Geometric Aggregation of the Social Welfare in Online Resource Allocation

Yinyu Ye
Stanford University
June 21, 2022

Joint work with many

@ The Nanjing U

There are many settings when we need to fairly allocate shared resources to users online

Public Good Allocation

Medical materials Allocation

A key question is how to aggregate society's (linear) utilities to reflect a fair division of resources

Efficiency Objective

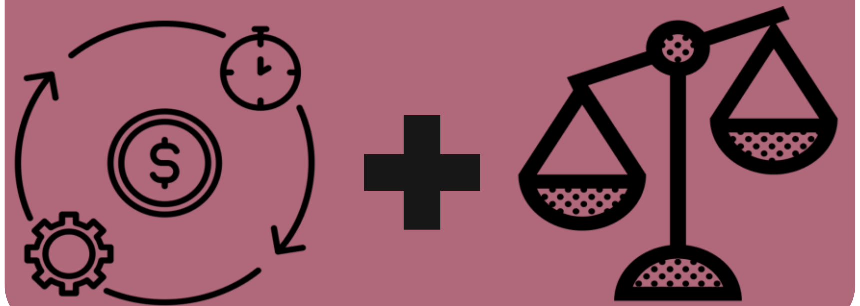
$$\max \sum_i w_i U_i(\boldsymbol{x}_i)$$

Maximize the (weighted) arithmetic sum of agent's utilities, known as **Linear Programming** if u is linear

Nash Social Welfare (NSW) Objective

$$\max \prod_i U_i(\boldsymbol{x}_i)^{w_i}$$

Maximize the (weighted) geometric sum of agent's utilities



[Nash, 1950], [Kaneko, Nakamura, 1979]

Egalitarian Objective

$$\max \min_i w_i \, U_i(\boldsymbol{x}_i)$$

Maximize the minimum (weighted) utility of any agent

w_i: population size or budget of type-i agent

The NSW objective provides a compromise between the efficiency and egalitarian ideals of society

Arithmetic Objective

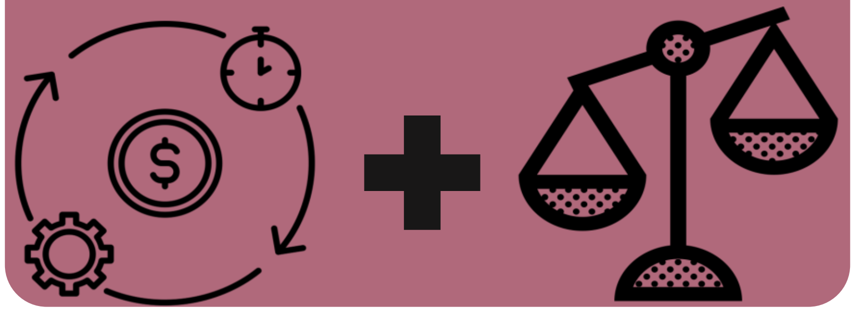
$$\max \sum_i w_i U_i(\boldsymbol{x}_i)$$

Maximize the (weighted) arithmetic sum of agent’s utilities, known as Linear Programming if u is linear

Nash Social Welfare (NSW) Objective

$$\max \prod_i U_i(\boldsymbol{x}_i)^{w_i}$$

Maximize the (weighted) geometric sum of agent’s utilities



Egalitarian Objective

$$\max \min_i (w_i \, U_i(\boldsymbol{x}_i))$$

Maximize the minimum (weighted) utility of any agent

Robustness Property: Provides a lower bound for arithmetic mean objective

Geometric mean objective has several advantages

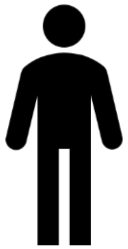
Larger weight (priority) implies higher utility unlike egalitarian objective

Organization

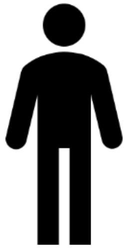
- **Advantages/Properties of (Weighted) Geometric Mean Objective**
- Online Linear Programming
- Online Fisher Markets
- Summaries

Fairness: with the geometric mean objective, all users are guaranteed to get at least some fraction of the resources

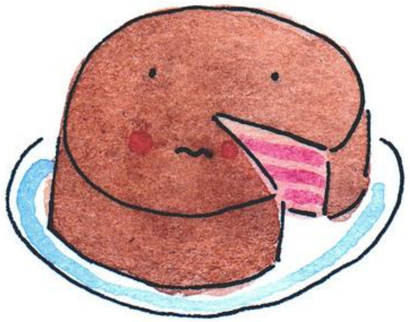
2 Agents



$u_{11} > u_{12}$



1 unit of a divisible resource



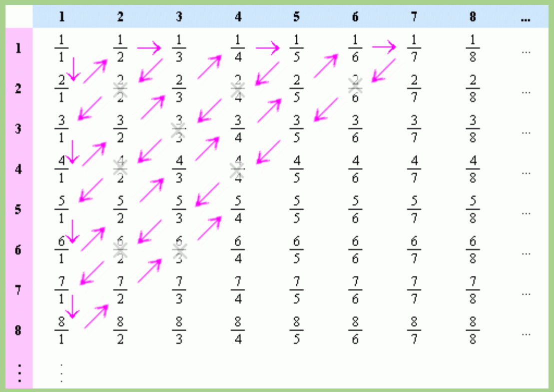
Arithmetic Allocation:
Under the arithmetic mean objective, the entire resource is allocated to agent 1: "big" takes all

Nash welfare allocation:
Under the geometric mean objective each agent receives some portion of the resource

u_{ij} : Preference of Agent i for one unit of good j

The geometric mean objective retains several computational advantages

Rationality of data implies rationality of solution



Exact computation of optimal solutions is possible

The objective can be formulated as a convex optimization problem

$$\max \prod_i U_i(\boldsymbol{x}_i)^{w_i}$$

$$\downarrow$$

$$\max \sum_i w_i \log(U_i(\boldsymbol{x}_i))$$

Computational Complexity is identical to that of a linear program via Interior-Point Method

Optimal solution can be efficiently computed in polynomial time

[Jain 2007], [Y 2008], [Vazirazi 2012],…

The geometric mean objective has several additional advantages

The resulting allocation is envy-free

Each agent prefers their allocation to that of any other agent

The resulting allocation is Pareto efficient

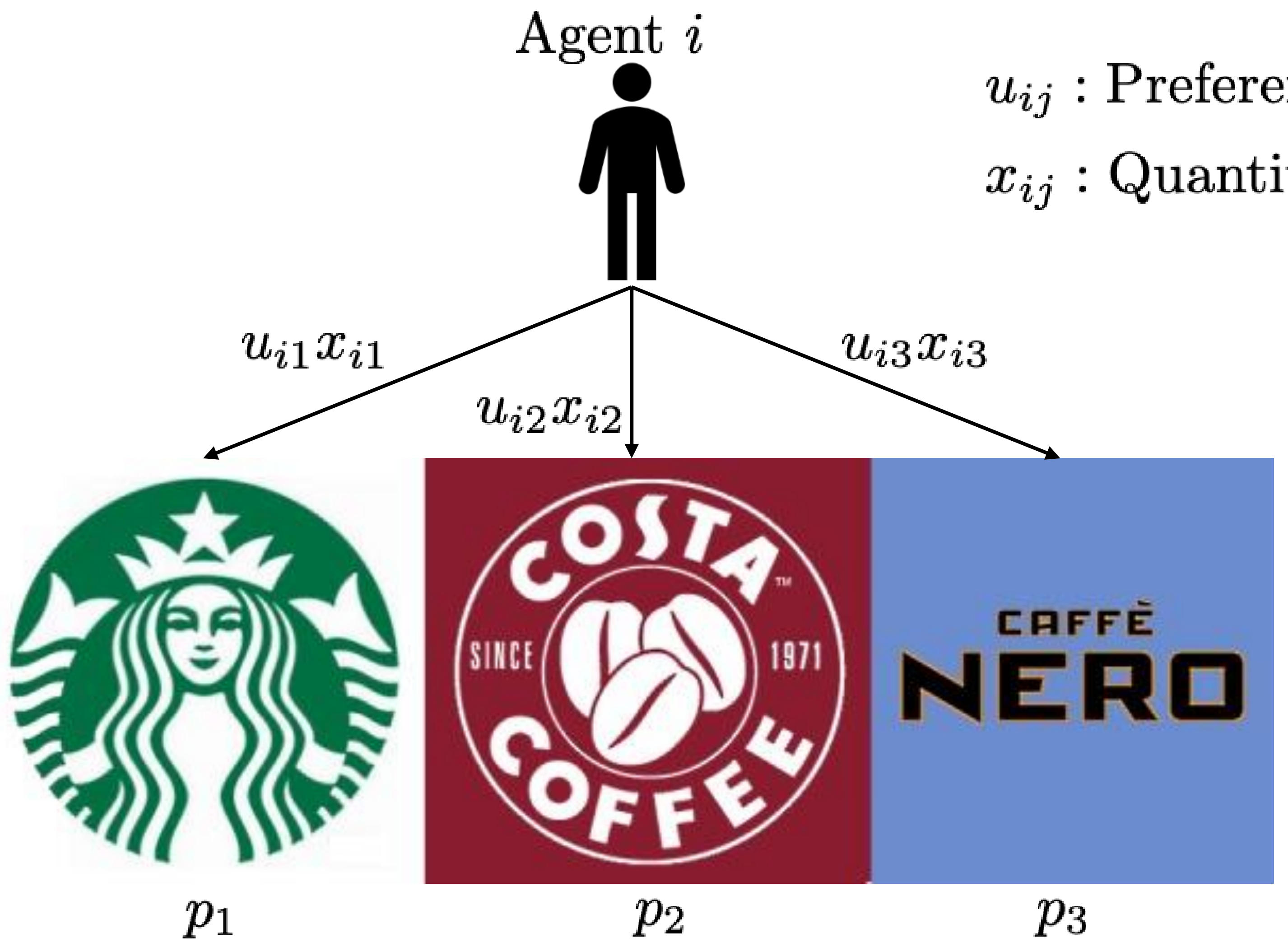
The objective can be formulated as a convex optimization problem

$$\max \prod_i U_i(\boldsymbol{x}_i)^{w_i}$$

↓

$$\max \sum_i w_i \log(U_i(\boldsymbol{x}_i))$$

The NSW objective has a decentralization property captured through the framework of Fisher Markets



M = Total Number of Goods

u_{ij} : Preference of Agent i for one unit of good j

x_{ij} : Quantity of good j purchased by person i

p_j : Price of Good j

w_i : Budget of Agent i

Individual Optimization Problem:

$$\max_{\mathbf{x}_i} \sum_j u_{ij} x_{ij}$$

$$\text{s.t. } \mathbf{p}^T \mathbf{x}_i \leq w_i$$

$$\mathbf{x}_i \geq \mathbf{0}$$

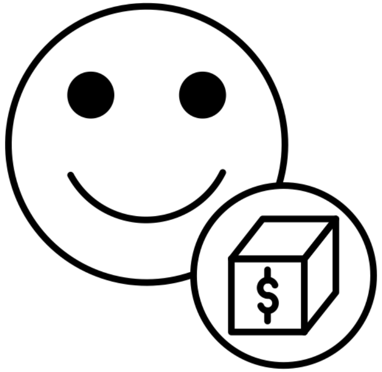
The prices can be derived from a centralized optimization problem with a budget weighted geometric mean objective: freedom-of-choice ⇔ fairness

Individual Optimization Problem:

$$\max_{\mathbf{x}_i} \sum_j u_{ij} x_{ij}$$

$$\text{s.t. } \mathbf{p}^T \mathbf{x}_i \leq w_i$$

$$\mathbf{x}_i \geq \mathbf{0}$$



$$\Longleftrightarrow$$

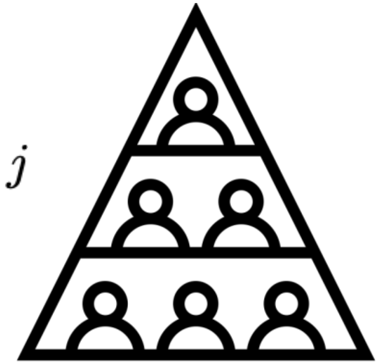
Social Optimization Problem:

$$\max_{\mathbf{x}_i, \forall i \in [N]} \sum_i w_i \log \left(\sum_j u_{ij} x_{ij} \right)$$

$$\text{s.t. } \sum_i x_{ij} \leq c_j, \forall j \in [M]$$

Capacity Constraints

$$x_{ij} \geq 0, \forall i, j$$



p_j : Price of Good j = Dual Variable of Constraint j

$$x_{ij*} = \frac{w_i}{p_{j*}}, \qquad j* = \text{argmin}\{\frac{p_j}{u_{ij}} : u_{ij} > 0\}$$

The applicability of Fisher markets is restricted to the "complete information setting"

Individual Optimization Problem:

$$\max_{\mathbf{x}_i} \sum_j u_{ij} x_{ij}$$

$$\text{s.t. } \mathbf{p}^T \mathbf{x}_i \leq w_i$$

$$\mathbf{x}_i \geq \mathbf{0}$$

$\Longleftrightarrow$

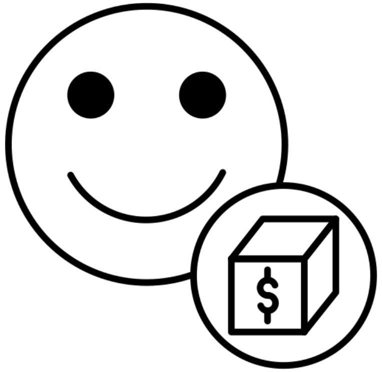
Social Optimization Problem:

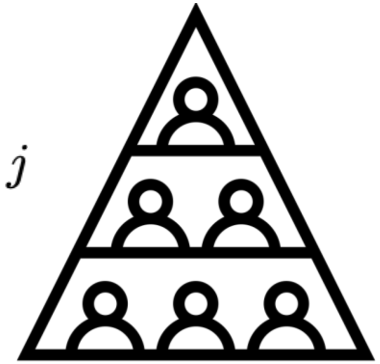
$$\max_{\mathbf{x}_i, \forall i \in [N]} \sum_i w_i \log \left(\sum_j u_{ij} x_{ij} \right)$$

$$\text{s.t. } \sum_i x_{ij} \leq c_j, \forall j \in [M]$$

Capacity Constraints

$$x_{ij} \geq 0, \forall i, j$$

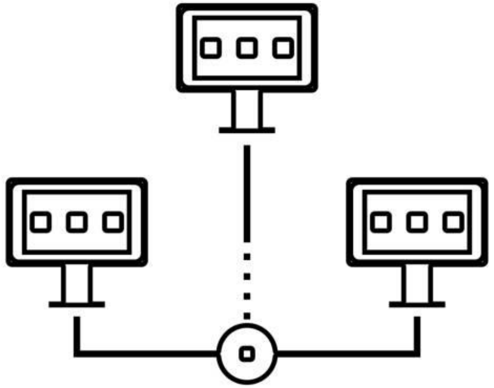


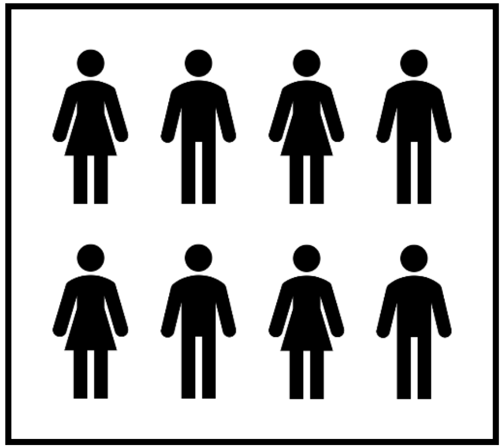


p_j : Price of Good j = Dual Variable of Constraint j

$$x_{ij*} = \frac{w_i}{p_{j*}}, \qquad j* = \text{argmin}\{\frac{p_j}{u_{ij}} : u_{ij} > 0\}$$

Can markets be implemented in an online setting but still achieve social fairness, efficiency and agent-privacy





Each agent distributedly optimizes their individual objectives in response to the set prices

Buyers arrive sequentially with utility and budget parameters in real time

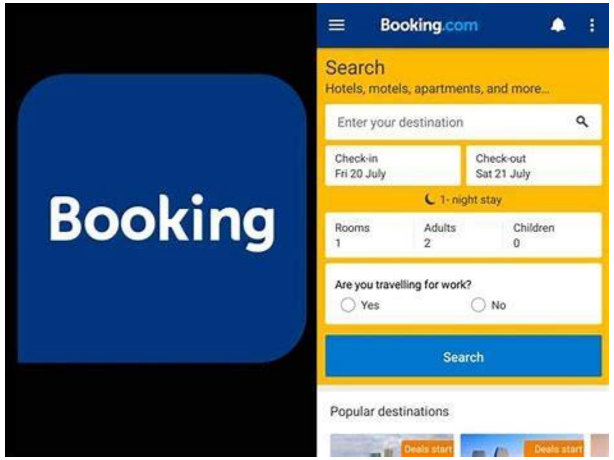
Simulated Market: No trade takes place until equilibrium prices are reached [Cole, Fleischer, 2008] [Panageas, Tröbst, Vazirani, 2021], [Jelota et al. 2021]

Real Market: Market designer learns prices from past buying behavior of users and makes an online decision

Organization

- Advantages of (Weighted) Geometric Mean Objective
- **Online Linear Programming**
- Online Fisher Markets
- Summaries

Online Resource Allocation & Revenue Management



- m type of resources; T customers
- Decision maker needs to decide whether and how much resources are allocated to each customer
- Resources are limited!
- **Online setting**:
  - Customers arrive sequentially and the decision needs to be made instantly upon the customer arrival: **Sell or No-sell**?

$$\max \sum_{t=1}^{T} r_t x_t$$

$$\text{s.t.} \sum_{t=1}^{T} a_{it} x_t \leq b_i, \quad i = 1, \dots, m$$

$$0 \leq x_t \leq 1 \ \text{ or } \ x_t \in \{0, 1\}, \quad t = 1, \dots, T$$

Performance of online algorithm measured with respect to regret from the offline linear objective

[Agrawal et al. 2010, 2014], [Kesselheim et al 2014]

[Li/Ye, 2019], [Li et al. 2020],

Online Seller's Market: An Illustration Example

Bid #	$100	$30	….	…	…	Inventory	
Decision	X1=?	X2=?					
Pants	**1**	**0**	**….**	**…**	**…**	**100**	
Shoes	**1**	**0**				**50**	
T-Shirts	**0**	**1**				**500**	
Jackets	**0**	**0**				**200**	
Hats	**1**	**1**	**…**	**…**	**…**	**1000**	

Online Linear Programming

- Agents/Traders come one by one sequentially, buy or sell, or combination, with a combinatorial order/bid **($\mathbf{a}_t, \pi_t$)**
- The seller/market-maker has to make an order-fill decision as soon as an order arrives
- The seller/market-maker faces:
  - **Sell or No-sell – this is an irrevocable decision**
- Optimal Policy/Mechanism?
- The off-line problem can be an (0 1) linear program

$$\max \sum_{t=1}^{T} r_t x_t$$

$$\text{s.t.} \sum_{t=1}^{T} a_{it} x_t \leq b_i, \quad i = 1, \ldots, m$$

$$0 \leq x_t \leq 1 \text{ or } x_t \in \{0, 1\}, \quad t = 1, \ldots, T$$

Off-Line LP

Regret-Ratio for Online Algorithm/Mechanism

$$
\begin{aligned}
\mathrm{OPT}(A,\pi) = \max \quad & \sum_k \pi_k x_k \\
\text{s.t.} \quad & \sum_k a_{ik} x_k \le b_i \quad \forall\, i \in S \\
& 0 \le x_k \le 1 \quad \forall\, k \in N
\end{aligned}
$$

- We know the total number of customers, say *n;*
- Assume customers arrive in a random order or with i.i.d distributions.
- For a given online algorithm/decision-policy/mechanism

$$Z(A,\pi) = E_\sigma[\, \sum_1^n \pi_k x_k \,]$$

$$R(A,\pi) = 1 - \frac{Z(A,\pi)}{OPT(A,\pi)}$$

$$R = \sup_{(A,\pi)} R(A,\pi)$$

Impossibility Result on Regret-Ratio

Theorem: There is no online algorithm/decision-policy/mechanism such that

$$R \leq \mathrm{O}\left(\sqrt{\log(m)/B} \right), \quad B = \min{}_{\mathrm{i}} b_i .$$

Corollary: If $B \leq \log(m)/\varepsilon^2$, then it is impossible to have a decision policy/mechanism such that $R \leq \mathrm{O}(\varepsilon)$.

Agrawal, Wang and Y, “A Dynamic Near-Optimal Algorithm for Online Linear Programming,” 2010.

Possibility Result on Regret-Ratio

Theorem: There is an online algorithm/decision-policy/mechanism such that

$$R \leq \mathrm{O}\left(\sqrt{m\log(n)/B} \right), \quad B = \min_{\mathrm{i}} b_i .$$

Corollary: If $B > m\log(n)/\varepsilon^2$, then there is an online algorithm/decision-policy/mechanism such that $R \leq \mathrm{O}(\varepsilon)$.

Agrawal, Wang and Y, “A Dynamic Near-Optimal Algorithm for Online Linear Programming,” 2010.

Theorem: If $B > \log(mn)/\varepsilon^2$, then there is an online algorithm/decision-policy/mechanism such that $R \leq \mathrm{O}(\varepsilon)$.

Kesselheim et al. “Primal Beat the Dual…,” 2014, …

Online Algorithm and Price-Mechanism: Learning-while-Doing

- **Learn "ideal" itemized-prices**
- **Use the prices to price each bid**
- **Accept if it is an over bid, and reject otherwise**

Bid #	$100	$30	….	…	…	Inventory	Price?
Decision	x1	x2					
Pants	1	0	….	…	…	100	45
Shoes	1	0				50	45
T-Shirts	0	1				500	10
Jackets	0	0				200	55
Hats	1	1	…	…	…	1000	15

Such ideal prices exist and they are shadow/dual prices of the offline LP

How to Learn Shadow Prices Online

For a given ε, solve the sample LP at t=εn, 2εn, 4εn, …; and use the new shadow prices for the decision in the coming period.

$$
\begin{aligned}
\max \quad & \sum_{k=1}^{t} \pi_k x_k \\
\text{s.t.} \quad & \sum_{k=1}^{t} a_{ik} x_k \le (1-h_t)\frac{t}{n} b_i \quad \forall\, i \in S \\
& 0 \le x_k \le 1 \qquad\qquad\qquad \forall\, k \in N
\end{aligned}
$$

Finite-Customer-Type Based LP formulation

In the original offline LP formulation, x_t represents the decision for the t-th customer, $\mathbf{a}_t$ represents the request vector of the t-th customer, and r_t represents the reward of the t-th customer

$$\max \sum_{t=1}^{T} r_t x_t \quad \text{s.t.} \quad \sum_{t=1}^{T} a_t x_t \leq b, \quad x_t \in [0,1]$$

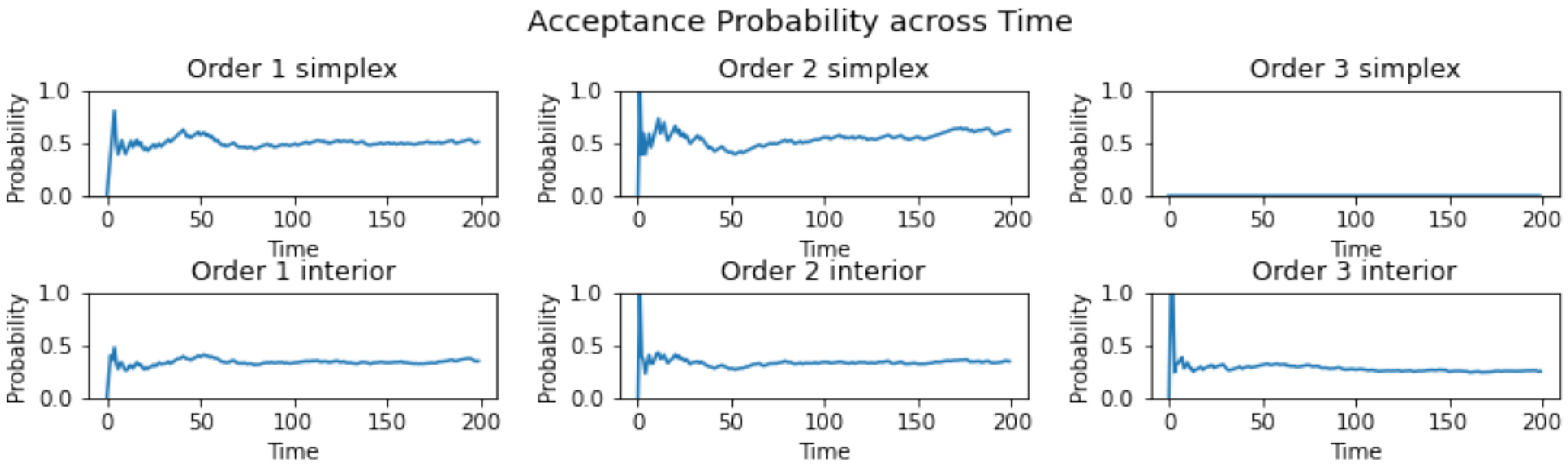
In the finite-customer-type based formulation, there are in total J types of customers. The j-th type arrives with a probability p_j (proportion of type j but unknown); the request vector and reward of the j-th type customer is $\boldsymbol{c}_j$ and μ_j

$$\max \sum_{j=1}^{J} p_j \mu_j y_j \quad \text{s.t.} \quad \sum_{j=1}^{J} p_j c_j y_j \leq b/T, \quad y_j \in [0,1]$$

The decision variable y_j represents the fraction/probability of j-th type customer being accepted. But, in real applications, most LPs have nonunique solutions…

A Motivation Example

- Consider an allocation problem: there exists three types of orders/customers, where the first two types have the reward/resource characteristics that are considered equivalent from the system.
- The following plots show the acceptance fraction/probability of the three types across time by two different online algorithms: the simplex and interior-point methods (Jasin 2015, Chen et al 2021).



Fairness Desiderata

- Technically, Non-Uniqueness/Degeneracy degrades the quality of online algorithm since the learning "targets" are ambiguous – no ground-truth.

- More importantly, Individual Fairness needs to be achieved: similar customers should be treated similarly. Since the optimal object value depends on the total resources spent, not on the resources spent on which groups, some individual or group may be ignored by a particular online algorithm/allocation-rule.

- Also, Time Fairness: The algorithm may tend to accept mainly the first half (or the second half of the orders), which is unfair or unideal…

Fair Optimal Solution for Offline Problem

$$\max \sum_{j=1}^{J} p_j \mu_j y_j \quad \text{s.t.} \quad \sum_{j=1}^{J} p_j c_j y_j \leq b/T, \quad y_j \in [0,1]$$

- We define $\boldsymbol{y}^*$ the fair offline optimal solution of the LP problem as the analytical center of the optimal solution set, which represents an "average" of all the optimal corner solutions – their product is maximized.

- The fair solution $\boldsymbol{y}^*$ will treat individuals fairly, based on their similar reward and resource consumption.

- An online interior-point learning algorithm would use the data points up to time t and solve the sample-based linear program to decide fair $\boldsymbol{y}_t$.

Fairness-Performance Measure

- Let $\boldsymbol{y}_t$ be the allocation rule at time t which encodes the accepting probabilities under the online algorithm π. Then we define the cumulative unfairness of the online algorithm π as

$$UF_T(\pi) = E[\sum_{t=1}^{T} ||\boldsymbol{y}_t - \boldsymbol{y}^*||_2^2]$$

- Intuition: If $UF_T(\pi)$ is sub-linear, we know Time Fairness is satisfied since the deviation of the online solution cannot be large. Moreover, Individual Fairness is satisfied because we know $UF_T(\pi)$ being sub-linear implies $\boldsymbol{y}_t$ converging to $\boldsymbol{y}^*$.
- Let j_t denote the incoming customer type at time t, the Revenue Regret is defined as
- $$\mathrm{Reg}_T(\pi) = E[\sum_{t=1}^{T} r_t(y_{j_t}^* - y_{t,j_t})]$$

Regret measures the performance loss compared to the optimal policy.

Our Result

- We develop an algorithm [Chen, Li & Y (2021)] that achieve

$$UF_T(\pi) = O(\log T)$$

$Reg_T(\pi)$ Bounded independent of T

- Key ideas in algorithm design:
  - At each time t, we use interior-point method to obtain the sample analytic-center solution and randomly make decision based on sample solution $\boldsymbol{y}_t$.
  - We also adjust the right-hand-side resource of the LP to ensure the depletion of binding resources and non-binding resources does not affect the fairness.
  - This state of the art result removes typical non-degeneracy or non-uniqueness assumption in the OLP literature.

The Online Algorithm can be Extended to Bandits with Knapsack (BwK) Applications

- For the previous problem, the decision maker first wait and observe the customer order/arm and then decide whether to accept/play it or not.
- An alternative setting is that the decision maker first decides which order/arm (s)he may accept/play, and then receive a random resource consumption vector $\mathbf{a}_j$ and yield a random reward π_j of the pulled arm.
- Known as the Bandits with Knapsacks, and it is a tradeoff exploration v.s. exploitation

$$\max \quad \sum \pi_j x_j \quad \text{s.t.} \quad \sum_j \boldsymbol{a}_j x_j \le \boldsymbol{b}, \quad x_j \quad \ge 0 \qquad \forall\ j = 1, \ldots, J$$

- The decision variable x_j represents the total-times of pulling the j-th arm.
- We have developed a two-phase algorithm
  - Phase I: Distinguish the optimal super-basic variables/arms from the optimal non-basic variables/arms with as fewer number of plays as possible
  - Phase II: Use the arms in the optimal face to exhaust the resource through an adaptive procedure and achieve fairness
- The algorithm achieves a problem dependent regret that bears a logarithmic dependence on the horizon T. Also, it identifies a number of LP-related parameters as the bottleneck or condition-numbers for the problem
  - Minimum non-zero reduced cost
  - Minimum singular-values of the optimal basis matrix.

Takeaway:

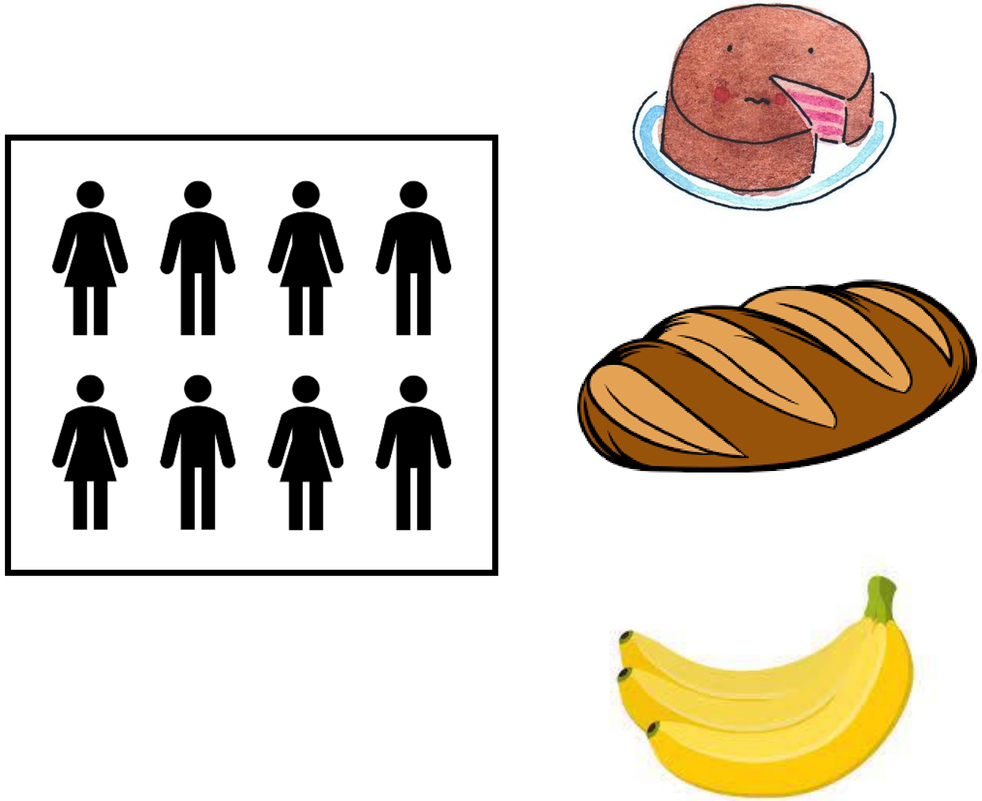
Stochastic data are learnable and partial social fairness is achievable for online linear programming

- **First algorithm** to achieve the $\boldsymbol{O(\log T)}$ regret bound [Li, Sun & Y 2021].

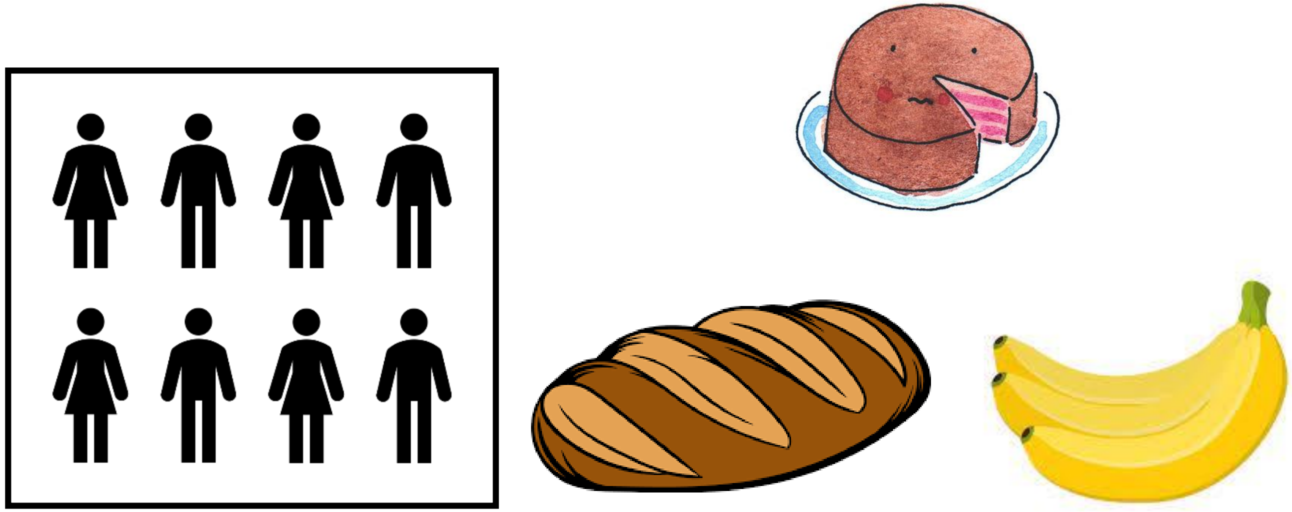
Organization

- Advantages of (Weighted) Geometric Mean Objective
- Online Linear Programming
- **Online Fisher Markets (Real Market)**
- Summaries

Prior work on online variants of Fisher markets have considered the setting of goods arriving sequentially



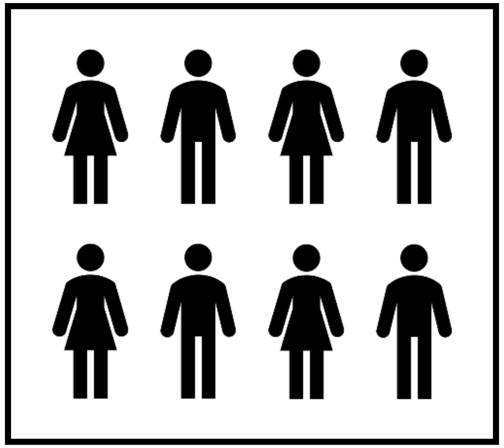
Prior Work: Goods Arrive Online
[Gorokh, Banerjee, Iyer, 2021]

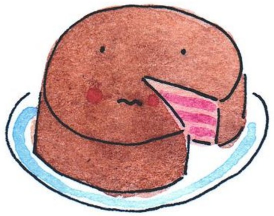


This Work: Agents arrive Online and an irrevocable allocation has to be made:

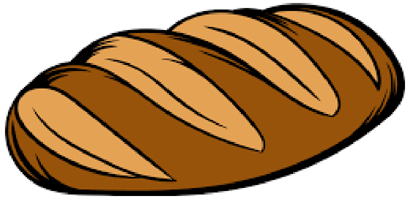
How much the objective value degraded from offline version?

Agents maximize individual utilities based on posted prices that are adjusted based on discrepancy between supply & demand – Buyers' market in real time





Price p_1^t

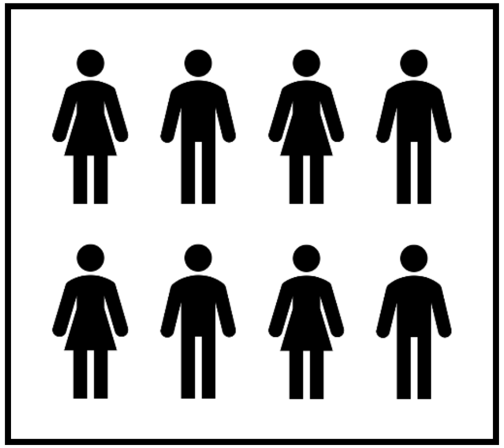


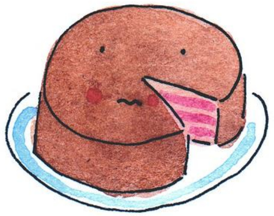
p_2^t

An agent i, with budget w_i, purchases an **"regularized"** optimal bundle $\boldsymbol{x}_i^t$ given price $\boldsymbol{p}^t$

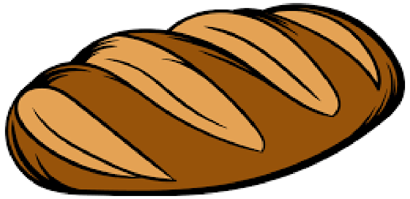
p_3^t

Then prices at time $t+1$ are updated based on observed consumptions $\boldsymbol{x}_i^t$ at time t





p_1^{t+1}



p_2^{t+1}

p_3^{t+1}

Increase Prices: $p_j^{t+1} > p_j^t$ if $\sum_i x_{ij}^t > c_j$

Decrease Prices: $p_j^{t+1} < p_j^t$ if $\sum_i x_{ij}^t > c_j$

Online for Geometric Objective: evaluate algorithms through the absolute regret of social welfare and capacity violation

Regret (Optimality Gap)

Difference in the Optimal Social Objective of the online policy $\boldsymbol{\pi}$ to that of the optimal offline social value

$$R_n(\boldsymbol{\pi}) =$$

$$\sum_i w_i \log\left(\sum_j u_{ij} x_{ij}^*\right) - \sum_i w_i \log\left(\sum_j u_{ij} x_{ij}(\boldsymbol{\pi})\right)$$

Optimal Offline Objective

Objective of online policy

Prior Work on concave objectives [Agrawal/Devanur 2014; Lu, Balserio, Mirrkoni, 2020] assume non-negativity and boundedness of utilities, none of which are true for the NSW

Constraint Violation

Norm of the violation of capacity constraints of the online policy $\boldsymbol{\pi}$

$$V_j(\boldsymbol{\pi}) = \sum_j x_{ij}(\boldsymbol{\pi}) - c_j$$

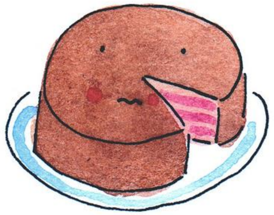
Violation of Capacity Constraint of good *j*

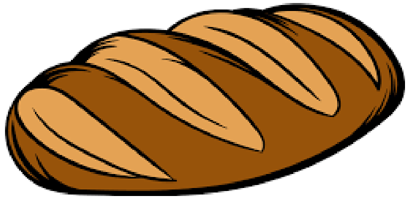
$$V_n(\boldsymbol{\pi}) = ||\mathbb{E}[V(\boldsymbol{\pi})^+]||_2$$

Norm of the expected constraint violation

For any static pricing policy (including the optimal expect prices), either the expected regret or constraint violation is $\boldsymbol{\Omega(\sqrt{n})}$, where $\boldsymbol{n}$ is the number of total agents

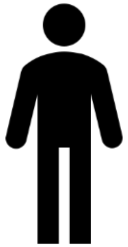
2 goods, each with a capacity of $\boldsymbol{n}$





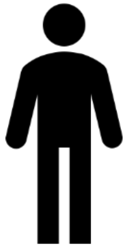
Two agent types specified by (Utility for Good 1, Utility for Good 2)

Type I: (1, 0)



Arrival Probability = 0.5

Type II: (0, 1)



Arrival Probability = 0.5

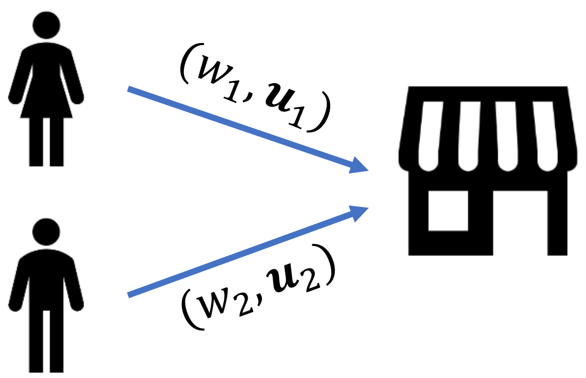
Expected Optimal Objective $\approx \boldsymbol{n\, log(2)}$
Since Type I users receive two units of good one, while type two receive two units of good two

While $\frac{n}{2}$ users of Type I arrive in expectation, the realized arrivals of type I users deviates by $O(\sqrt{n})$

$\sqrt{n}$ − regret of NSW means: $\dfrac{\text{SW optimal geometric mean}}{\text{SW geometric mean of online algorithm}} \leq e^{\frac{1}{\sqrt{n}}}$

Primal algorithms are often computationally expensive and do not preserve user-privacy

User parameters $(w, \boldsymbol{u})$ are revealed



Such algorithms require information on user parameters, which may not be known in practice

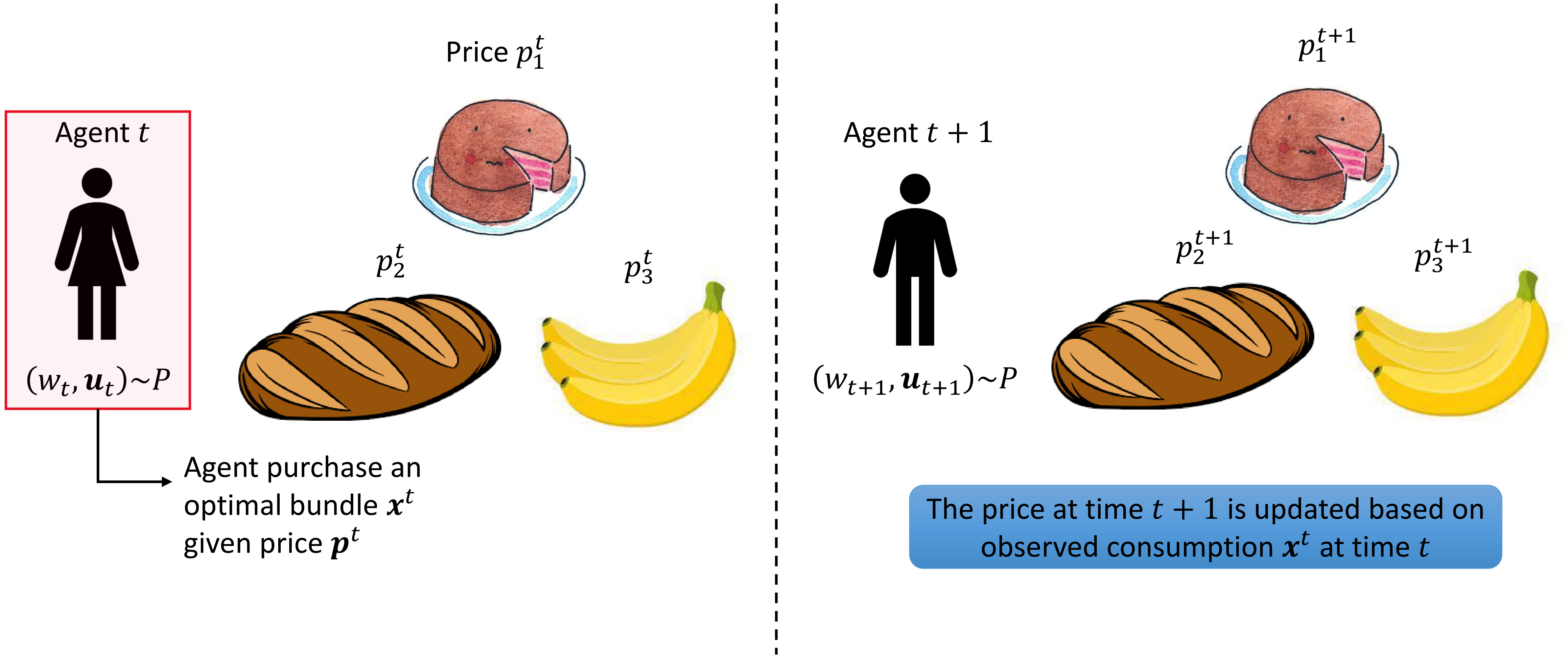
With parameters until user t arrives, we can solve the following primal problem

$$
\begin{aligned}
\max_{\mathbf{x}_i \in \mathbb{R}^m, \forall i \in [t]} \quad & \sum_{i=1}^{t} w_i \log \left(\sum_{j=1}^{m} u_{ij} x_{ij} \right) \\
\text{s.t.} \quad & \sum_{i=1}^{t} x_{ij} \leq \frac{t}{n} c_j, \quad \forall j \in [m] \\
& x_{ij} \geq 0, \quad \forall i \in [t], j \in [m]
\end{aligned}
$$

Prices can be set based on dual of capacity constraints

At each time instance, we solve a larger convex program, which may become computationally expensive in real time

We design a dual based algorithm, wherein users see posted prices at each time they arrive and make buy decisions (no need to worry truthfulness)



Applying gradient descent to the dual of the social optimization problem motivates a natural algorithm

Dual of social optimization problem with Lagrange multiplier of the capacity constraints p_j

$$\min_{\mathbf{p}} \sum_{t=1}^{n} w_t \log(w_t) - \sum_{t=1}^{n} w_t \log\left(\min_{j\in[m]} \frac{p_j}{u_{tj}}\right) + \sum_{j=1}^{m} p_j c_j - \sum_{t=1}^{n} w_t$$

Equivalent Sample Average Approximation (SAA) of Dual Problem

$$\min_{\mathbf{p}} \quad D_n(\mathbf{p}) = \sum_{j=1}^{m} p_j \frac{c_j}{n} + \frac{1}{n}\sum_{t=1}^{n}\left(w_t \log(w_t) - w_t \log(\min_{j\in[m]} \frac{p_j}{u_{tj}}) - w_t\right)$$

(Sub)-gradient descent of dual problem for each agent: $O(m)$ complexity of price update

$$\partial_{\mathbf{p}}\left(\sum_{j\in[m]} p_j \frac{c_j}{n} + w\log(w) - w\log\left(\min_{j\in[m]}\frac{p_j}{u_j}\right) - w\right)\Bigg|_{\mathbf{p}=\mathbf{p}^t} = \frac{1}{n}\mathbf{c} - \mathbf{x}_t$$

Difference between market share of each agent and goods purchased

The privacy-preserving algorithm has sub-linear regret and constraint violation guarantees

Algorithm 1: Privacy Preserving Online Algorithm

Input : Number of users n, Vector of good capacities $\mathbf{c}$

Initialize $\mathbf{p}^1 > \mathbf{0}$;

for $t = 1, 2, ..., n$ **do**

  Phase I: User Optimization

  Each agent purchases an optimal bundle of goods $\mathbf{x}_t$ given the price $\mathbf{p}^t$;

  Phase II: Price Update

  $\mathbf{p}^{t+1} \leftarrow \mathbf{p}^t - \gamma_t \left(\frac{\mathbf{c}}{n} - \mathbf{x}_t\right)$;

end

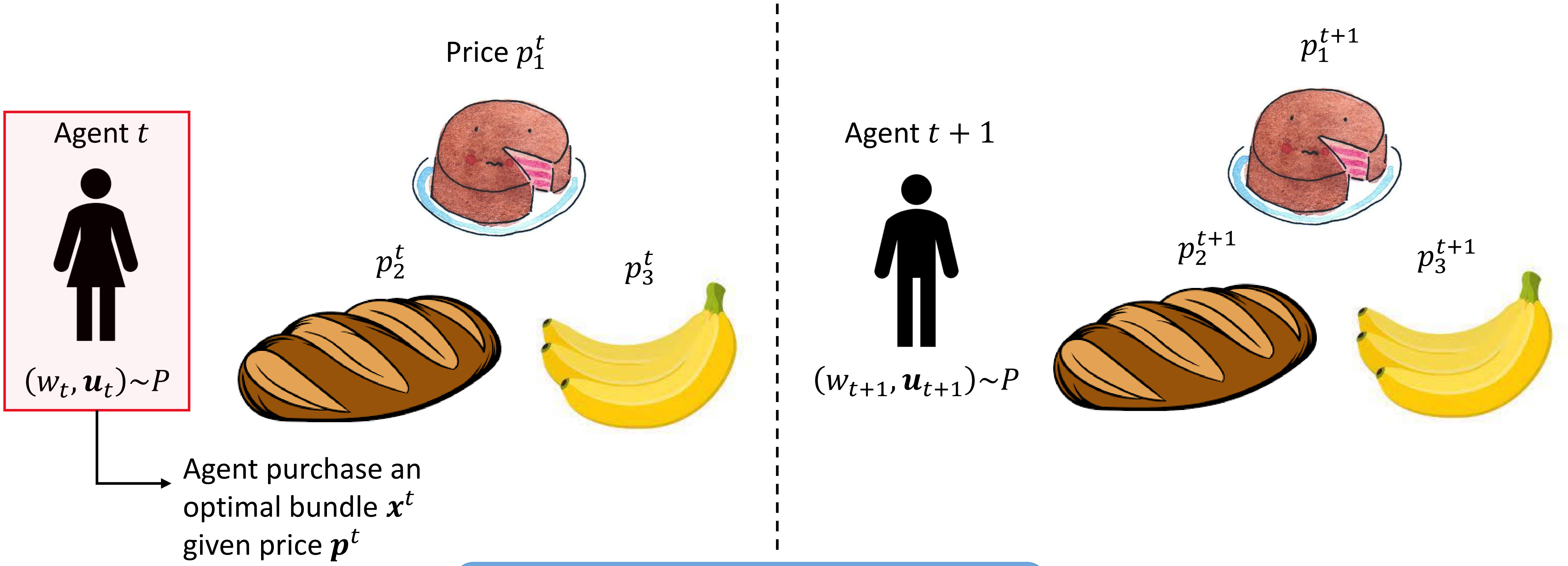
Difference between market share of each agent and goods purchased

Step-size: 1/\sqrt{n}

Only requires knowledge of user consumption (and not their budgets or utilities) to update prices

Theorem [Jelota & Y 2022]: Under i.i.d. budget and utility parameters and when good capacities are $O(n)$, Algorithm 1 achieves an expected regret $R_n(\boldsymbol{\pi}) \leq O(\sqrt{n})$ and the expected constraint violation $V_n(\boldsymbol{\pi}) \leq O(\sqrt{n})$, where n is the number of arriving users.

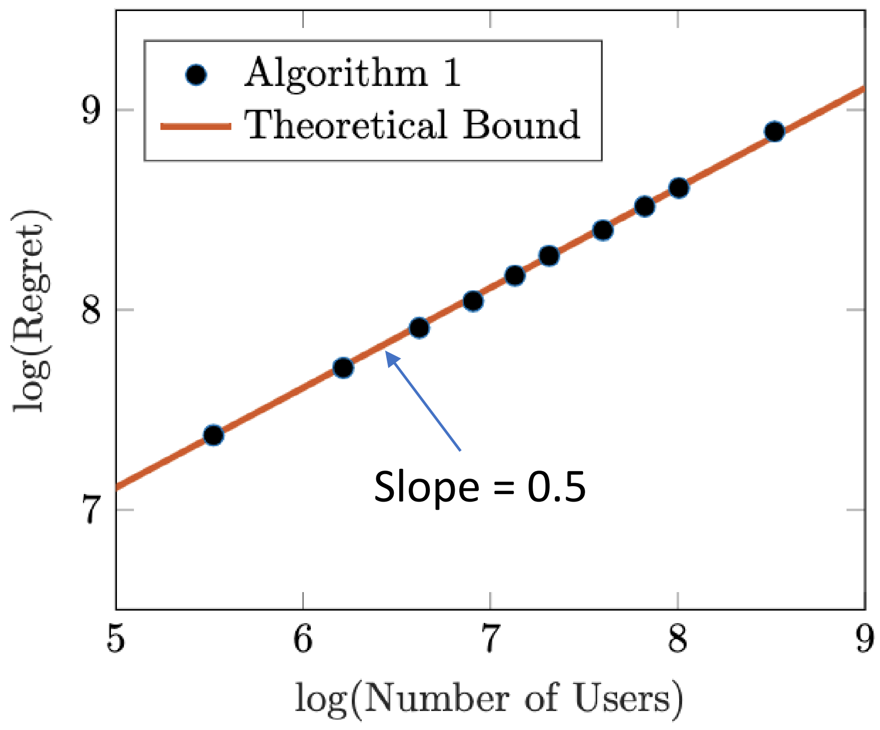
Again, the price of a good is increased if the arriving user purchase more than its market share of the good

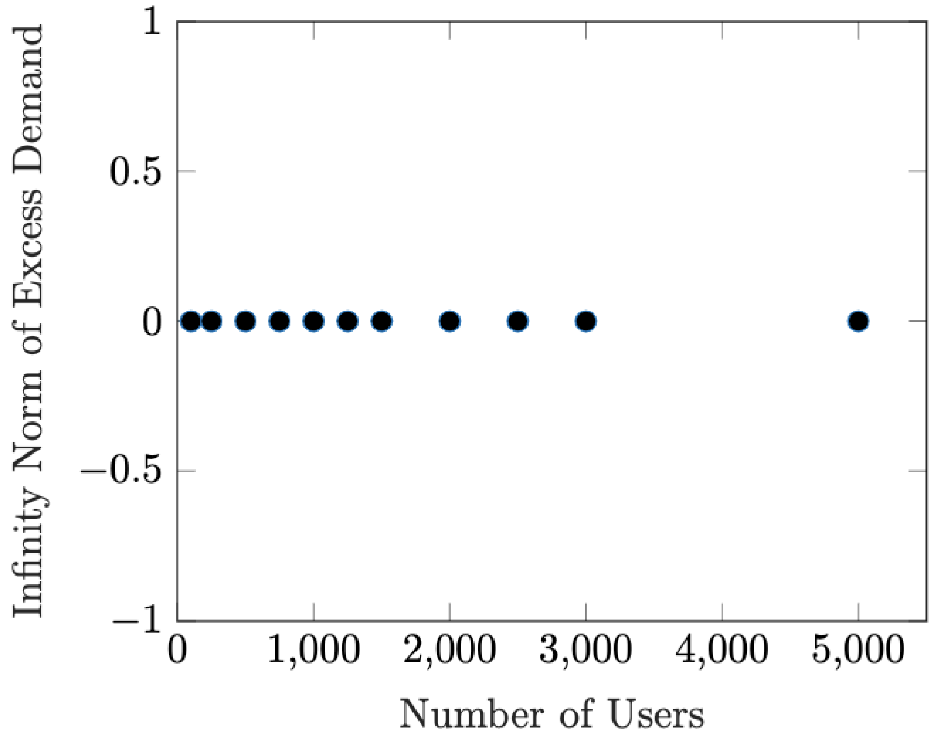


Increase Prices: $p_j^{t+1} > p_j^t$ if $x_j^{t+1} > \frac{c_j}{n}$

Decrease Prices: $p_j^{t+1} < p_j^t$ if $x_j^{t+1} < \frac{c_j}{n}$

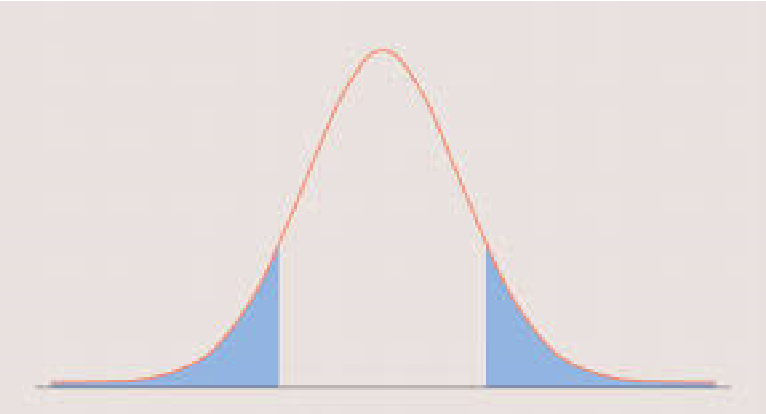
Our numerical results verify the obtained theoretical guarantee





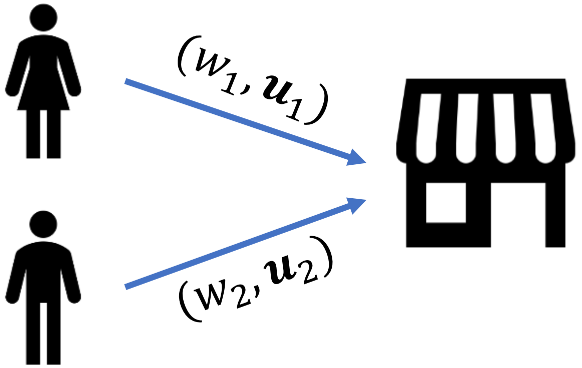
We also develop benchmarks that have access to more information to compare our algorithm's performance

Known Probability Distribution



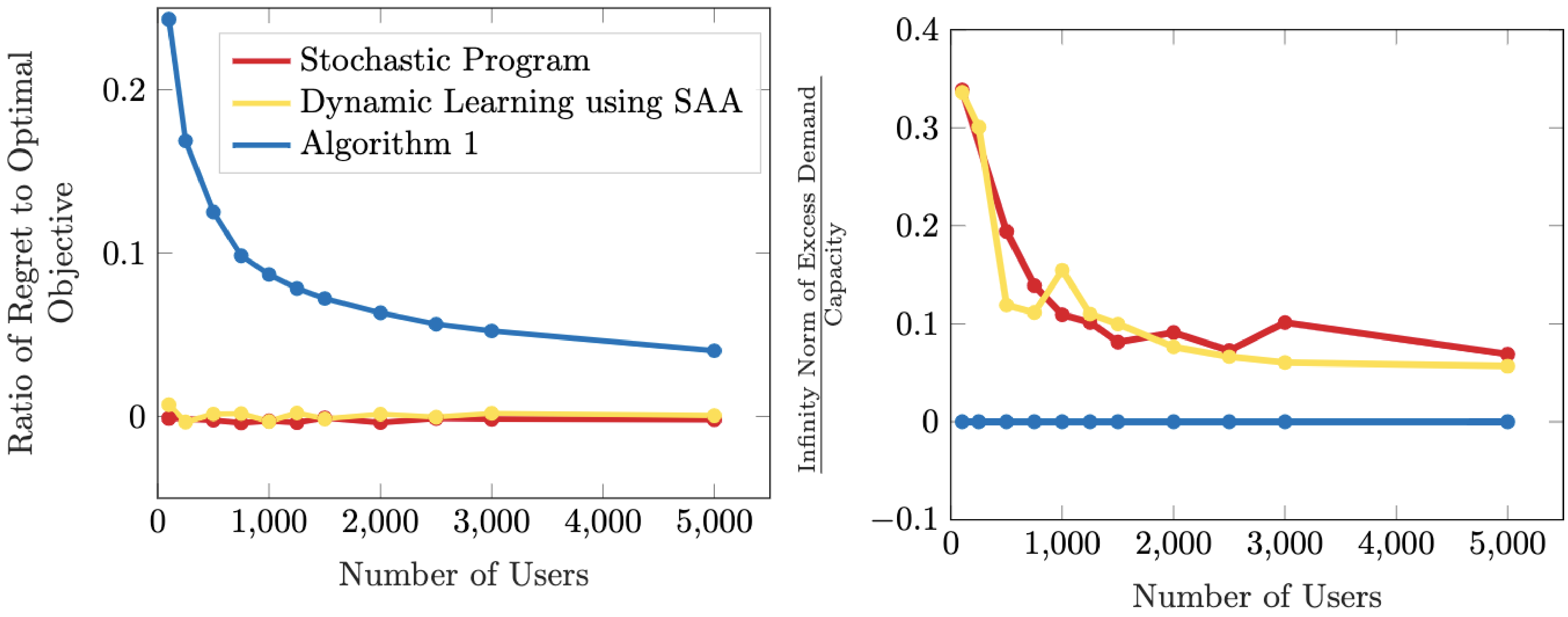
Benchmark 1: Set price based on solution of Stochastic Program

User parameters $(w, \boldsymbol{u})$ are revealed

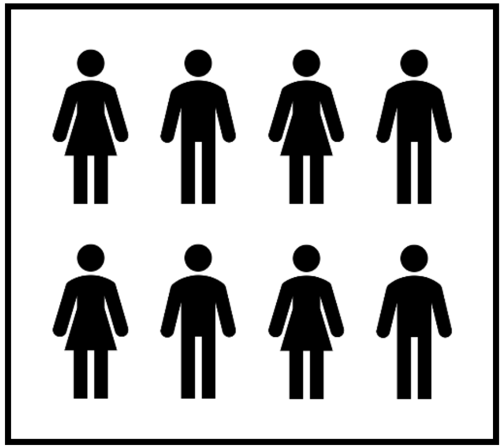


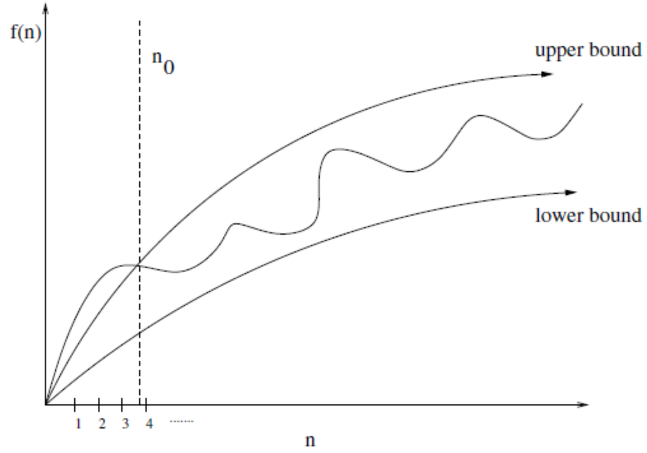
Benchmark 2: Set prices based on a sequence of dual problems using revealed parameters

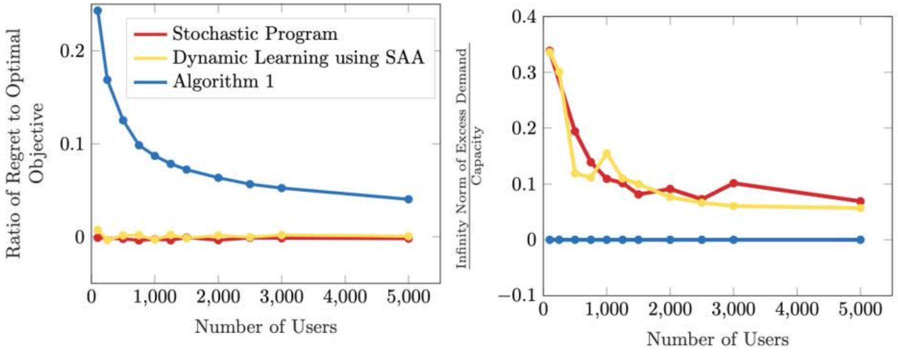
Our numerical results demonstrate a tradeoff between regret and constraint violation



Summary: online algorithms are applicable to Fisher markets with geometric aggregation of social welfare and sub-linear regret guarantees







Buyers arrive sequentially with utility and budget parameters drawn as $(w, \mathbf{u}) \overset{i.i.d.}{\sim} \mathcal{P}$

There is a fundamental trade-off between regret and constraint violation metrics

Online Algorithm with sub-linear regret and constraint violation guarantees

Organization

- Advantages of (Weighted) Geometric Mean Objective
- Distributed ADMM Algorithm for Fisher Markets (Simulated Market)
- Online Fisher Markets (Real Market)
- **Summaries**

Geometrically aggregated welfare optimization: it is as easy as linear programming and more desirable in many social/economical settings

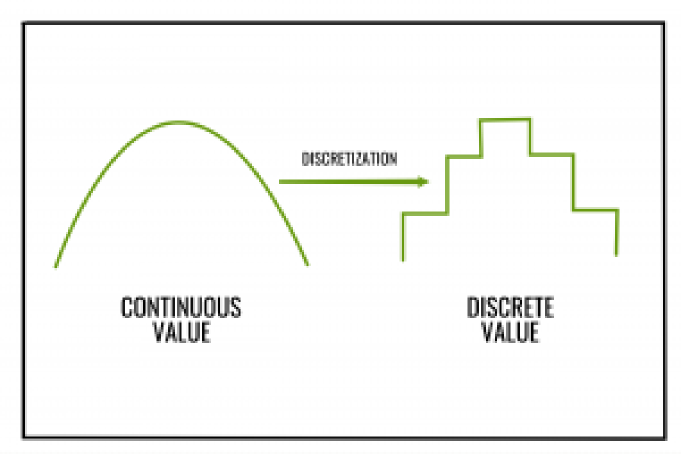
The weighted geometric average objective has several advantages including fairness, computational complexity, and the resulting allocation can be distributed using prices through Fisher markets

The Nash social welfare maximizing allocations can be computed in a distributed fashion by using the primal-dual and/or ADMM methods while preserving the privacy of individual utilities

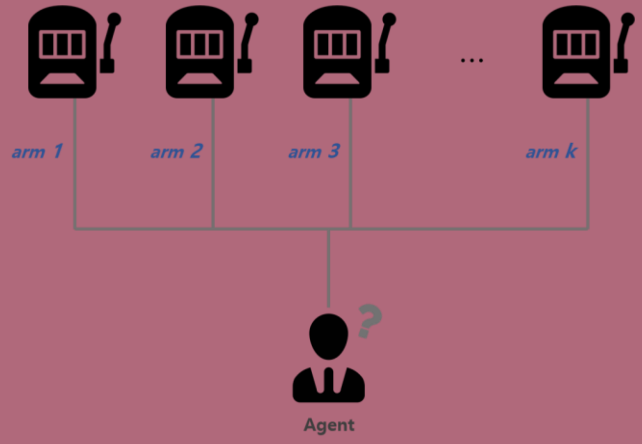
The corresponding allocations can be implemented in the online setting with a sublinear regret

Future Work

Loss in social objective under integral allocations



Extensions of geometric social objective for online allocation in bandit and reinforcement learning problems



Extension of online Fisher markets under general concave utility functions and tight regret bounds