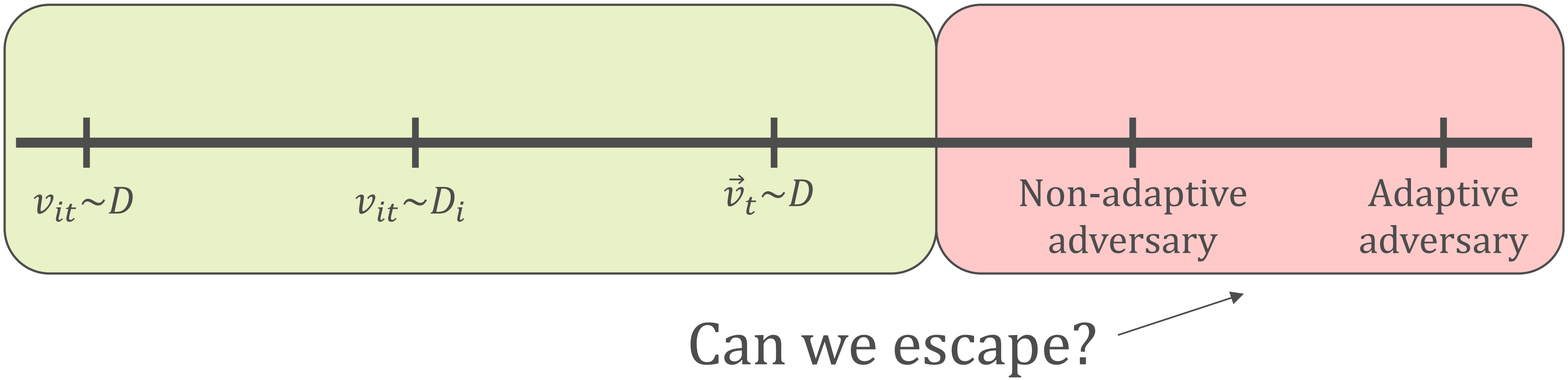
Fair and Efficient Online Allocations: Part II

Alexandros (Alex) Psomas
Purdue University


$v_{it} \sim D$
$v_{it} \sim D_i$
$\vec{v}_t \sim D$
Non-adaptive adversary
Adaptive adversary
Can we escape?

OLD SETUP

- n additive agents and T indivisible items.
- For each agent i and each item $t \in T$, let $\boldsymbol{v_{it}}$ be the preference/valuation agent i for item t.
- The items arrive online (one per round) and the agents' values are revealed when the items arrive (and are chosen by a non-adaptive adversary)

NEW SETUP

- Two agents and T divisible items.
- For each agent i and each item $t \in T$, let $\boldsymbol{v_{it}}$ be the preference/valuation agent i for item t.
- The items arrive online (one per round) and the agents' values are revealed when the items arrive (and are chosen by a non-adaptive adversary).
- Agents' valuations are normalized so that $\sum_{t \in T} \boldsymbol{v_{it}} = \mathbf{1}$.
- A fractional allocation $\mathbf{x}$ defines fraction $\boldsymbol{x_{it}}$ of item t agent i is going to receive.
- The values are additive, given an allocation rule $\mathbf{x}$, the ***utility*** of agent i is defined as:

$$\boldsymbol{u_i}(\mathbf{x}) = \sum_{t \in T} \boldsymbol{v_{it} x_{it}}$$

SETUP

- Our Goal: **Maximize Social Welfare**

$$\max_{\mathrm{x}} \sum_{i \in N} u_i(\mathbf{x})$$

- Fairness Constraint: **Fair share**

$$u_i(\mathbf{x}) \geq 1/2 \quad \text{for all i}$$

- Feasibility Constraint:

$$\sum_{i \in N} x_{it} \leq 1 \text{ for all } t$$

- Performance Measurement: we say some algorithm $\mathcal{A}$ is an α-approximation to the **optimal social welfare** if:

$$\min_{\mathrm{v}} \frac{SW\left(\mathrm{x}^{\mathcal{A}}(\mathrm{v})\right)}{SW\left(\mathrm{x}^{\mathrm{OPT}}(\mathrm{v})\right)} \geq \alpha$$

ONLINE VS OFFLINE

Values	Agent 1	Agent 2
Item 1	0.9	0.6

What should we do?

Allocations	Agent 1	Agent 2
Item 1	**1**	**0**

Values	Agent 1	Agent 2
Item 2	0.1	0.4

Fair share is violated

Allocations	Agent 1	Agent 2
Item 1	**0.5**	**0.5**

Values	Agent 1	Agent 2
Item 2	0.1	0.4
Allocations	**0.5**	**0.5**

$$\text{Optimal Social Welfare} = 0.9 + 0.4 = 1.3$$

$$\text{Algorithm Output} = 1$$

$$\alpha = 76.9\%$$

ONLINE VS OFFLINE

Values	Agent 1	Agent 2
Item 1	0.9	0.6
Item 2	0.1	0.4

Allocations	Agent 1	Agent 2
	5/6	1/6
	0	1

Optimal Social Welfare $= 0.9 + 0.4 = 1.3$

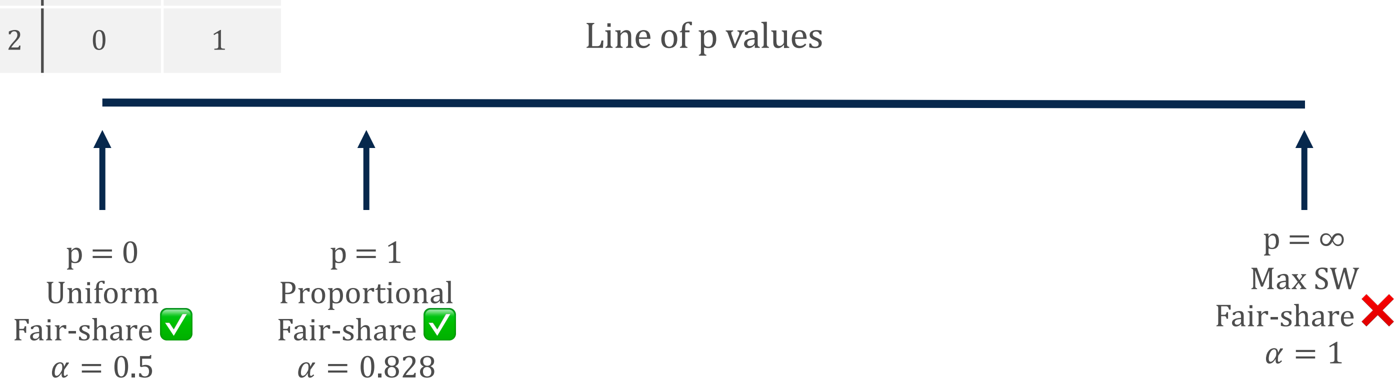
Fair-share Optimal Welfare $= 0.75 + 0.5 = 1.25$

$\alpha = 96.2\%$

POLY-PROPORTIONAL ALGORITHM

Definition: Poly-proportional algorithms are a family of **non-adaptive**, **anonymous** algorithms that allocate an item "proportionally" with some power p: $x_{it} = \frac{v_{it}^p}{\sum_{j=1}^{n} v_{jt}^p}$

	Agent 1	Agent 2
Item 1	1	0
Item 2	0	1



$p = 1$: FAIR SHARE PROOF

- Milne's inequality:

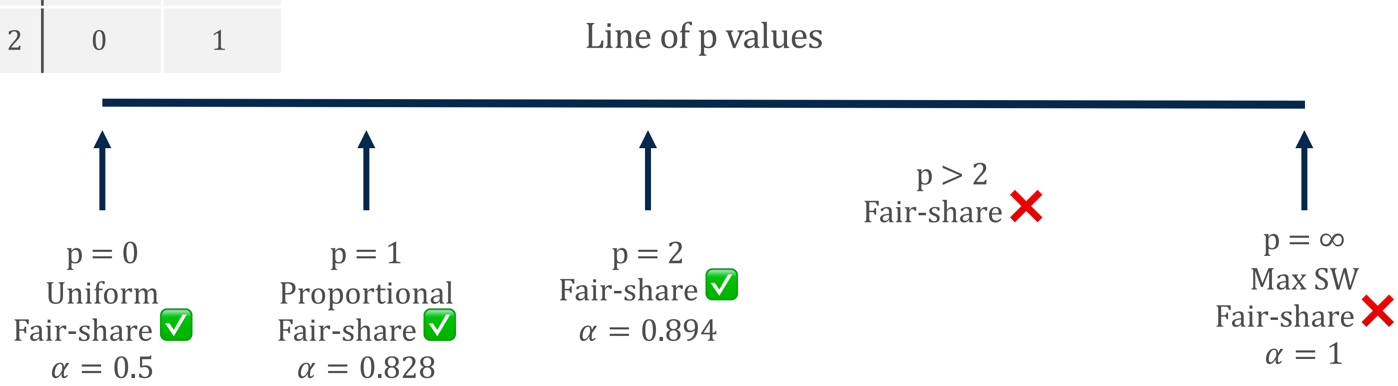
$$\sum_{j=1}^{m} \frac{x_j y_j}{x_j + y_j} \leq \frac{(\sum_{j=1}^{m} x_j)(\sum_{j=1}^{m} y_j)}{\sum_{j=1}^{m} x_j + \sum_{j=1}^{m} y_j}$$

- Plugging in $x_j = v_{1,j}$ and $y_j = v_{2,j}$:
- LHS is the value of agent 1 for agent 2's allocation
- RHS $= 1/2$

POLY-PROPORTIONAL ALGORITHM

Definition: Poly-proportional algorithms are a family of **non-adaptive**, **anonymous** algorithms that allocate an item "proportionally" with some power p: $x_{it} = \frac{v_{it}^p}{\sum_{j=1}^n v_{jt}^p}$

	Agent 1	Agent 2
Item 1	1	0
Item 2	0	1



CRITICAL POINT

- Let's consider p = 3 on the following instance:

Values	Agent 1	Agent 2	Allocations	Agent 1	Agent 2	Utility of agent 2
Item 1	0.6386	0.5		0.6757	0.3243	0.1622
Item 2	0.3065	0.24		0.6756	0.3244	**0.24**
Item 3	0.0549	0.26		**0**	**1**	

	Agent 1	Agent 2
Utility	0.6386✅	0.5✅

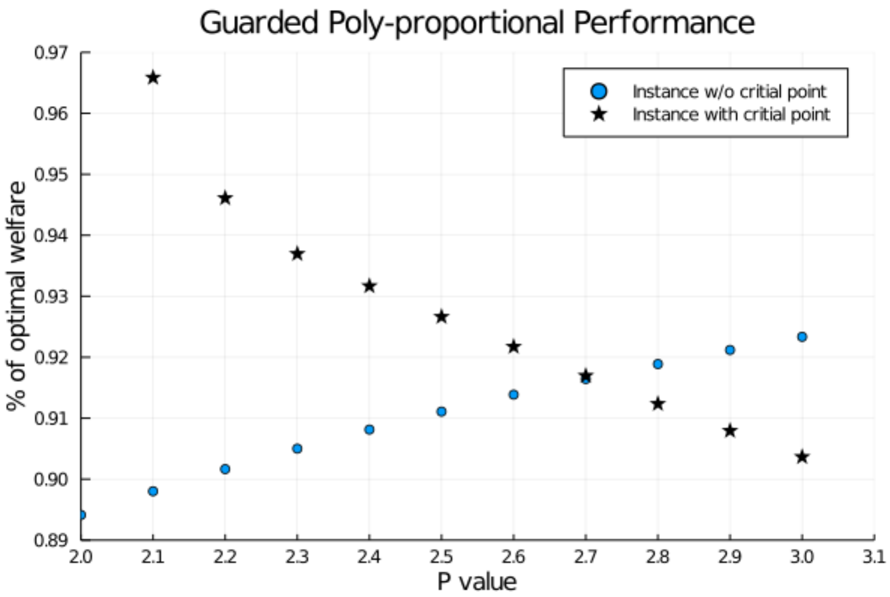
Guarded Poly-proportional

- **Critical Point**: At the end of some round c, the utility that agent i has received so far plus her value for all the remaining items is exactly 1/2 , i.e.,

$$\sum_{t=1}^{c} v_{it}x_{it} + \sum_{t=c+1}^{T} v_{it} = 1/2$$

- **Guarded Poly-Proportional Algorithms**: Perform poly-proportional until one of the agent reaches critical point(if there is one), then fully allocates all the remaining items to that agent.

- Lemma: The guarded poly-proportional algorithm with any $p \geq 0$ satisfies fair-share. ✅
- For $p \leq 2$, critical point never occurs

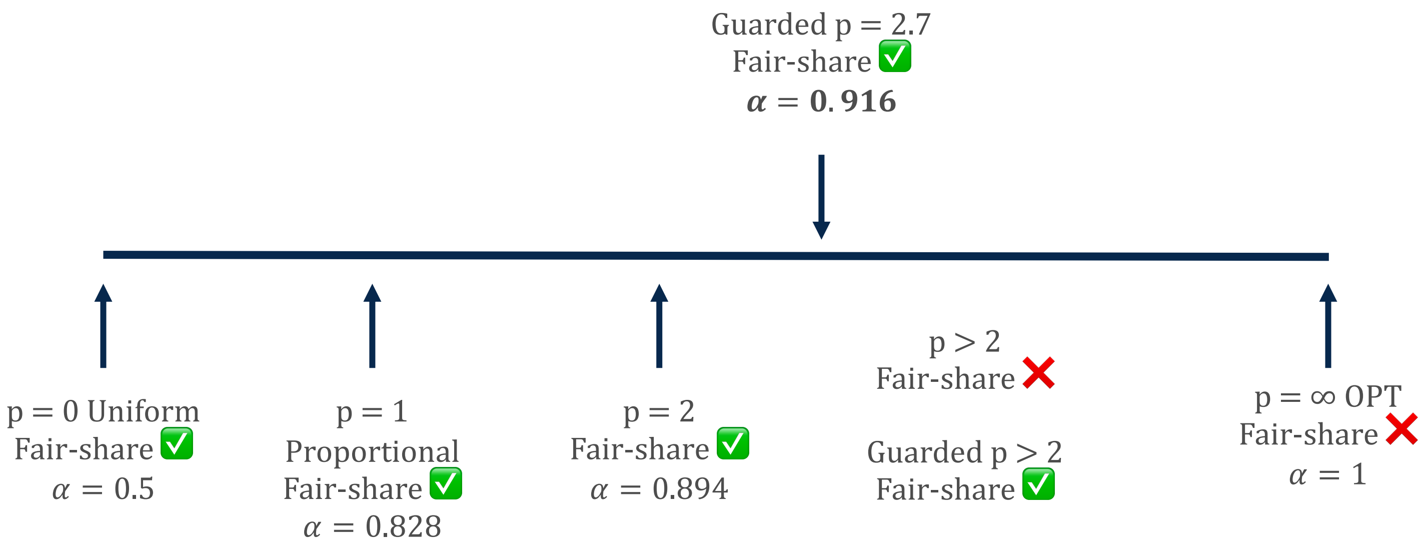


Guarded poly-proportional with p = 2.7:

$\alpha = 0.916$

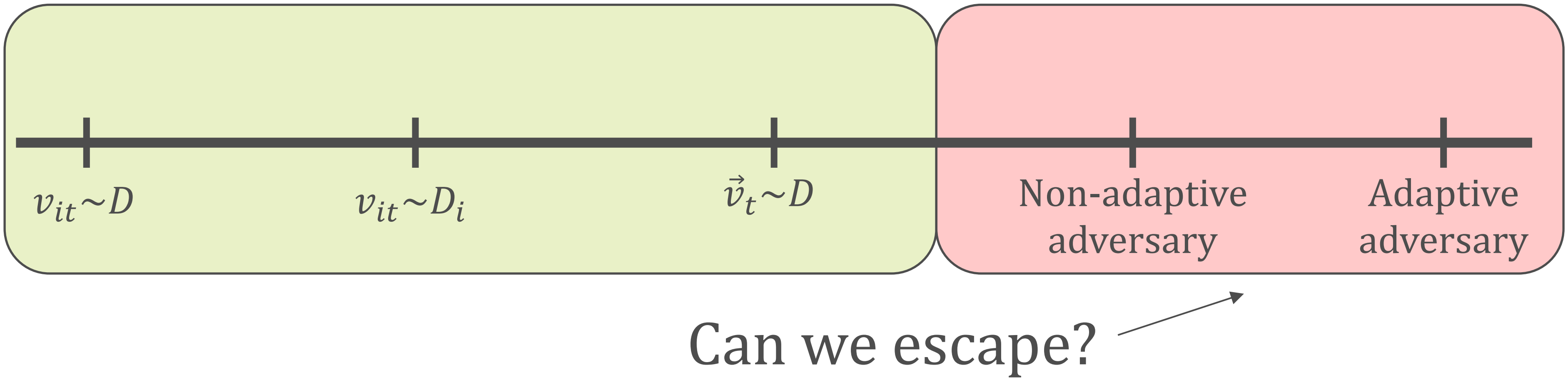
SUMMARY

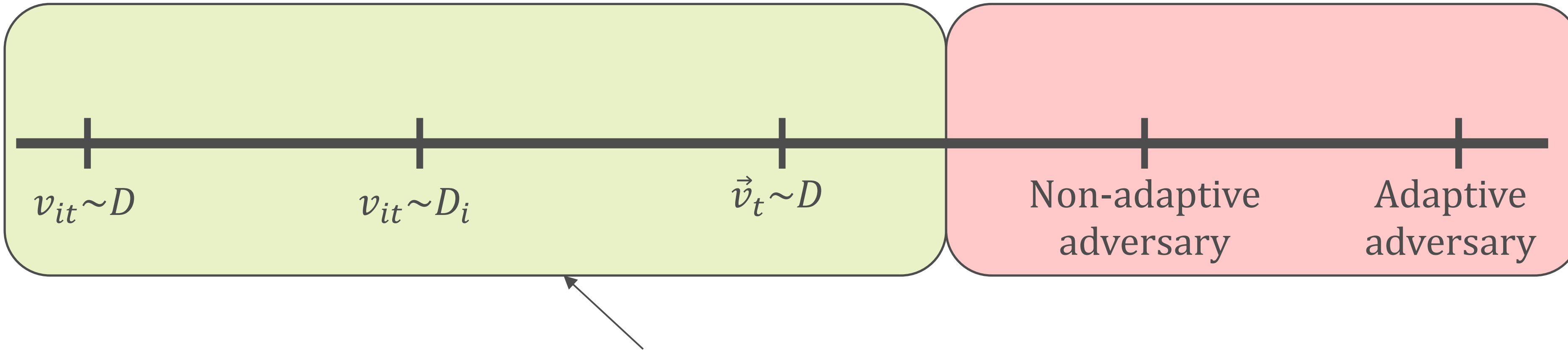
<u>Theorem</u>[GPT 2021]: There is no online fair-share algorithm that achieves an approximation to the optimal welfare better than **0.933**



MULTIPLE AGENTS CASE

- Caragiannis et al.(2012) prove that even if we knew all the values in advance, the *price of fairness* is $O\left(\frac{1}{\sqrt{n}}\right)$.
- The proportional algorithm matches this bound in an online manner, and therefore achieves the optimal approximation.
- If we were to restrict the benchmark to be the **optimal social welfare subject to the fair-share constraint**, still no online algorithm could achieve an approximation better than $\Omega\left(\frac{1}{\sqrt{n}}\right)$.


$v_{it} \sim D$
$v_{it} \sim D_i$
$\vec{v}_t \sim D$
Non-adaptive adversary
Adaptive adversary
Can we escape?



Can we get even more?

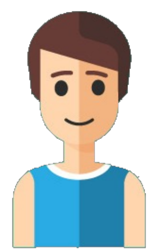
- Major issue with model so far: agents must be expressive
  - Reporting an exact numerical value for each item is too much for many applications of interest

OLD SETUP

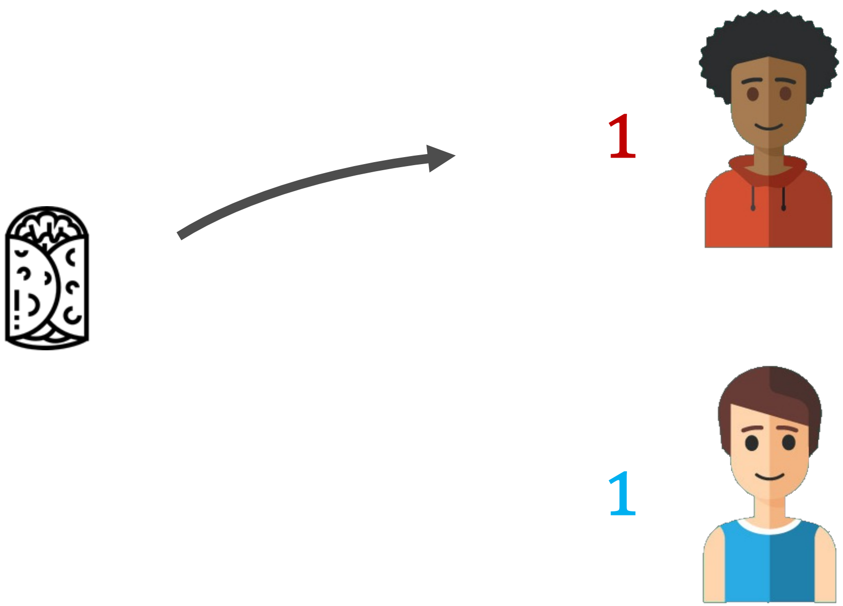
- n additive agents and T indivisible items.
- For each agent i and each item $t \in T$, let $\boldsymbol{v_{it}}$ be the preference/valuation agent i for item t.
- The items arrive online (one per round) and the agents' values are revealed when the items arrive
- There is a known distribution D_i for each agent i from which her values are drawn from

NEW SETUP: PARTIAL INFORMATION [BHP 2022; UNPUBLISHED]

- n additive agents and T indivisible items.
- For each agent i and each item $t \in T$, let $\boldsymbol{v_{it}}$ be the preference/valuation agent i for item t.
- The items arrive online (one per round) and the agents' values are realized when the items arrive
- There is a unknown distribution D_i for each agent i from which her values are drawn from
- Our algorithms never learn the value of an item
- Instead, we learn the relative rank of agent i for item t, with respect to previously allocated items

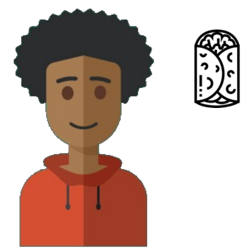


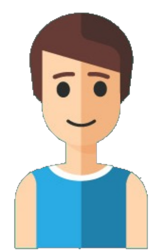
What the algorithm knows

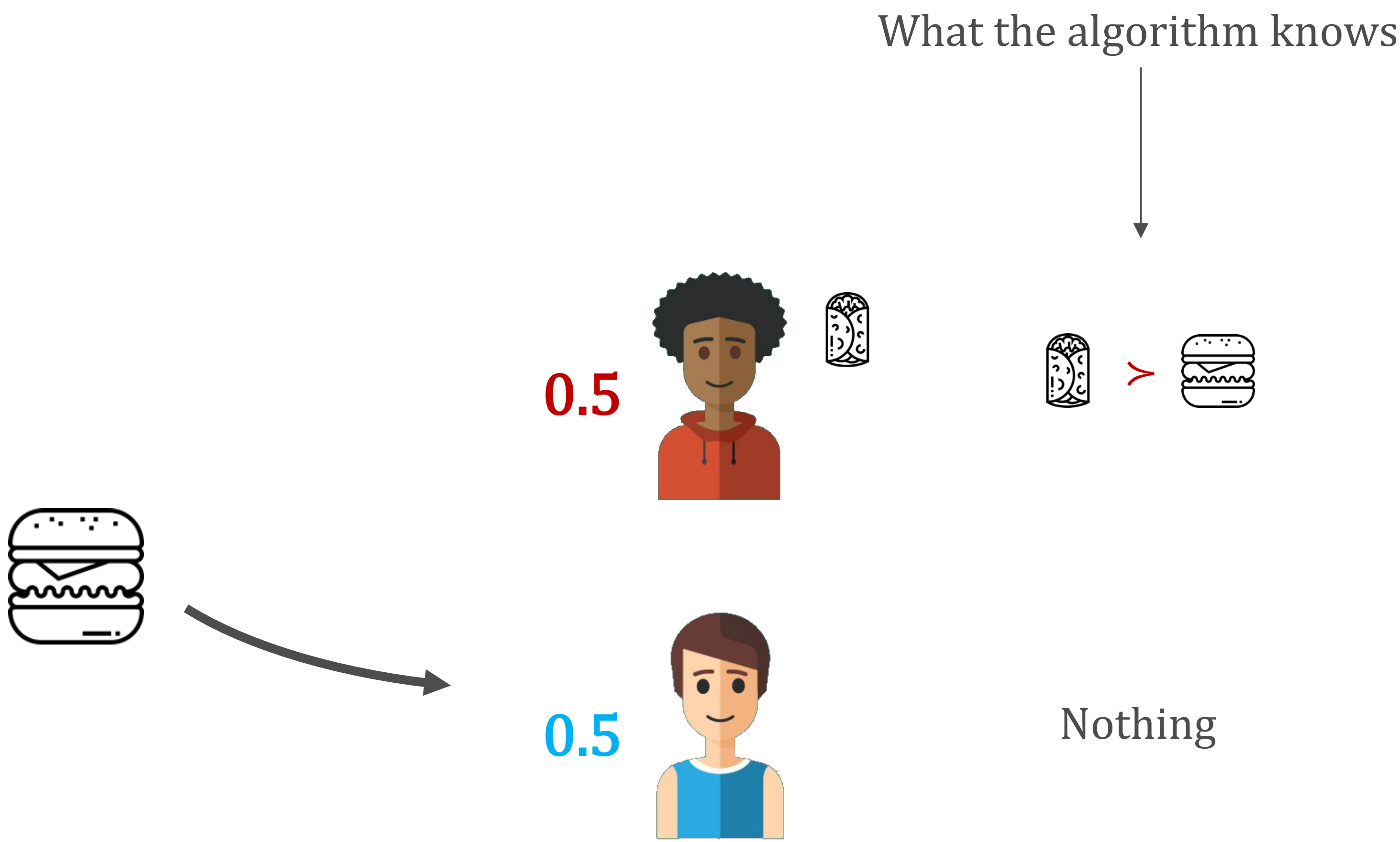


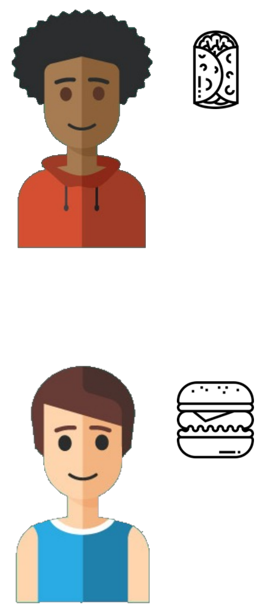
Nothing

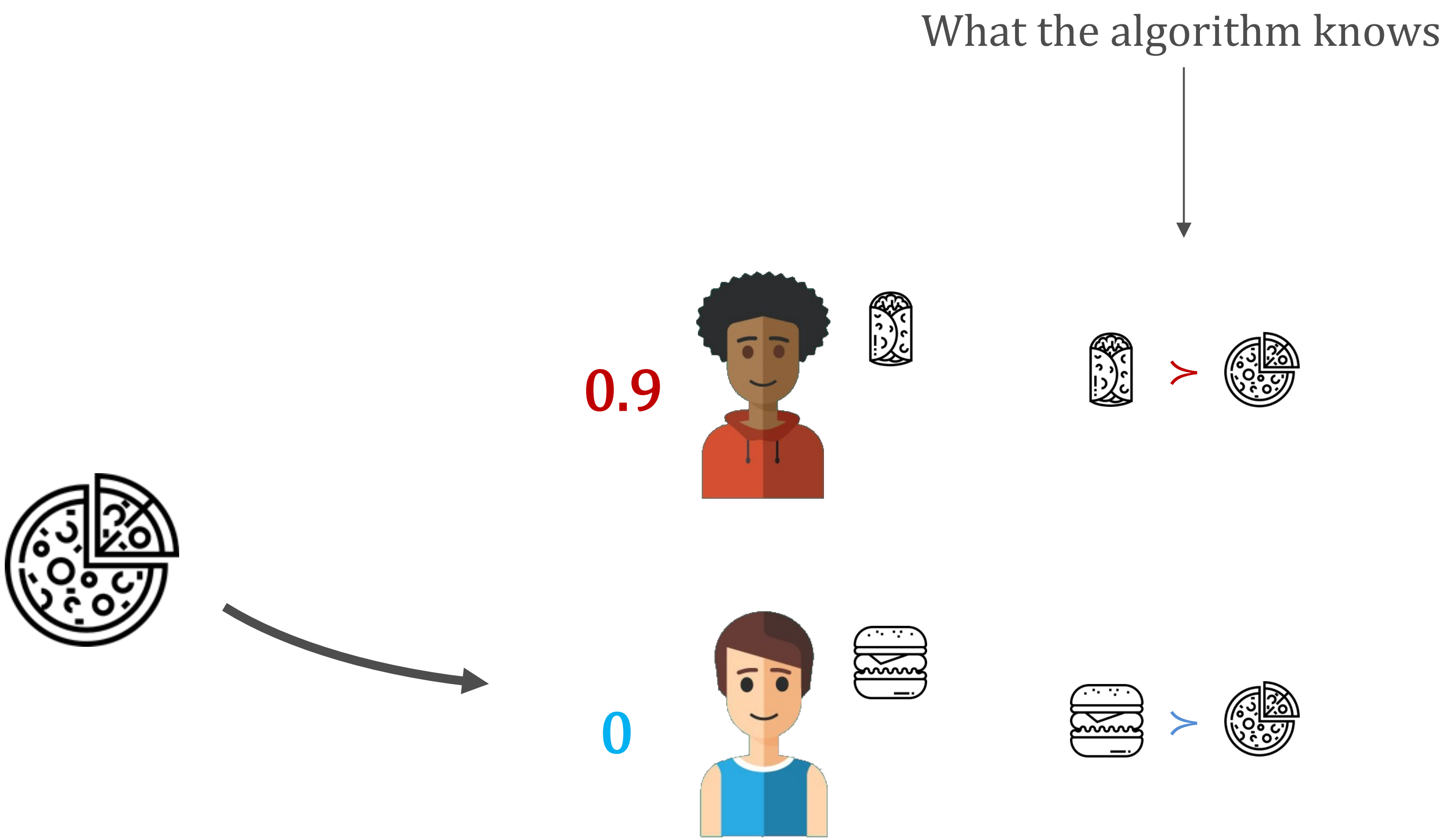
Nothing






What the algorithm knows
0.5
≻
0.5
Nothing





What the algorithm knows
0.9
0
≻
≻

WHAT CAN WE DO?

- Empirical quantiles will be important for us
- Given a fresh item t, we will try to estimate its true quantile value $q_{i,t} = \Pr[\, D_i \leq v_{i,t}]$
- We will do almost as well as an ideal algorithm that has access to true quantiles

IDEAL ALGORITHMS

- Quantile maximization: allocate each item to the agent with the highest quantile)
- q-threshold: allocate each item uniformly at random among agents whose quantile is at least q.
- Lemma[DGKPS 14]: Both algorithms are strongly envy-free with high probability.
- Lemma: The $\frac{n-1}{n}$-threshold algorithm guarantees a $\left(1-\frac{1}{e}\right)^2 \approx$ 0.4 approximation to welfare in the i.i.d. setting
- Property $\mathcal{P}^*$: if there is exactly one agent whose quantile is at least $1-1/n$, she gets the item
- Lemma: Every algorithm that satisfies $\mathcal{P}^*$ guarantees a $1/\mathrm{e} \approx$ 0.36 approximation to welfare in the non-i.i.d. setting.

WHAT CAN WE NOT DO?

- **Theorem**: Even for $n = 2$ agents, there is no algorithm $\mathcal{A}$ that is one-swap Pareto efficient and envy-free whp, even when agents values are drawn i.i.d. from a known distribution D.
- Proof sketch:
  - The first item must be allocated arbitrarily (e.g. to agent 1 wlog), since no information is available
  - With a constant probability, agent 2 really likes the item (her value is in the top 0.1 quantile), but agent 1 does not (her value is in the bottom 0.1 quantile).
  - Our first decision is an irrevocable mistake: the first item we give to agent 2 has a constant probability of having the opposite quantiles (high for 1, low for 2)
    - And, agent 2 should get items in order to satisfy EF whp

MATCHING THE IDEAL ALGORITHMS

Algorithm 1

- For epoch $k = 1, 2, \ldots :$
  - Explore ($n \cdot k^4$ items):
    - Give k^4 to each agent
  - Exploit (k^8 items):
    - Each item g goes to the agent with the highest empirical quantile, with respect to the exploration phase of epoch k

MATCHING THE IDEAL ALGORITHMS

- Giving m (random) items to each agent, we can get (probabilistic) bounds on the empirical quantile of fresh items
  - "The sample is ϵ-accurate with probability at least $1 - \delta$"
  - ϵ-accurate: the relative rank of a fresh item is correct with probability at least $1 - \epsilon$
- However, we still need epochs!
- The underlying distribution is unknown, so we cannot fix a target accuracy even when shooting for a constant approximation to efficiency

MATCHING THE IDEAL ALGORITHMS

Algorithm 1

- For epoch $k = 1, 2, \ldots$:
  - Explore ($n \cdot k^4$ items):
    - Give k^4 to each agent
  - Exploit (k^8 items):
    - Each item g goes to the agent with the highest empirical quantile, with respect to the exploration phase of epoch k

Lemma: The allocation of Algorithm 1 differs from that of the quantile maximization algorithm after T steps by at most $f(T)$ items, whp, where $f(T) \in O\left(poly(n) \cdot T^{\frac{15}{16}}\right)$.

MATCHING THE IDEAL ALGORITHMS

Algorithm 1

- For epoch $k = 1, 2, \ldots$:
  - Explore ($n \cdot k^4$ items):
    - Give k^4 to each agent
  - Exploit (k^8 items):
    - Each item g goes to the agent with the highest empirical quantile, with respect to the exploration phase of epoch k

Theorem: In the i.i.d. model Algorithm 1 gives a $(1 - \epsilon)$-approximation to welfare for all $\epsilon > 0$, and is envy-free, with high probability.

MATCHING THE IDEAL ALGORITHMS

Algorithm 1

- For epoch $k = 1, 2, \ldots :$
  - Explore ($n \cdot k^4$ items):
    - Give k^4 to each agent
  - Exploit (k^8 items):
    - Each item g goes to the agent with the highest empirical quantile, with respect to the exploration phase of epoch k

Theorem: In the non i.i.d. model Algorithm 1 gives a $1/e$-approximation to welfare for all $\epsilon > 0$, and is envy-free, with high probability.

A MATCHING LOWER BOUND

- Theorem: In the non i.i.d. model, no algorithm is EF and 0.81-PO with probability $p > 2/3$, even for $n = 2$ agents
- Sketch:
    - We consider two distributions $D_{flat} = U[1-w, 1]$ and $D_{skewed} = 1$ w.p. z (and 0 w.p. $1 - z$)
    - The algorithm must be EF + c-PO with probability $2/3$ at time $t \geq T^*$, for some T^*, for each combo of distributions for the agents
    - With constant probability, at time t, the number of items with high quantiles for each agent is near its expectation
        - E.g. Number of items of agent 1 with quantile at least $1 - z$ is $z \pm \delta$
    - Via union bound, there must exist a sequence of items, where a number of things happen: the algorithm satisfies the properties for all combos of distributions, and the sample is “nice”
    - Envy-freeness implies a certain distribution of the high quantile items; we give a Pareto improvement.

PUSHING THE LIMITS EVEN MORE

- Partial information is great
- However, perhaps it is still unreasonable to expect comparisons with all previously allocated items
- What can we do with a fixed memory?
  - An agent can compare fresh items only with items in her memory
  - An algorithm in this model can decide to replace an item in memory with a fresh item
- What can we do with a memory of one item?

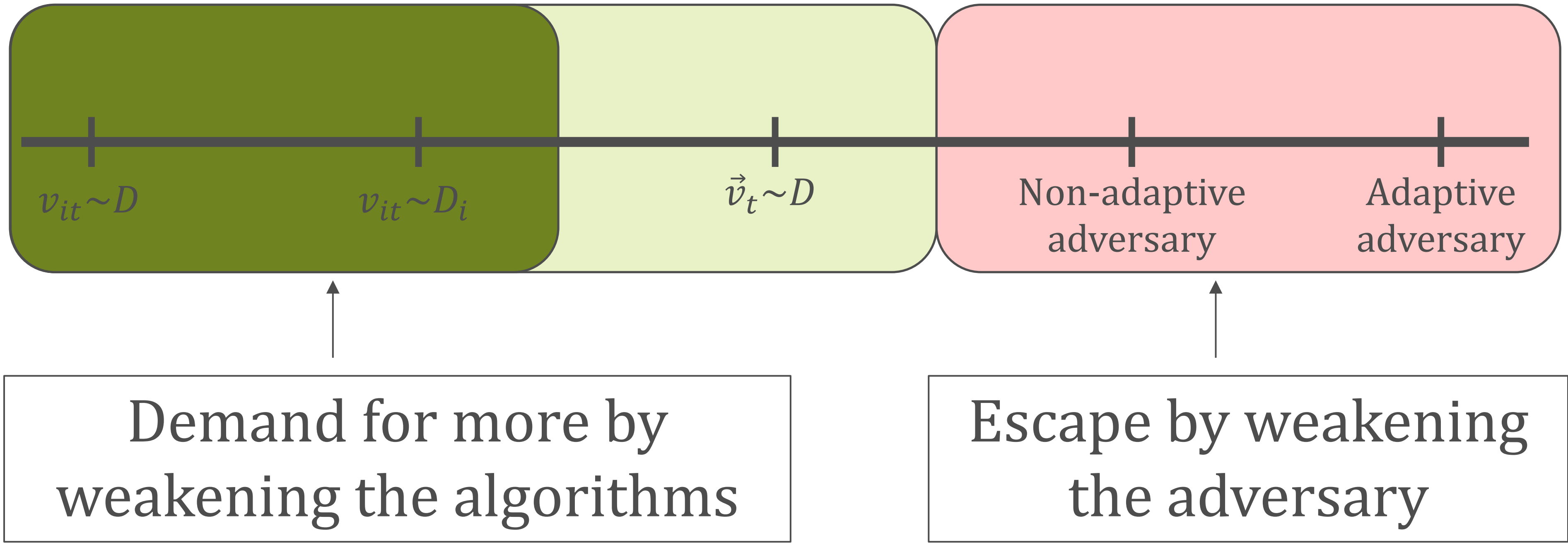
A LOWER BOUND

- **Theorem:** In the i.i.d. model, given a memory of one item per agent, there is no algorithm $\mathcal{A}$ that is 0.999-welfare maximizing with high probability
- So, constant approximations are necessary

A MATCHING UPPER BOUND

- **<u>Theorem:</u>** There exists an algorithm that achieves envy-freeness and a $\left(1-\frac{1}{e}\right)^2 \approx 0.4$ approximation to welfare, with high probability, in the i.i.d. model.
  - The approximation is $1/e$ in the non-i.i.d. model
- Algorithm 2 is similar to Algorithm 1: exploration & exploitation phases
  - We update the memory, and then check if the quantile of the item in memory is useful (close to the ideal $q^* = 1 - 1/n$)
  - But, we need to account for more things going wrong
  - Could be that the item in memory is bad, or that the sample we use to check is bad

SUMMARY


$v_{it} \sim D$
$v_{it} \sim D_i$
$\vec{v}_t \sim D$
Non-adaptive adversary
Adaptive adversary
Demand for more by weakening the algorithms
Escape by weakening the adversary

REFERENCES

- Fair and Efficient Online Allocations with Normalized Valuations. Gkatzelis, Psomas, Tan. AAAI 2021
- Dynamic Fair Division with Partial Information. Benade, Halpern, Psomas. Under submission.
- The efficiencyof fair division. Caragiannis, Kaklamanis, Kanellopoulos, Kyropoulou. TCS 2012

THANK YOU!