v(T) \geq v(S \cup T)\ \forall (S, T)$



Accessing valuation functions

•

- **Valuation functions** are functions v_i defined from 2^M to R^+.
- That means the input is exponentially large.
- **Value queries**: Presented a bundle S bidder i outputs his valuation of the bundle $v_i(S)$
- **Demand queries**: Presented a price vector **p** bidder i outputs the bundle S' that maximizes his utility, that is:

$$S' \in \arg\max_{S \subseteq M} \{v_i(S) - p(S)\}$$

Value queries:

$v_i(\{1,2,3\}) = 12$

Demand queries:

With these prices
I want S = {2,3}

p {1} = 9 p {2} = 3 p {3} = 5

Utility – Revenue and Social Welfare

Given a price vector **p** and an allocation $A = (A_1, \ldots, A_n)$ we can define:

- $Utility(i) = v_i(A_i) - p(A_i)$
- $Rev(A_i) = \sum_{e \in A_i} p(e) = p(A_i)$
- **Social Welfare** = $\sum_{i \in N} v_i(A_i) = \sum_{i \in N}(Utility(i) + Rev(A_i))$

Algorithmic Viewpoint

- Can we optimally solve the problem knowing that bidders do not misreport?
- The problem for general valuations is **NP-hard** (set packing).
- There are constant approximation algorithms for the algorithmic problem for many valuation function classes.

Papers\Valuations	Submodular Approximation ratio	Subadditive Approximation ratio
B. Lehmann et al., 2006	2	
Jan Vondrák, 2006		
Dobzinski et al., 2005		
Feige and Vondrák 2006		2

Algorithmic vs Game Theoretic version

- **NP-Hard** for Submodular and greater classes (set packing reduction).
- **Constant** factor **approximations** for the Submodular $\left(\frac{e}{e-1}\right)$ and Subadditive (2) algorithmic problem.
- **Truthfulness** makes the problem much more difficult to approximate.

Papers\Valuations	Submodular Approximation ratio	Subadditive Approximation ratio
Dobzinski et al., 2006		
P. Krysta and B. Vöcking, 2012		
Dobzinski, 2016		
S. Assadi, S. Singla, 2019		
Dobzinski et al., 2005		
Dobzinski, 2007		
S. Assadi, T. Kesselheim, S. Singla, 2021		

Sampling and Randomization

- Bidder i is a **Dominant bidder** if $v_i(M) \geq \frac{OPT}{a}$.

- Instances **with** a dominant bidder we run a Second-Price Auction on the whole bundle M and get an a-approximation.

- Design mechanisms for instances **without** dominant bidders. Use **Sampling**.

- A mechanism **MECH** for instances without dominant bidders can be equiped on a randomized mechanism that **flips a coin** and decides whether to run the **Second-Price Auction** or **MECH**.

$$E[Welfare] = \frac{1}{2}E[Welfare \mid MECH] + \frac{1}{2}E[Welfare \mid Second\ Price\ Auction]$$

Price vectors – Learning procedures

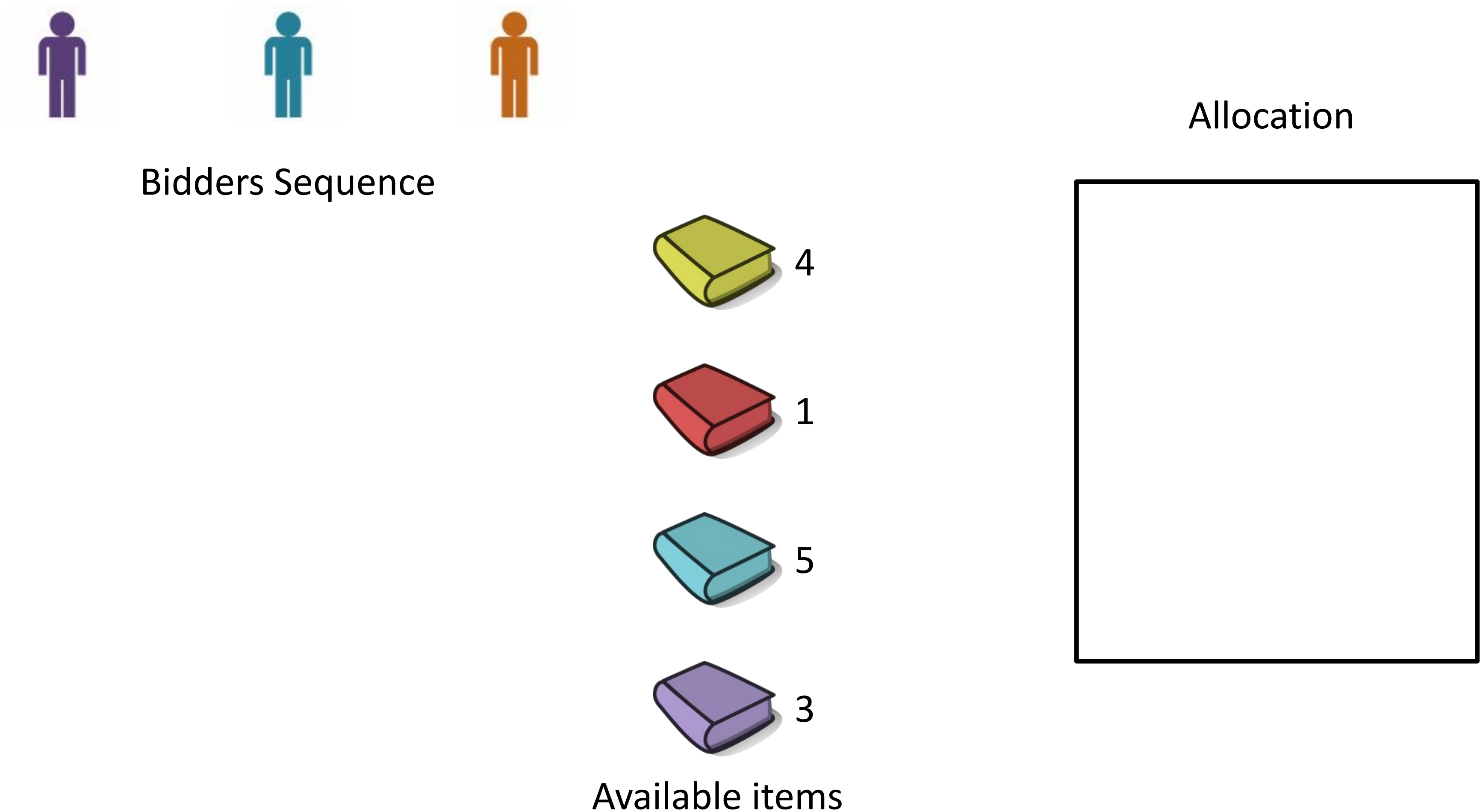
- **Demand queries** require a price vector ***p.***
- **Truthfulness** requires that no bidder can **affect** her utility **by misreporting** her valuation.
- We need to learn from bidders better prices ***p*** but discourage bidders from taking advantage. How?
- We design mechanisms that work the problem **online**.
- **Bidders** come in **once**, answer their query, get allocated something (sometimes) and **leave**. This way we ensure learning and also truthfulness.
- What can we do with **good estimates** of the prices?

Fixed-Price Auction

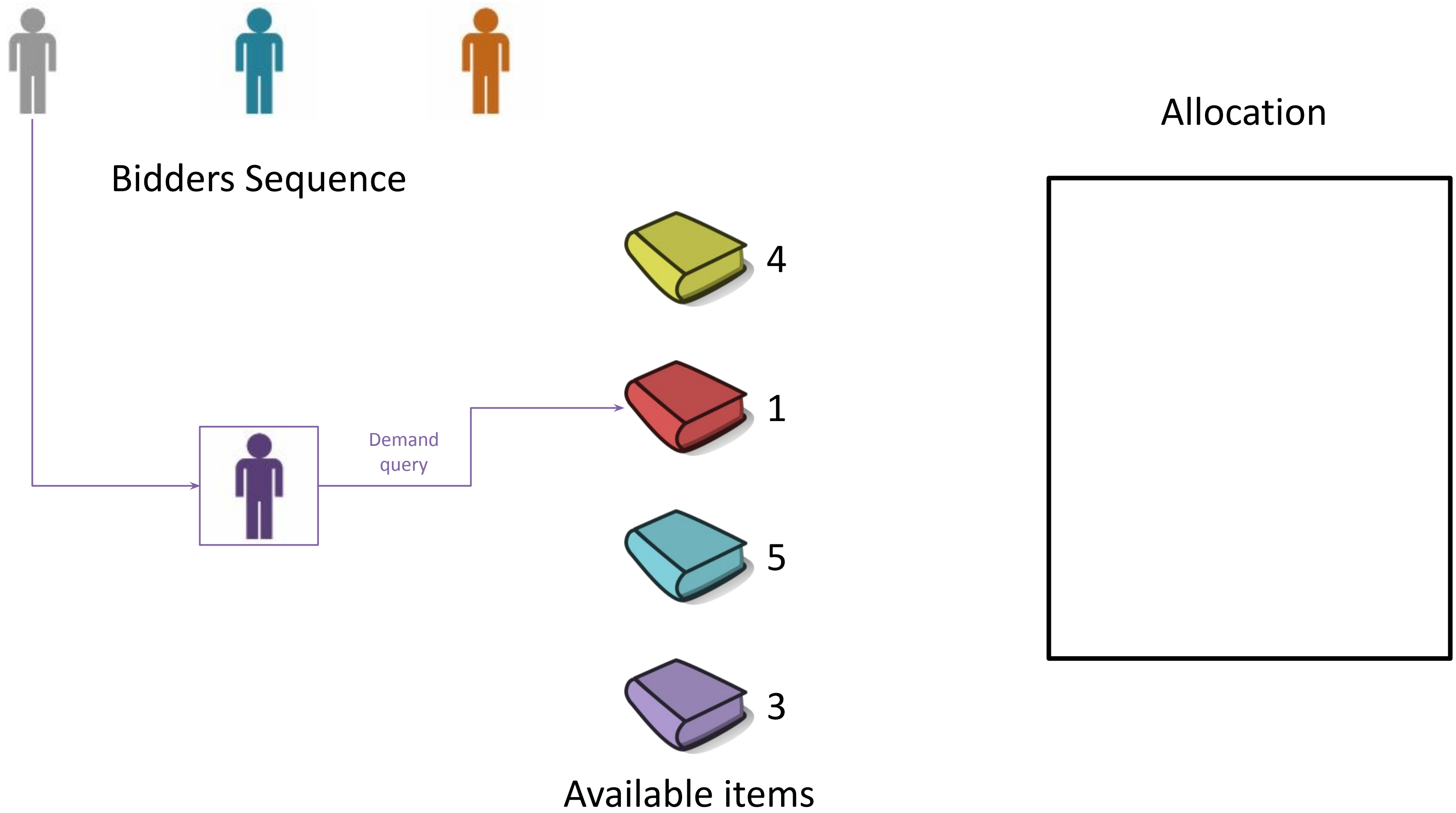
Mechanism 2: Fixed-Price auction $(\mathbf{p}, M, N)$

Input: A price vector $\mathbf{p}$, a set of items M, an ordered set of bidders N
Output: An Allocation $\mathbf{A} = (A_1, ..., A_n)$
for $i \in N$ **do**
  Suppose S_i is bidder i response to the demand query with items M and price vector $\mathbf{p}$.
  $A_i \leftarrow S_i$
  $M \leftarrow M \setminus A_i$
end
return $\boldsymbol{A} = (A_1, ..., A_n)$

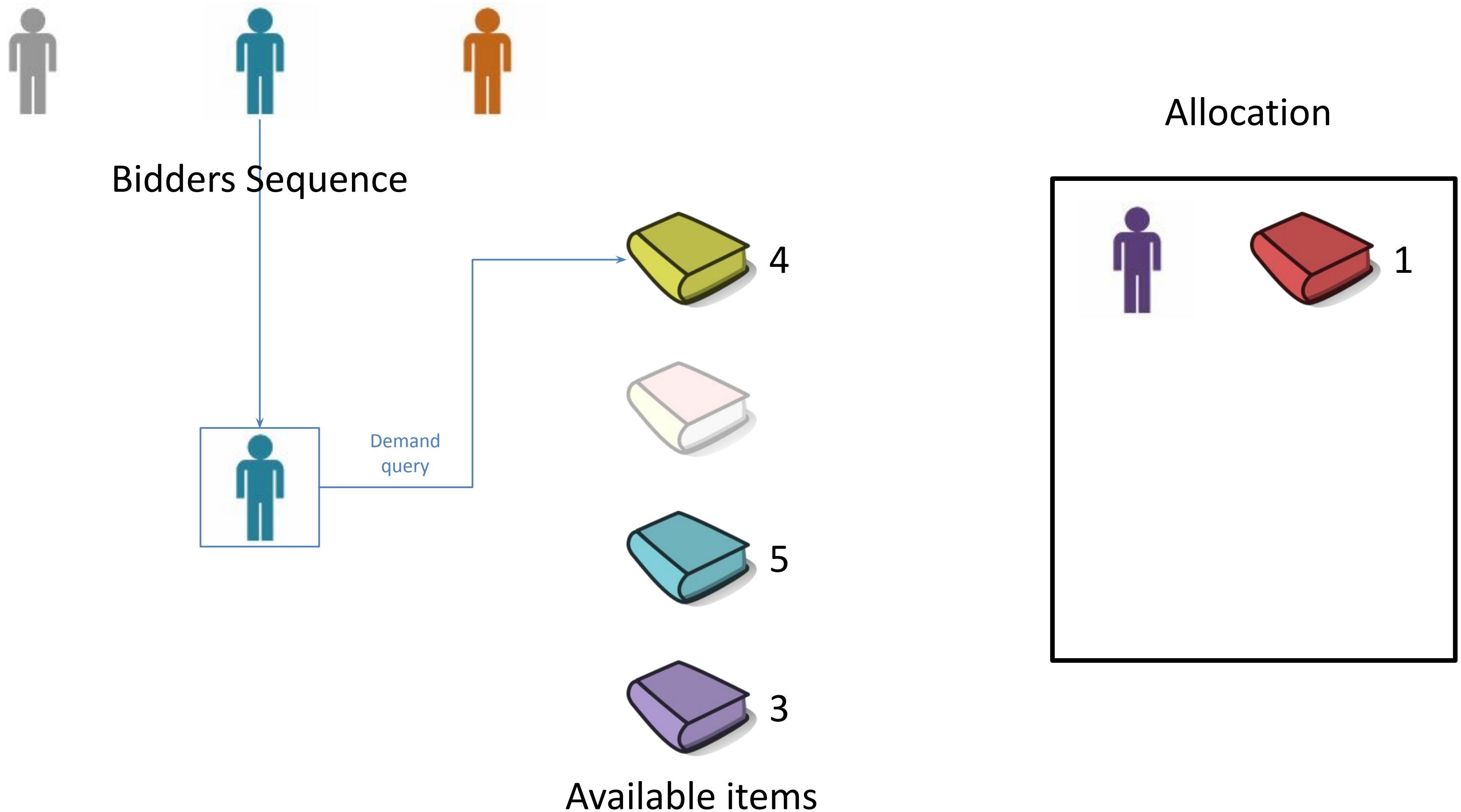
Fixed-Price Auction



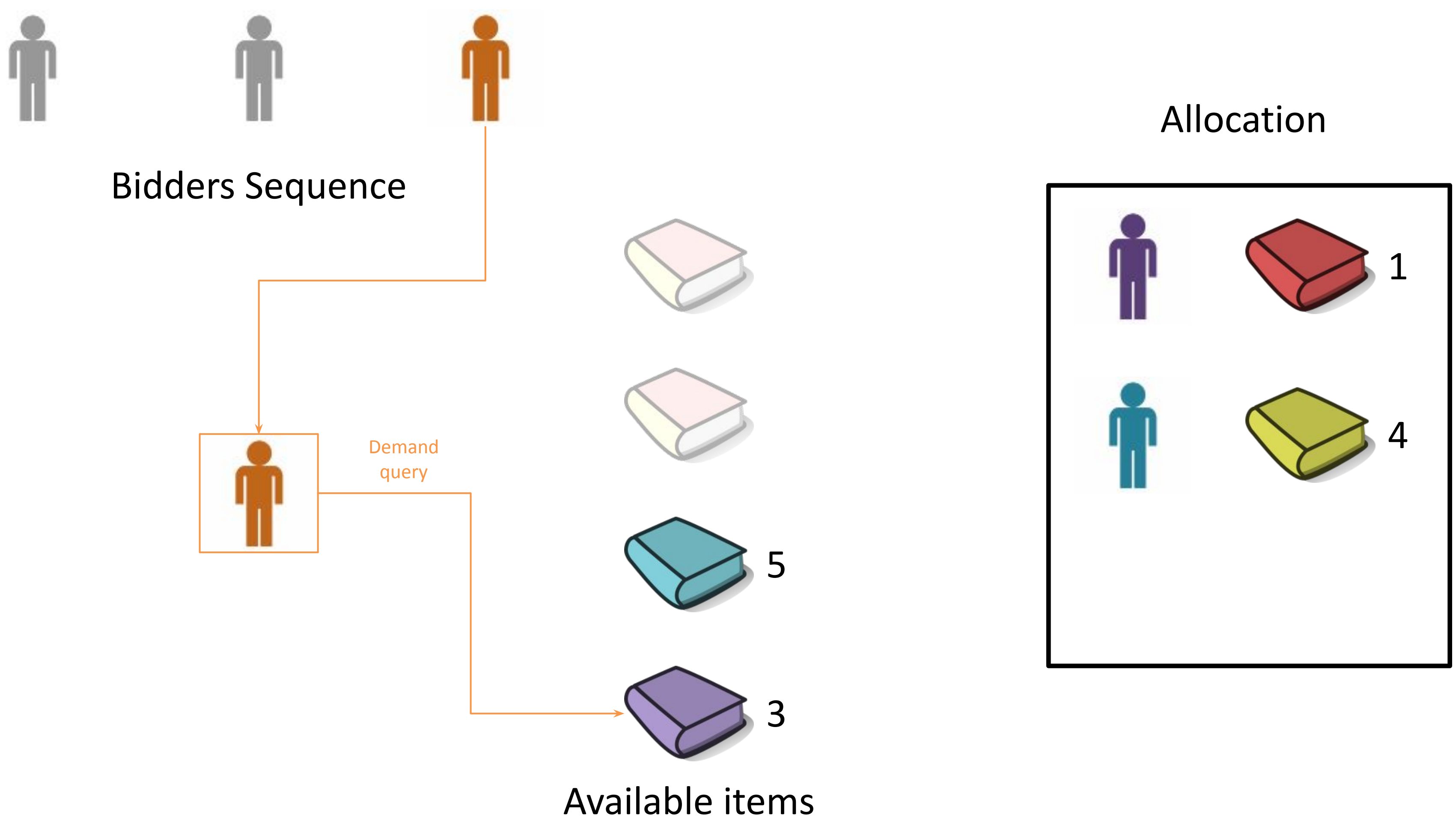
Round 1


Allocation
Bidders Sequence
4
Demand
query
1
5
3
Available items

Round 2

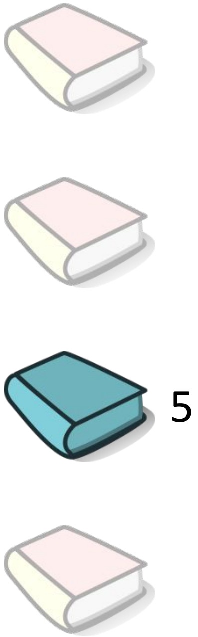

Allocation
Bidders Sequence
4
1
Demand
query
5
3
Available items

Round 3


Allocation
Bidders Sequence
1
4
Demand
query
5
3
Available items

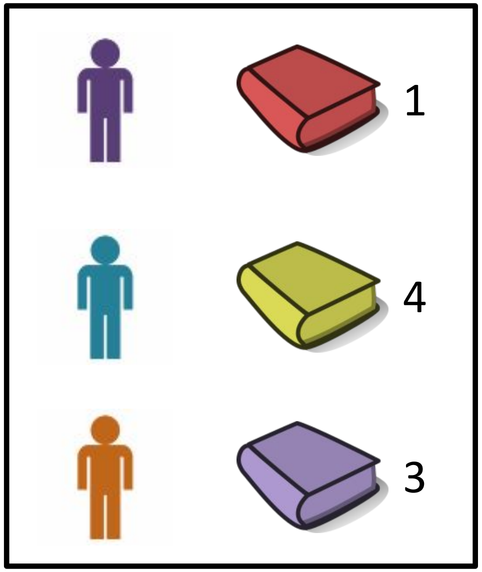
End

Bidders Sequence



Available items

Allocation



Combinatorial Auctions with predictions

•

Social Welfare = $\sum_{i \in N} v_i(A_i) \geq 0$

State of the art mechanism **S** is an $O(a)$-approximation.

Design an $O(1)$-consistent approximation mechanism **MECH**.

A **randomized** mechanism that flips a coin and runs either **S** or **MECH** is:

- $O(1)$-consistent
- $O(a)$-robust.

$$E[Welfare] = \frac{1}{2} E[Welfare \,|MECH] + \frac{1}{2} E[Welfare \,|S]$$

Combinatorial Auctions with predictions

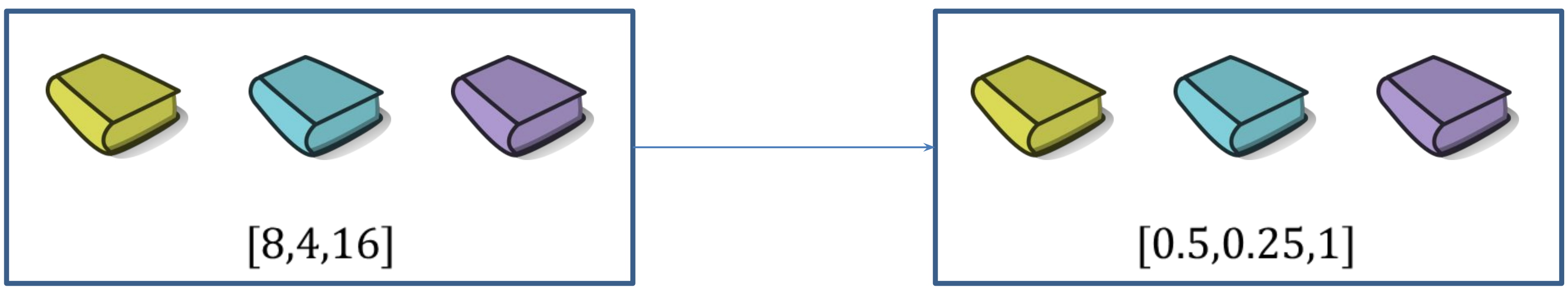
- The prediction is a price vector $\boldsymbol{p}$.

4-criteria of interest:

- Proportionality → Is the information given in a proportional form?
- Richness of Prices → Guarantees of the price vector.
- Consistency → Is the price vector erroneous?
- Mechanism Class → What type of mechanisms will we be using?

Proportionality

Definition 5.1.1 (proportional price vector). *We define as proportional price vector* $\mathbf{p}$ *the price vector that results from a point-wise multiplication with a constant* c *that places all items* j *at prices* p_j *such that* $p_j \in (0, 1]$.



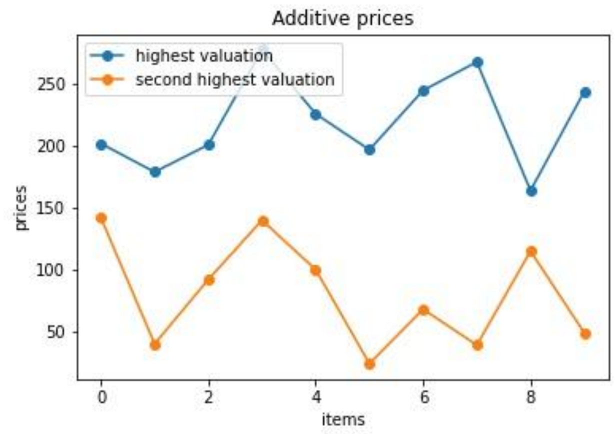
price vector with some property

corresponding <u>proportional</u> price vector

Richness of prices

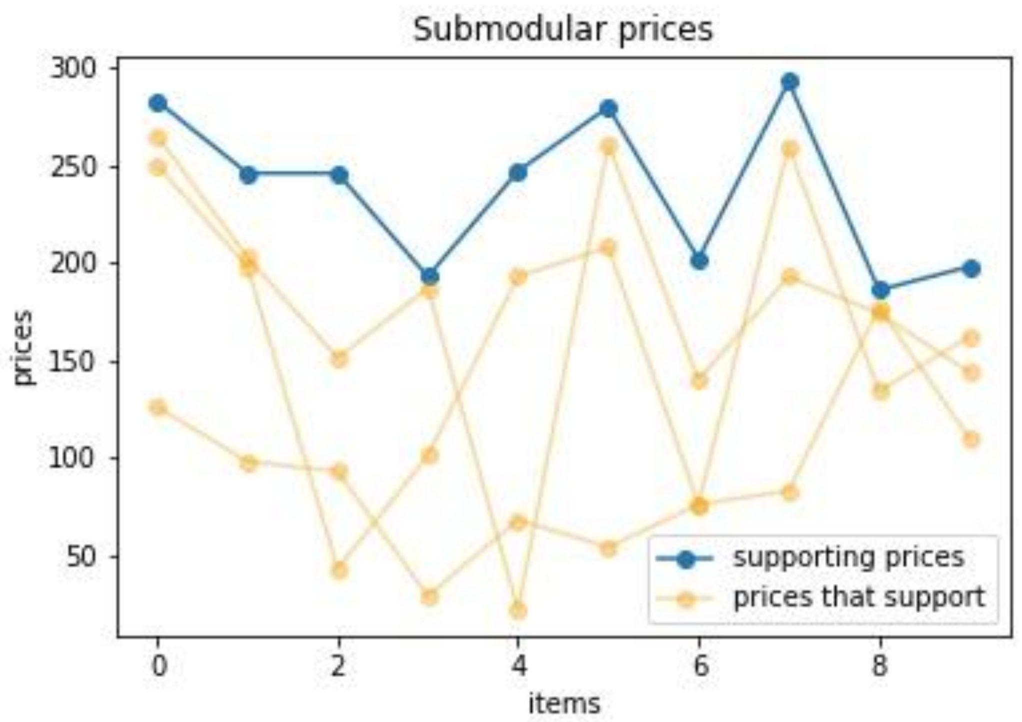
Additive:

- **Optimal prices** p_j.



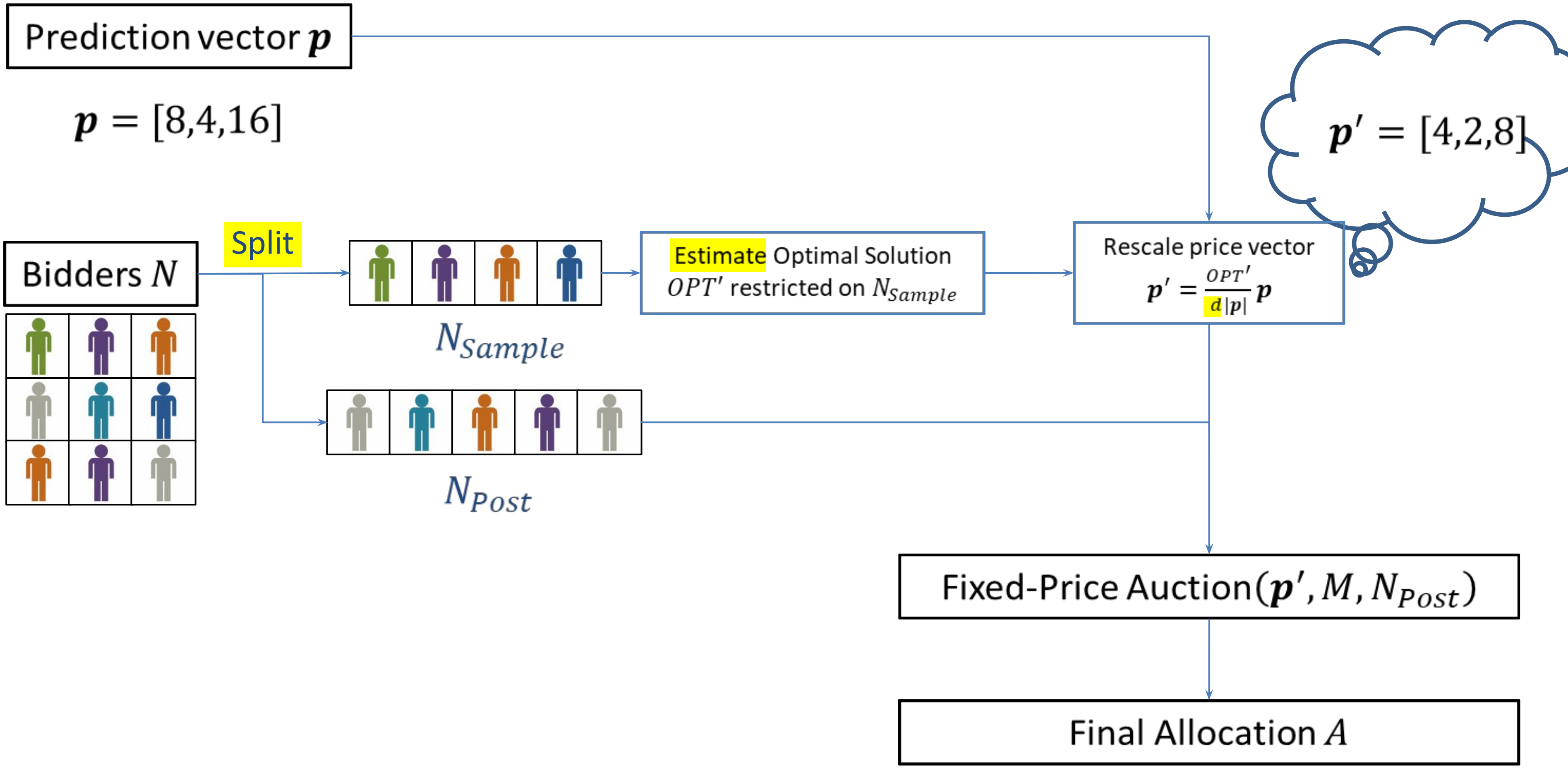
Submodular:

- **Supporting prices** q_j of an allocation $A = (A_1, \dots, A_n)$.
  - If bidder i gets bundle A_i then $\sum_{j \in A_i} q_j = v_i(A_i)$.
- **Prices** p_j that **support** the allocation $A = (A_1, \dots, A_n)$ are prices such that $p_j \leq q_j, \ \forall j \in M$.

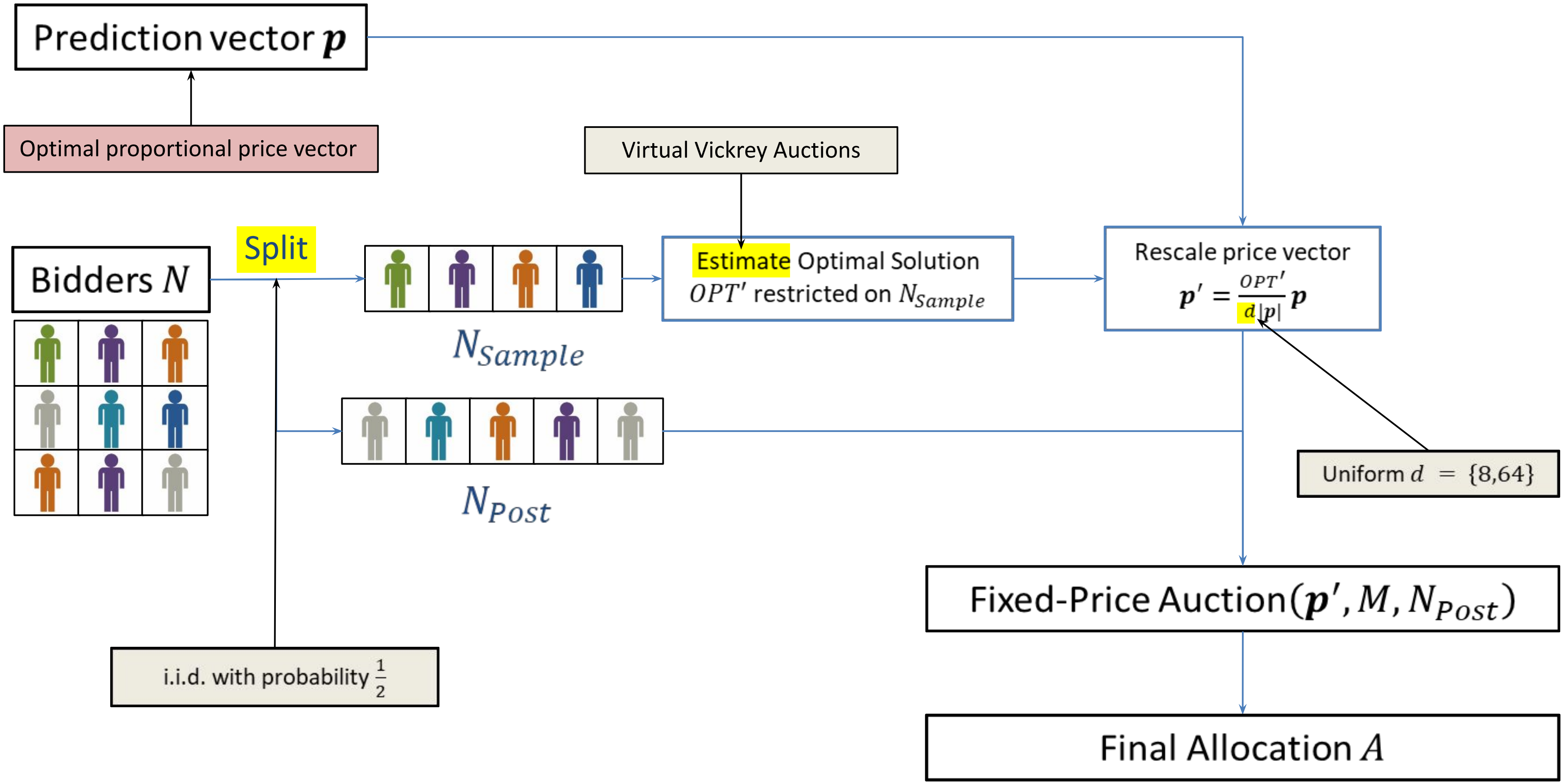


Lemma 4.2.1. *[17] For an allocation* $O = (O_1, ..., O_n)$ *and a price vector* $\boldsymbol{p} = (p_1, ..., p_m)$ *that supports this allocation, a fixed price auction using prices* $\frac{\boldsymbol{p}}{2} = \left(\frac{p_1}{2}, ..., \frac{p_m}{2}\right)$ *comes up with an allocation* $A = (A_1, ..., A_n)$ *that has* $\sum_i v_i(A_i) \geq \frac{\sum_{j \in O} p_j}{2}$.

Sampling & Multiplying Mechanisms



Additive Class - The Sampling Mechanism



The Sampling Mechanism

Mechanism 3: The Sampling Mechanism

Input: A proportional price vector $\mathbf{p}$
Output: An Allocation $\mathbf{A} = (A_1, ..., A_n)$
Split bidders randomly with probability $\frac{1}{2}$ into groups N_1, N_2.
Randomly assign N_1 and N_2 to N_{Sample} and N_{Post}.
Run a 'virtual' second price auction for each item $j \in M$ only on bidders $i \in N_{Sample}$ and output the corresponding Optimal Social Welfare as OPT'.
Randomly set $d = \{64, 8\}$ and define $\mathbf{p}' = \frac{OPT'}{d \cdot |\mathbf{p}|} \cdot \mathbf{p}$.
Run the FPA(N_{Post},M,$\mathbf{p}'$) and output its allocation $\mathbf{A} = (A_1, ..., A_n)$ as the final allocation of the Mechanism.

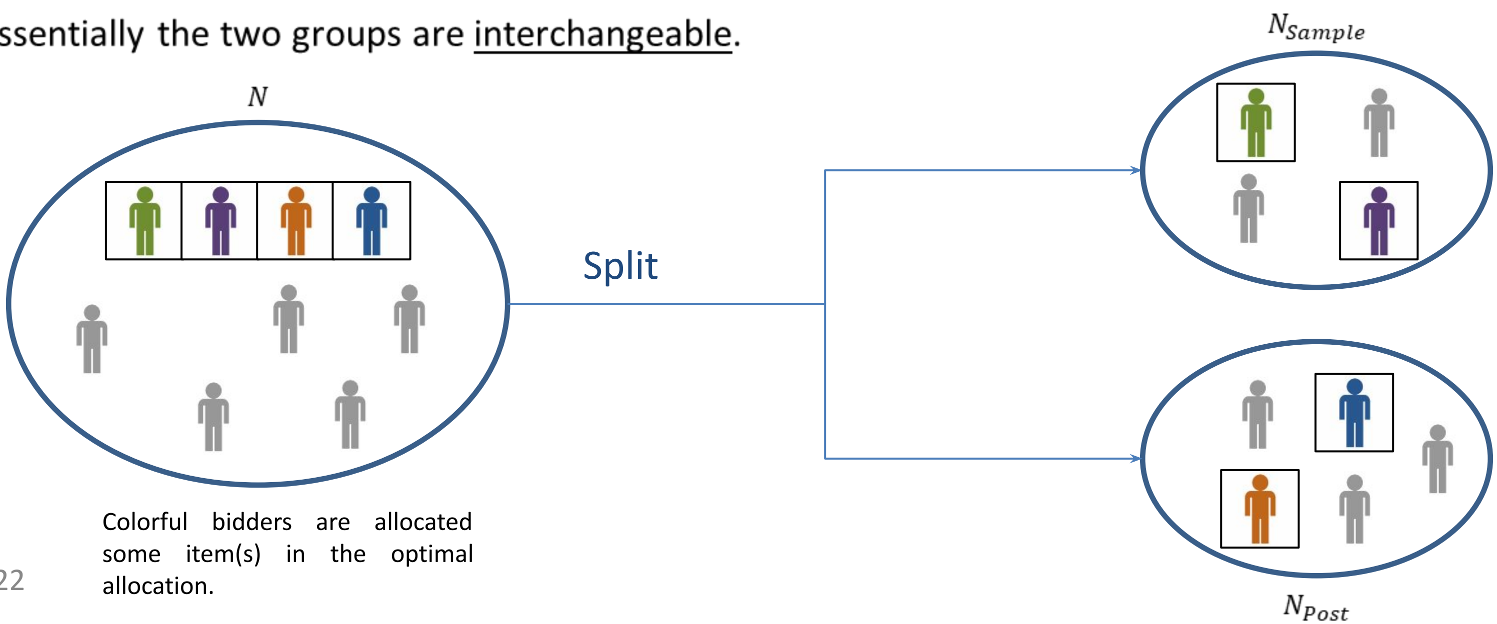
The power of i.i.d. splitting

- We can prove that for either group $N_k = \{N_{Sample}, N_{Post}\}$ with **high probability** it holds that:

$$OPT_{N_k,} \in \left(\frac{OPT}{4}, \frac{3OPT}{4}\right)$$

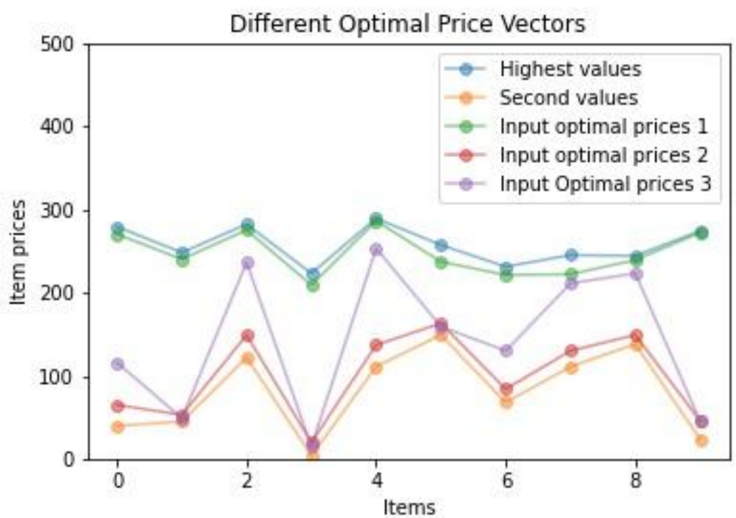
- Optimal allocation $O = (O_1, \dots, O_n)$.
- $OPT = \sum_{i \in N} v_i(O_i)$.
- $OPT_{N_k} = \sum_{i \in N_k} v_i(O_i)$.

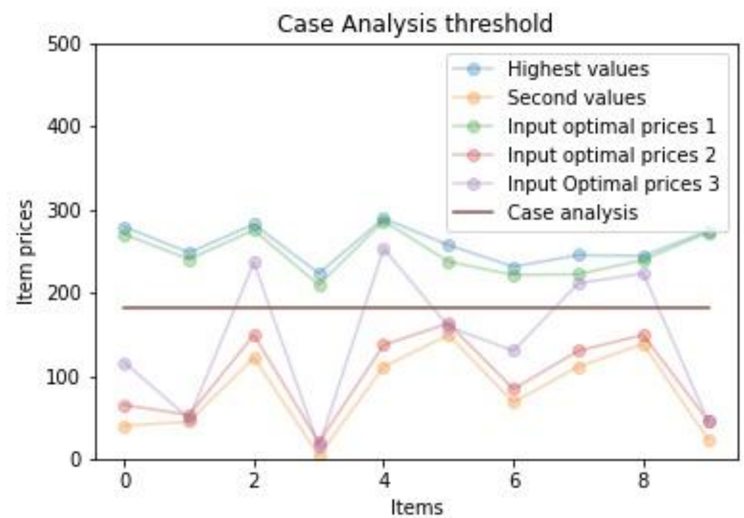
- Essentially the two groups are <u>interchangeable</u>.



Colorful bidders are allocated some item(s) in the optimal allocation.

Handling different optimal price vectors





Case analysis on the initial price vector

- ## <u>High revenue case</u>

- Undershooting $(d = 64)$ - **revenue** is already guaranteed to be **high**.

- Either N_{Sample} or N_{Post} <u>retain</u> high revenue.

- **Constant** approximation ratio for the Expected Welfare.

- ## <u>Low revenue case</u>

- Overshooting $(d = 8)$ - the prices were both **optimal** and **low** revenue.

- The **value** of the items that **surpass** their respective highest valuation <u>cannot be too high</u>!

- **Constant** approximation ratio for the Expected Welfare.

Approximation ratio

Theorem 5.2.1. *The Sampling Mechanism with input an optimal proportional price vector achieves a constant approximation ratio to the Optimal Social Welfare.*

The approximation ratio is $\frac{e^2-2}{2048e^2}$

Submodular – Proportional – Supporting prices of the optimal allocation

- Supporting prices $\boldsymbol{q}$ add up to exactly OPT.
- Estimate OPT in the sample (Non Truthfully -> 2 approximation).
- Bound $\boldsymbol{p}'$ such that $\boldsymbol{p}' < \frac{\boldsymbol{q}}{\mathbf{2}}$ and $\sum_{j \in M} p'_j$ is lower bounded.
- Run a Fixed-Price Auction $\boldsymbol{p}'$. Result holds due to the previous Lemma.

Theorem 5.3.1. *There exists a mechanism in the Sampling & Multiplying Mechanism class that, when given a proportional supporting price vector of the Optimal Allocation as input, achieves a constant approximation ratio on the Submodular Combinatorial Auction problem.*

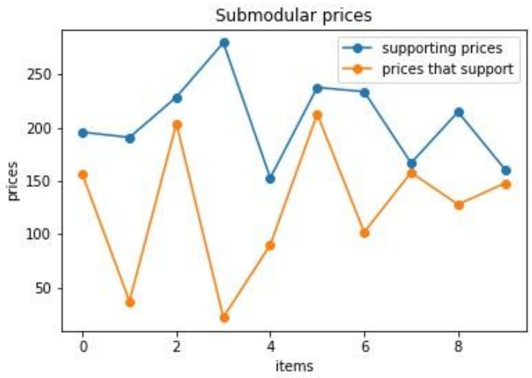
Submodular – Prices that support the optimal allocation

- **Proportional** prices that support the optimal allocation have no meaningful guarantee (e.g. $[0, \dots, 0]$).
  - Low revenue case can't work.

    Select prices such that some of the items are nearly overpriced while the rest are severely underpriced.

- **Non-proportionally** we can observe these prices $\boldsymbol{p}$ as **erroneous predictions**, define the error $\eta = \max_j \left\{\frac{q_j}{p_j}\right\}$ and get the following theorem:

Theorem 5.3.2. *With input prediction a price vector* $\mathbf{p}$ *that supports the Optimal allocation, A Fixed-Price Auction which posts price vector* $\frac{\mathbf{p}}{2}$ *achieves a* 2η*-approximation ratio on the XOS Combinatorial Auction problem.*



a-good price vector

Definition 5.3.1 (a-good proportional price vector). *We define as "a-good proportional price vector" a price vector* $\boldsymbol{p} \in (0,1]^m$, *such that* $\exists$ *some constant* c, *for which the outcome allocation of a Fixed-Price Auction posting prices* $c \cdot \boldsymbol{p}$ *has* $Welfare \geq \frac{OPT}{a}$ *(for any ordering of the bidders in* N*).*

- **Re-examine Additive** problem.

- Check **Low Revenue** case for the **Submodular** problem.

Proportionality - Bad news 1

Theorem 5.3.3. *For any distribution $\mathcal{D}$ and sample probability $\frac{1}{2}$, defining a specific* ***Sampling & Multiplying Mechanism****, there exists an instance of the problem $\mathcal{I}$ (even with Additive valuations), a constant a and a proportional a-good price vector* $\boldsymbol{p}$ *(guaranteeing Welfare $\frac{OPT}{a}$) for which the* ***Sampling & Multiplying Mechanism*** *yields arbitrarily small Welfare with probability $1 - o(1)$.*

Essentially distribution D needs to be $D = D(a)$.

Proof sketch:

- Two groups of items, the first has items priced at the highest optimal value, while the second has them priced near 0.
- Select the value of the first group to be $\frac{OPT}{a}$ and select a big enough such that with probability $1 - o(1)$ the rescaled price vector $\boldsymbol{p}'$ overshoots.

Non-Proportionality

v_i	B_1	B_2
i_1	A_1	0
i_2	$2A_2$	$\left(3+\frac{\epsilon}{2}\right)A_2$
i_3	$(1-\epsilon)\,A_2$	0
$c \cdot p$	A_2	$2A_2$

Suppose that $A_1 \gg A_2$.

In an **Fixed-Price Auction** the resulting allocation is $(B_1, B_2, \emptyset)$.

What happens on an Fixed-Price Auction with price vectors $\{c(1+\varepsilon)\boldsymbol{p}, c(1-\varepsilon)\boldsymbol{p}\}$.

With price vector $c(1+\varepsilon)\boldsymbol{p}$

v_i	B_1	B_2
i_1	A_1	0
i_2	$2A_2$	$\left(3+\frac{\epsilon}{2}\right)A_2$
i_3	$(1-\epsilon)A_2$	0
$c \cdot p$	A_2	$2A_2$

- Allocation $(B_1, B_2, \emptyset)$ if bidder i_1 comes before bidder i_2.
- Allocation $(\emptyset, B_1, \emptyset)$ if bidder i_2 comes before bidder i_1.

$$\mathbb{E}\left[Welfare\right] \geq \frac{A_1}{2}$$

With price vector $c(1-\varepsilon)\boldsymbol{p}$

•

v_i	B_1	B_2
i_1	A_1	0
i_2	$2A_2$	$\left(3+\frac{\epsilon}{2}\right)A_2$
i_3	$(1-\epsilon)A_2$	0
$c \cdot p$	A_2	$2A_2$

- Allocation $(B_1, B_2, \emptyset)$ if bidder i_1 comes before bidder i_3.
- Allocation $(\emptyset, B_2, B_1)$ if bidder i_3 comes before bidder i_1.

$$\boxed{\mathbb{E}\left[Welfare\right] \geq \frac{A_1}{2}}$$

Low Revenue Family of Instances

v_i	$B_{(1,k_1)}$	$B_{(1,k_1{}')}$	$B_{(1,k_1)} \cup B_{(1,k_1{}')}$	$B_{(2,k_2)}$	$B_{(2,k_2{}')}$	$B_{(2,k_2)} \cup B_{(2,k_2{}')}$
$i_{(1,k_1)}$	A_1	0	A_1	0	0	0
$i_{(2,k_2)}$	$2A_2$	$2A_2$	$2A_2$	$\left(3+\frac{\epsilon}{2}\right)A_2$	0	$\left(3+\frac{\epsilon}{2}\right)A_2$
$i_{(3,k_3)}$	$(1-\epsilon)A_2$	$(1-\epsilon)A_2$	$2(1-\epsilon)A_2$	0	0	0
$c \cdot p$	A_2	A_2	$2A_2$	$2A_2$	$2A_2$	$4A_2$

Groups S_1, S_2, S_3 with **representative** bidders $i_{(1,k_1)}$, $i_{(2,k_2)}$, $i_{(3,k_3)}$ respectively.

1. $Welfare = OPT.$
2. $Rev\{c \cdot \mathbf{p}\} < \frac{OPT}{32}.$

Instance guarantees

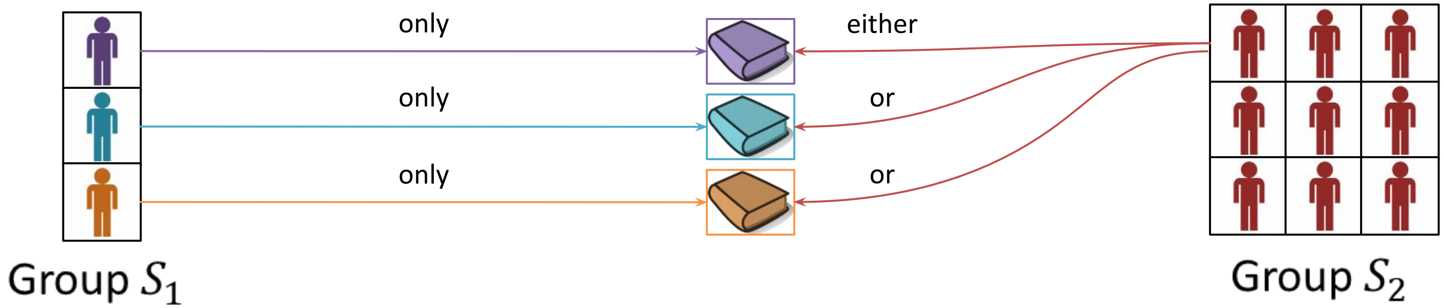
$$A_1 = O\left(\frac{OPT}{|S_1|}\right)$$

$$A_2 = O\left(\frac{OPT}{|S_1|^{k+1}}\right)$$

$$|S_3| = |S_2| = |S_1|^k, k \geq 3$$

$|S_1|$ is left as a **parameter**.

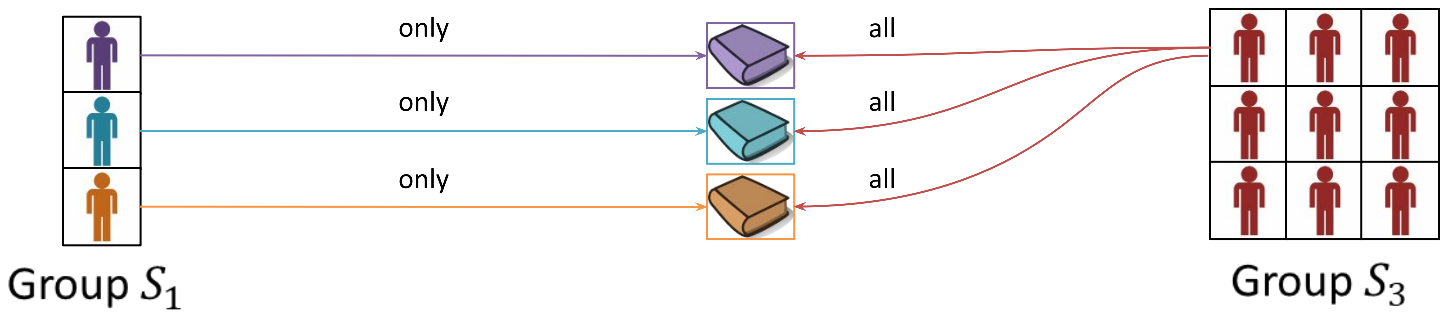
With price vector $c(1+\varepsilon)\boldsymbol{p}$



- The ordering of the Fixed-Price Auction can be studied as a **random experiment** where bidders are drawn from groups S_1 and S_2 **without replacement**.
- Count the **number of arrivals** of bidders from S_1 in the first $|S_1|$ positions (optimistic).
- $|S_2| = |S_1|^k \gg |S_1|$, thus Chebyshev's inequality yields the following bound:

$$\mathbb{E}\left[Welfare\right] = O\left(\frac{OPT}{|S_1|}\right)$$

With price vector $c(1-\varepsilon)\boldsymbol{p}$

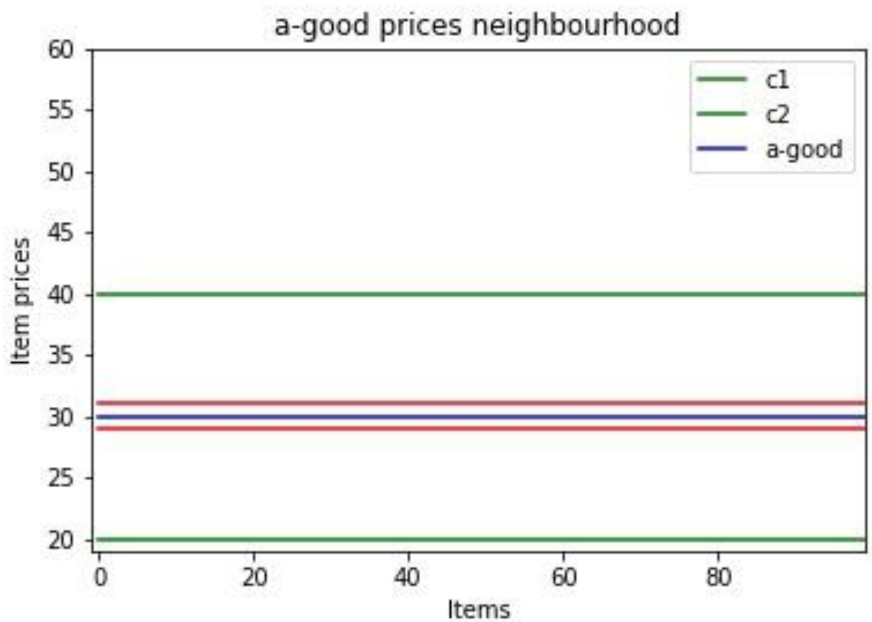


- Simpler game because we are **ONLY** interested in the **first arrival** of **one** bidder from S_3.
- Again we can apply the Chebyshev's inequality to prove the same bound:

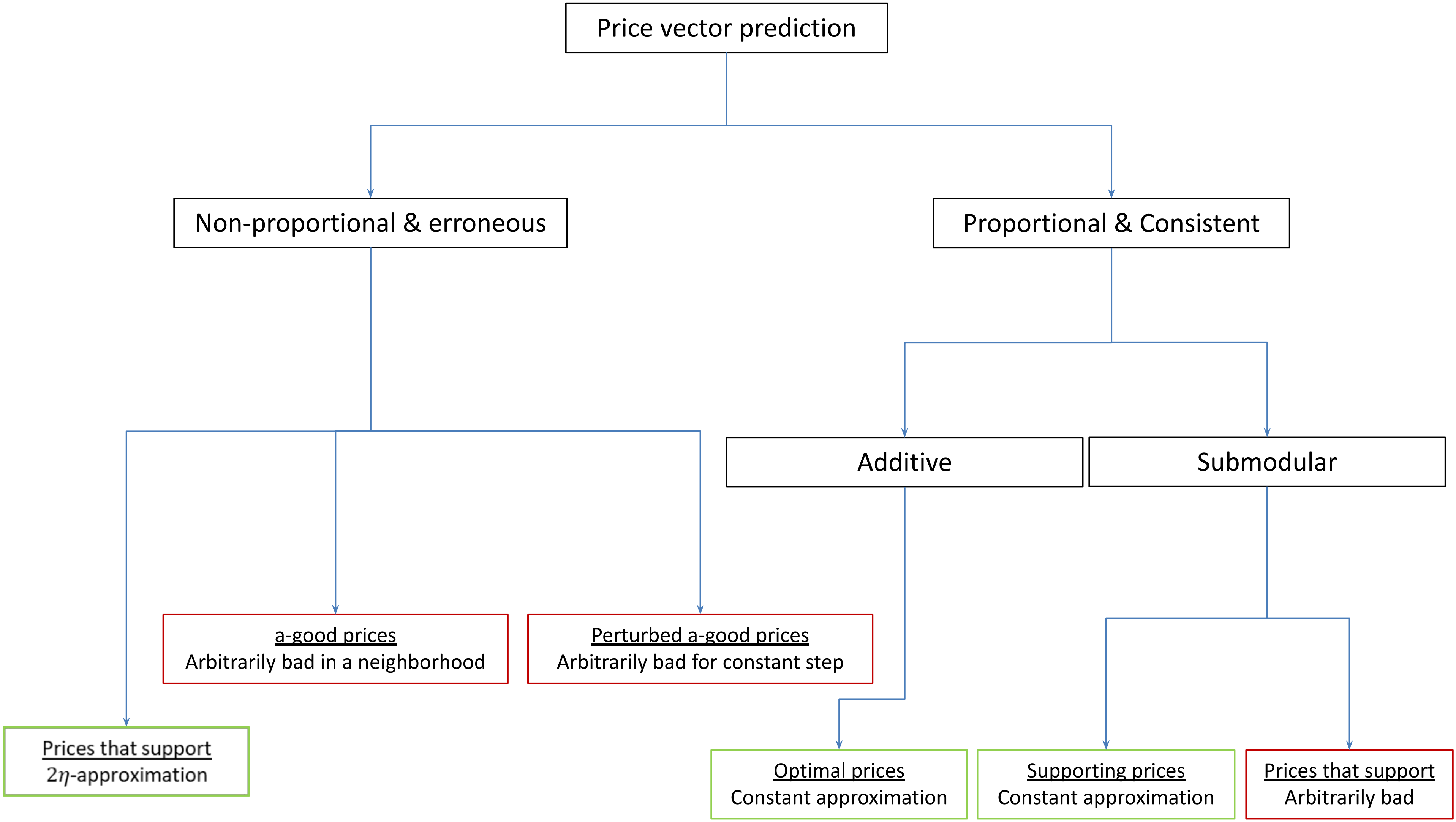
$$\mathbb{E}\left[Welfare\right] = O\left(\frac{OPT}{|S_1|}\right)$$

Bad News 2.

Theorem 5.3.4. *The exist a family of a-good price vectors* $\{C \cdot \boldsymbol{P}\}$ *such that for a specific price vector* $c \cdot \boldsymbol{p}$ *we can define a family of instances of the problem* $\mathcal{I}$ *with instance specific constants* $\{c_1, c_2, \epsilon\}$ *, such that* $c_1 < c < c_2$ *and for any* $c' \in (c_1, (1-\epsilon)c) \cup ((1+\epsilon)c, c_2)$*, the* $\mathbb{E}[Welfare]$ *of a Fixed-Price Auction posting prices* $c' \cdot p$ *is arbitrarily small, where expectation is taken over the random order of the bidders.*



Quick Recap



Price vector prediction
Non-proportional & erroneous
Proportional & Consistent
Additive
Submodular
a-good prices
Arbitrarily bad in a neighborhood
Perturbed a-good prices
Arbitrarily bad for constant step
Prices that support
2η-approximation
Optimal prices
Constant approximation
Supporting prices
Constant approximation
Prices that support
Arbitrarily bad

Thank you!