Geometric Aggregation of the Social Welfare Function in Resource Allocation

Yinyu Ye
Stanford University
April 8, 2022

Joint work with Devansh Jalota

ACO Annual Distinguished Lecture @UCI

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

Public Good Allocation

Vaccine Allocation

A key question is how to aggregate society's preferences to reflect a fair division of resources

Efficiency Objective

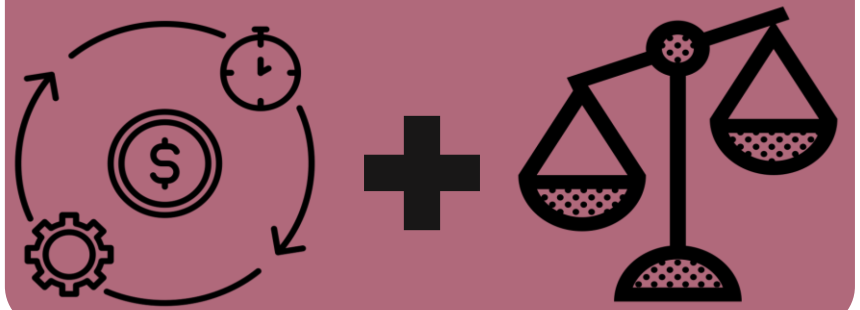
$$\max \sum_i w_i U_i(\boldsymbol{x}_i)$$

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

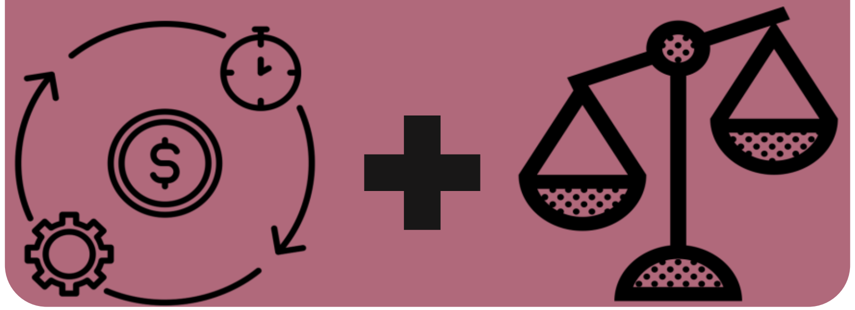
$$\max \sum_i w_i U_i(\boldsymbol{x}_i)$$

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

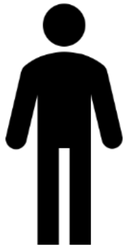
- Advantages/Properties of (Weighted) Geometric Mean Objective
- Distributed ADMM Algorithm for Fisher Markets (Simulated Market)
- Online Fisher Markets (Real Market)
- Conclusion

Organization

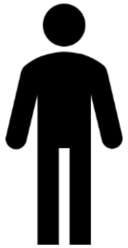
- **Advantages/Properties of (Weighted) Geometric Mean Objective**
- Distributed ADMM Algorithm for Fisher Markets (Simulated Market)
- Online Fisher Markets (Real Market)
- Conclusion

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

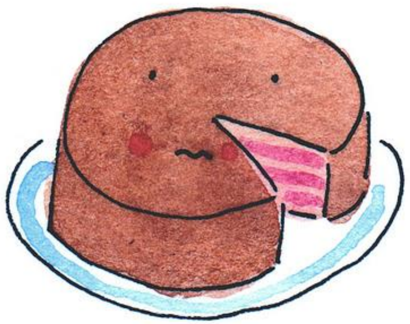
2 Agents



$u_{11} > u_{12}$



1 unit of a divisible resource



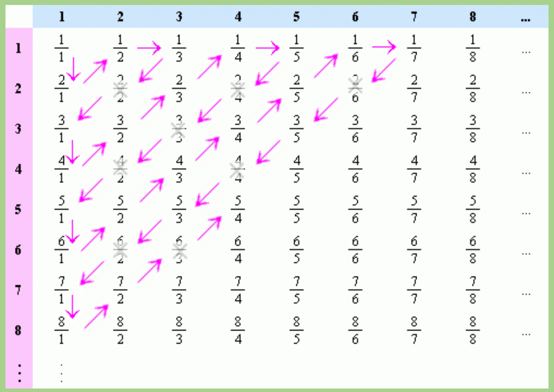
u_{ij} : Preference of Agent i for one unit of good j

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

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

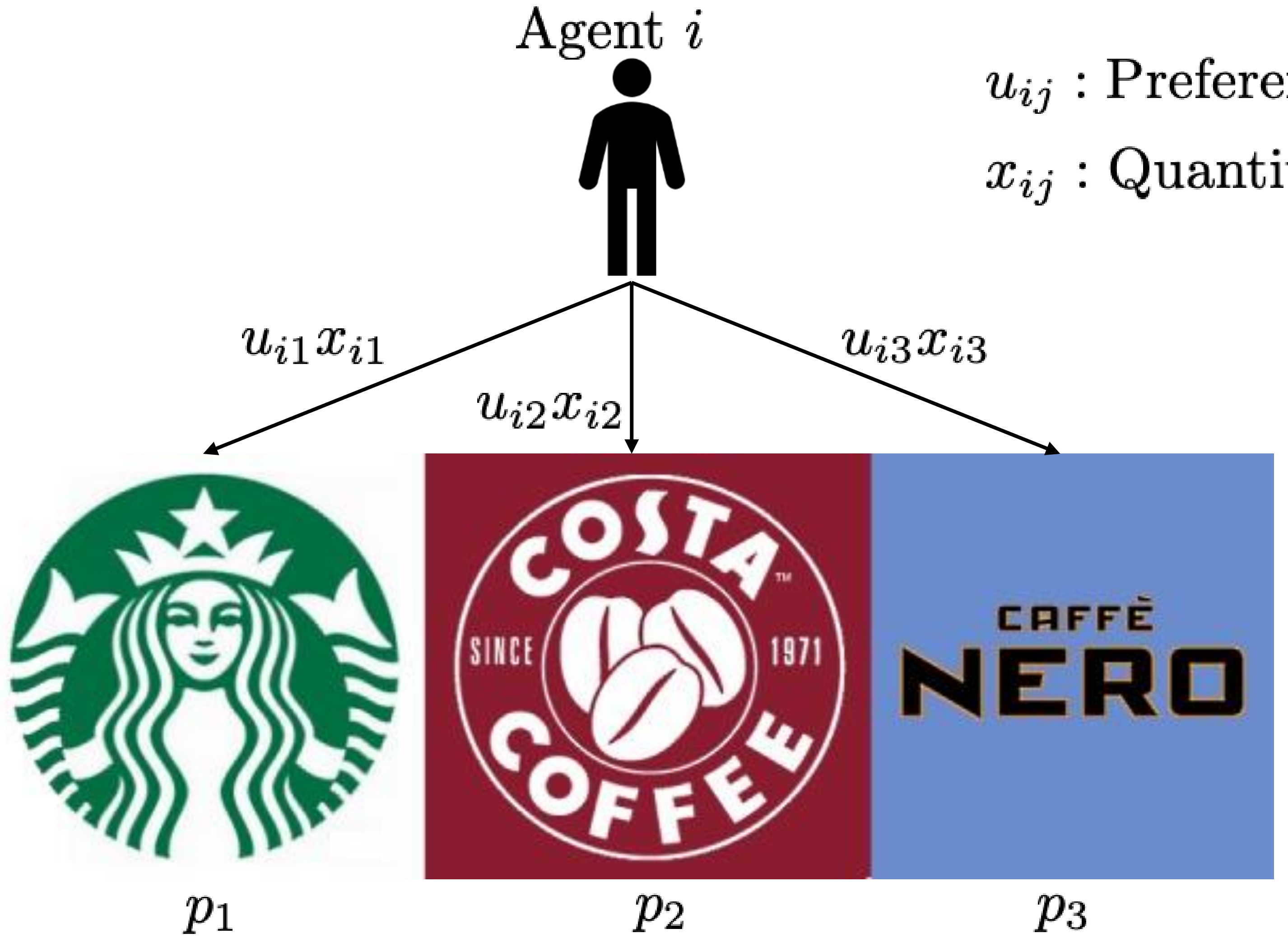
The resulting allocation is Pareto efficient

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

Individual Optimization Problem:

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

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

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

$$\Longleftrightarrow$$

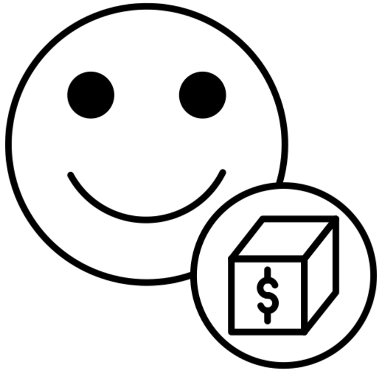
Social Optimization Problem:

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

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

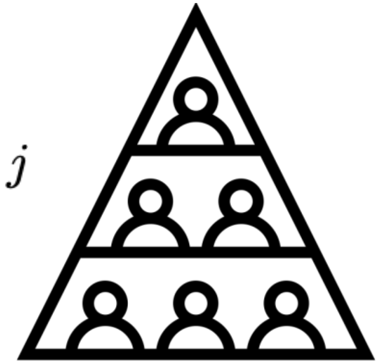
Capacity Constraints

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



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

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



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

Individual Optimization Problem:

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

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

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

$$\Longleftrightarrow$$

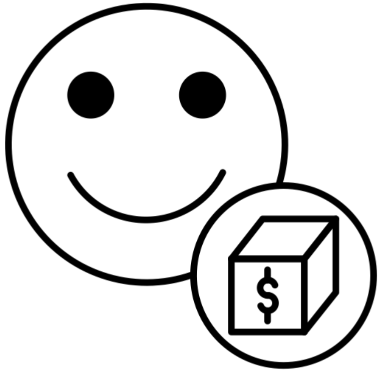
Social Optimization Problem:

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

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

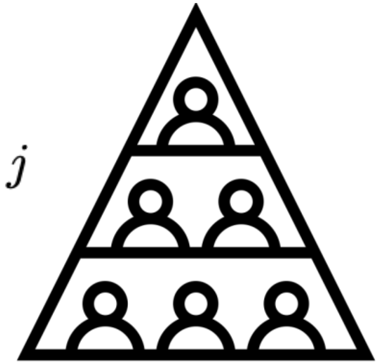
Capacity Constraints

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

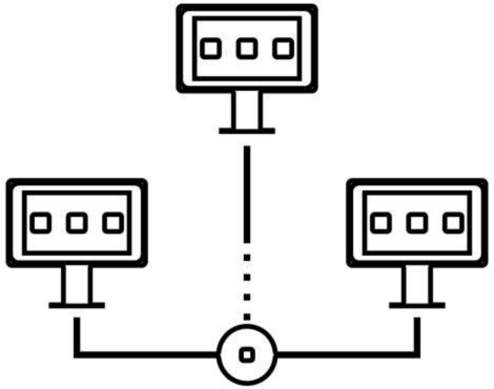


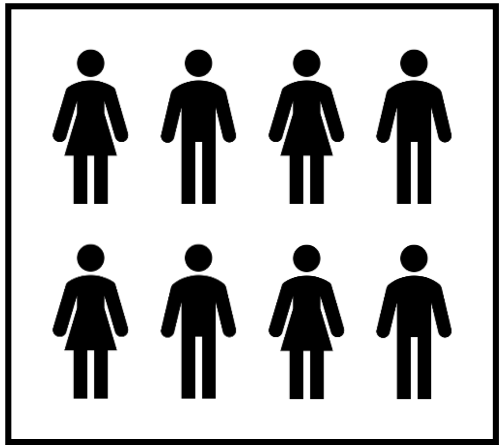
p_j : Price of Good j = Dual Variable of Constraint j

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



Distributed algorithms for Fisher markets and show that it can be implemented in an online setting





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

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

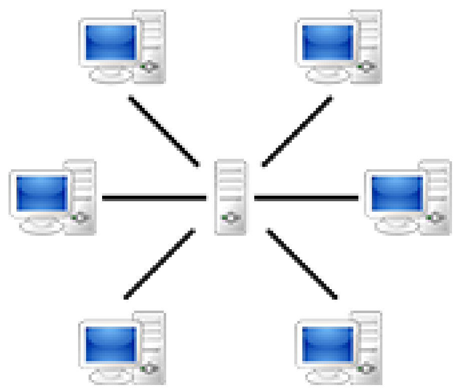
Buyers arrive sequentially with utility and budget parameters in real time

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

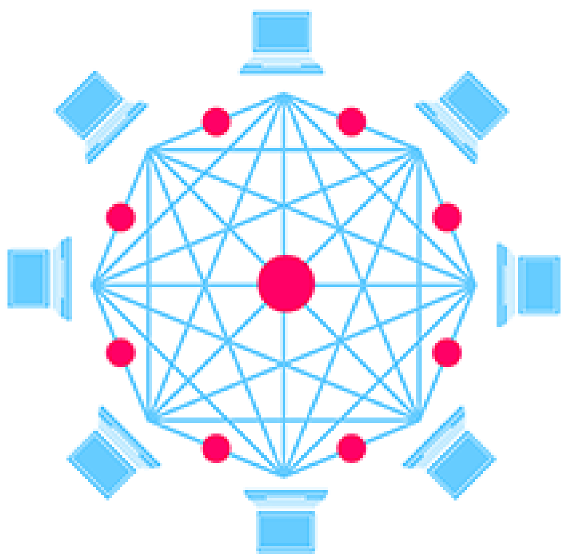
Organization

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

Distributed algorithms for Fisher markets are necessary since the utilities of buyers may not be known

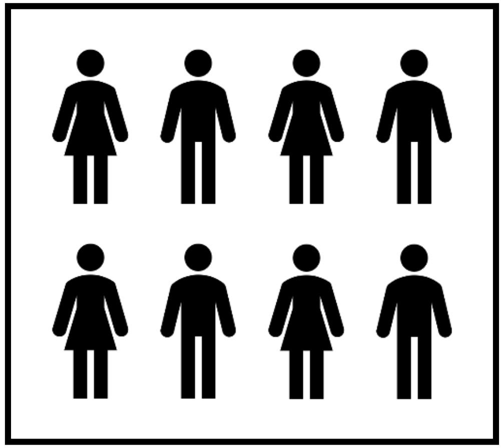


Centralized

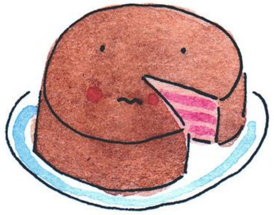


Distributed

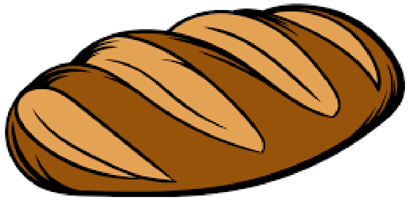
Review: Primal-Dual (Tatonnement) methods adjust prices based on discrepancy between supply & demand



Each agent i purchases an optimal bundle $\boldsymbol{x}_i^t$ given price $\boldsymbol{p}^t$

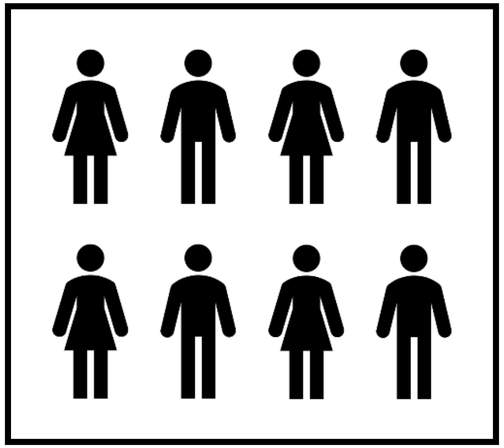


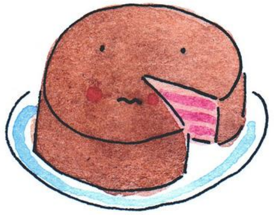
Price p_1^t



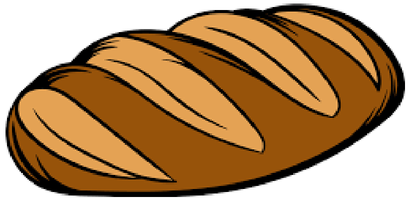
p_2^t

p_3^t





p_1^{t+1}



p_2^{t+1}

p_3^{t+1}

The price at time $t+1$ is updated based on observed consumptions $\boldsymbol{x}_i^t$ at time t

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 $\sum_i x_{ij}^t > c_j$

Review: Using primal-dual methods, convergence is only guaranteed for strongly concave utilities

Each agent solves their individual optimization problem

Prices Updated based on discrepancy between demand and supply

Algorithm 1: Tatonnement for Fisher Markets

Input : Initial price vector $\mathbf{p}$

Output: Equilibrium Price vector $\mathbf{p}^*$

for $k = 0, 1, 2, \ldots$ **do**

$\quad \mathbf{x}_i^{(k+1)} = \arg\max_{\mathbf{x}_i \in \mathcal{X}_i} \{w_i \log(u_i(\mathbf{x}_i)) - \sum_j p_j^{(k)} x_{ij}\}$, for all i ;

$\quad \mathbf{p}^{(k+1)} \leftarrow \mathbf{p}^{(k)} + \beta(\sum_i \mathbf{x}_i^{(k+1)} - \mathbf{c})$;

end

Theorem [Cole, Fleischer, 2008]
If the objective function is strongly concave, the convergence of the tatonnement algorithm to the optimal solution is linear

Review: Furthermore, the step-size of the price updates often depends on the type of utility function

Each agent solves their individual optimization problem

Prices Updated based on discrepancy between demand and supply

Algorithm 1: Tatonnement for Fisher Markets

Input : Initial price vector $\mathbf{p}$

Output: Equilibrium Price vector $\mathbf{p}^*$

for $k = 0, 1, 2, \dots$ **do**

$\quad \mathbf{x}_i^{(k+1)} = \arg\max_{\mathbf{x}_i \in \mathcal{X}_i} \{w_i \log(u_i(\mathbf{x}_i)) - \sum_j p_j^{(k)} x_{ij}\}$, for all i ;

$\quad \mathbf{p}^{(k+1)} \leftarrow \mathbf{p}^{(k)} + \beta(\sum_i \mathbf{x}_i^{(k+1)} - \mathbf{c})$;

end

Theorem [Cole, Fleischer, 2008]
If the objective function is strongly concave, the convergence of the tatonnement algorithm to the optimal solution is linear

We introduce ADMM, where a regularization term is added to obtain better convergence guarantees

$$\max_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} \quad h(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) + g(\mathbf{y})$$

$$\text{s.t.} \quad A\mathbf{x} + B\mathbf{y} = \mathbf{c}$$

$$\mathcal{L}_\beta(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) + g(\mathbf{y}) - \mu^T(A\mathbf{x} + B\mathbf{y} - \mathbf{c}) - \frac{\beta}{2}\|A\mathbf{x} + B\mathbf{y} - \mathbf{c}\|^2$$

Penalty for constraint violation

Algorithm 2: Two Block ADMM

Input : Initial dual multiplier $\lambda^{(0)}$, and initial vector $\mathbf{y}^{(0)}$

for $k = 0, 1, 2, \ldots$ **do**

$\quad \mathbf{x}^{(k+1)} = \arg\max_{\mathbf{x}\in\mathcal{X}} \mathcal{L}_\beta(\mathbf{x}, \mathbf{y}^{(k)})$;

$\quad \mathbf{y}^{(k+1)} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}_\beta(\mathbf{x}^{(k+1)}, \mathbf{y})$;

$\quad \mu^{(k+1)} \leftarrow \mu^{(k)} + \beta(A\mathbf{x}^{(k+1)} + B\mathbf{y}^{(k+1)} - \mathbf{c})$;

end

Dual variable of constraint

Glowinski&Marroco, 1975

Theorem [He&Yuan and Monteiro&Svaiter, 2010]: If the objective function is (weakly) concave, then ADMM converges to the optimal solution with rate $O(\frac{1}{k})$, where k is the number of iterations of the algorithm.
Under strong concavity assumptions, the convergence is linear.

The step-size of the price updates is independent of the utility functions of users

$$\max_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}} \quad h(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{y})$$

$$\text{s.t.} \quad A\mathbf{x} + B\mathbf{y} = \mathbf{c}$$

$$\mathcal{L}_\beta(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{y}) - \mu^T(A\mathbf{x} + B\mathbf{y} - \mathbf{c}) - \underbrace{\frac{\beta}{2}\|A\mathbf{x} + B\mathbf{y} - \mathbf{c}\|^2}_{\text{Penalty for constraint violation}}$$

Algorithm 2: Two Block ADMM

Input : Initial dual multiplier $\lambda^{(0)}$, and initial vector $\mathbf{y}^{(0)}$

for $k = 0, 1, 2, \ldots$ **do**

$\quad \mathbf{x}^{(k+1)} = \arg\max_{\mathbf{x} \in \mathcal{X}} \mathcal{L}_\beta(\mathbf{x}, \mathbf{y}^{(k)})$;

$\quad \mathbf{y}^{(k+1)} = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{L}_\beta(\mathbf{x}^{(k+1)}, \mathbf{y})$;

$\quad \mu^{(k+1)} \leftarrow \mu^{(k)} + \beta(A\mathbf{x}^{(k+1)} + B\mathbf{y}^{(k+1)} - \mathbf{c})$;

end

Dual variable of constraint

Theorem [He&Yuan and Monteiro&Svaiter, 2010]: If the objective function is (weakly) concave, then ADMM converges to the optimal solution with rate $O(\frac{1}{k})$, where k is the number of iterations of the algorithm.

Under strong concavity assumptions, the convergence is linear.

To apply ADMM for Fisher markets, we add an additional variable to achieve a distributed implementation

$$
\begin{aligned}
\max_{\mathbf{x}_i \in \mathcal{X}_i, \forall i \in [n]} \quad & \sum_i w_i \log\left(u_i(\mathbf{x}_i)\right), \\
\text{s.t.} \quad & \sum_i x_{ij} = c_j, \forall j \in [m].
\end{aligned}
$$

Add an additional variable to achieve distributed ADMM implementation

$$
\begin{aligned}
\max_{\mathbf{x}_i \in \mathcal{X}_i, \mathbf{y}_i \in \mathcal{Y}_i, \forall i \in [n]} \quad & \sum_i w_i \log\left(u_i(\mathbf{x}_i)\right), \\
\text{s.t.} \quad & \sum_i y_{ij} = c_j, \forall j \in [m], \\
& \mathbf{y}_i = \mathbf{x}_i, \forall i \in [n].
\end{aligned}
$$

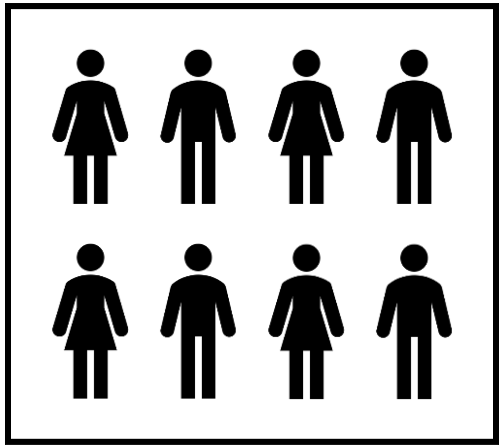
Each agent solves a "regularized" individual optimization problem

Prices Updated based on discrepancy between demand and supply

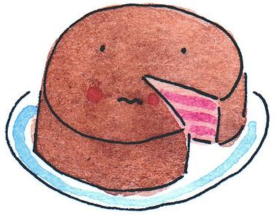
Algorithm 1: Two Block ADMM for Fisher Markets

Input : Initial price vector $\mathbf{p}$, and initial baseline demand $\mathbf{y}_i^{(0)}$
Output: Equilibrium Price vector $\mathbf{p}^*$
for $k = 0, 1, 2, \ldots$ **do**
  $\mathbf{x}_i^{(k+1)} = \arg\max_{\mathbf{x}_i \in \mathcal{X}_i} \{w_i \log\left(u_i(\mathbf{x}_i)\right) - \sum_j p_j^{(k)} x_{ij} - \frac{\beta}{2} \sum_{i,j} (x_{ij} - y_{ij}^{(k)})^2\}$, for all i ;
  $\mathbf{y}^{(k+1)} = \arg\max_{\mathbf{y}} \{-\frac{\beta}{2} \sum_{i,j} (x_{ij}^{(k+1)} - y_{ij})^2 - \frac{\beta}{2} \sum_j \left(\sum_i y_{ij} - c_j\right)^2\}$;
  $\mathbf{p}^{(k+1)} \leftarrow \mathbf{p}^{(k)} + \beta(\sum_i \mathbf{y}_i^{(k+1)} - \mathbf{c})$;
end

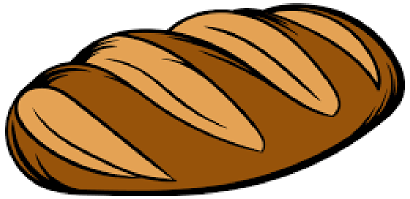
Agents again solve "regularized" objective and prices are adjusted based on discrepancy between supply & demand



Each agent i purchases an **"regularized"** optimal bundle $\boldsymbol{x}_i^t$ given price $\boldsymbol{p}^t$

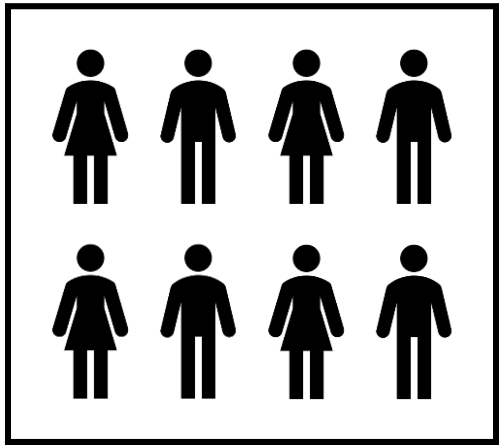


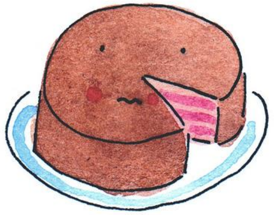
Price p_1^t



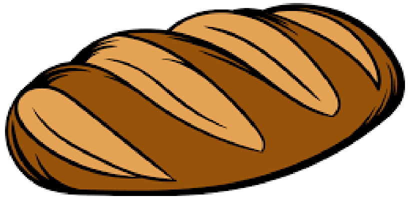
p_2^t

p_3^t





p_1^{t+1}



p_2^{t+1}

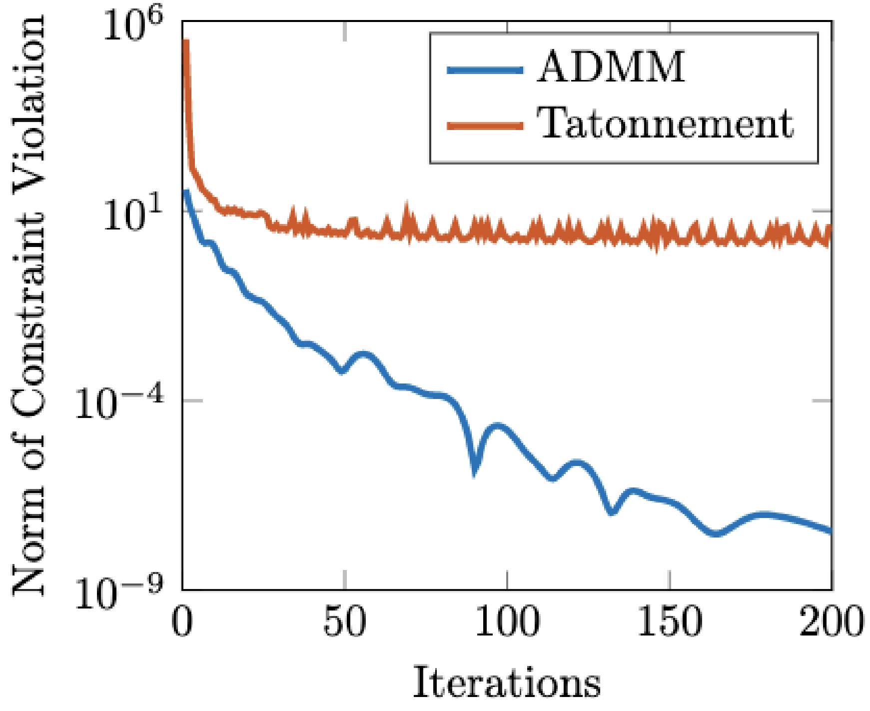
p_3^{t+1}

The price at time $t+1$ is updated based on observed consumptions $\boldsymbol{x}_i^t$ at time t

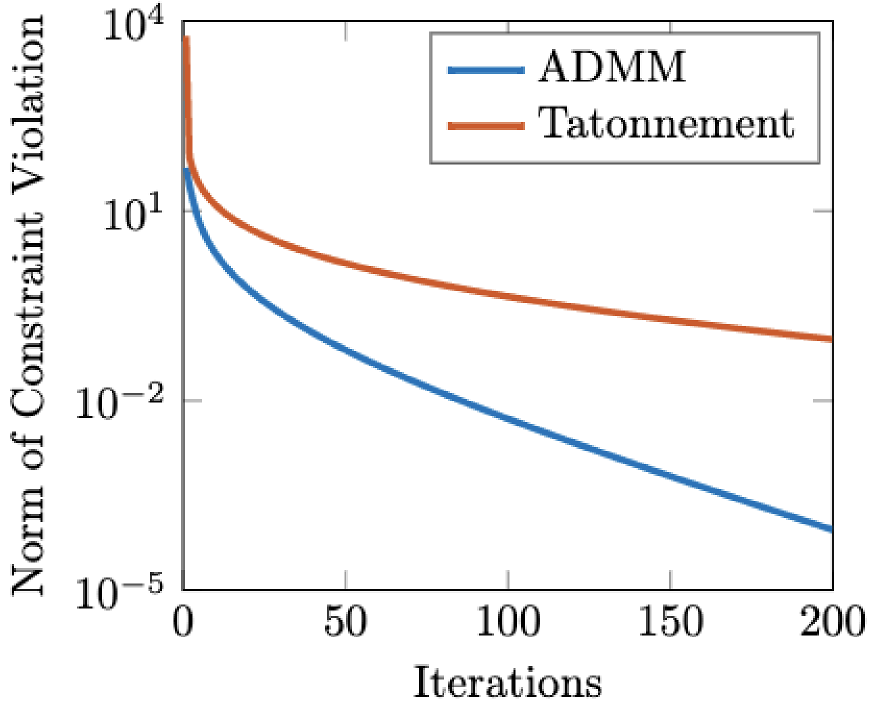
Increase Prices: $p_j^{t+1} > p_j^t$ if $\sum_i x_{ij}^t > c_j$
Decrease Prices: $p_j^{t+1} < p_j^t$ if $\sum_i x_{ij}^t > c_j$

Numerical results verify the theoretical guarantees for the two algorithms

Linear Utility (Weakly Concave)



Stone-Geary Utility (Strictly Concave)



ADMM provides strong convergence guarantees for a broad range of utility functions

ADMM converges for weakly concave utility functions, e.g., linear utilities

The step-size of the price updates is independent of the utility functions of users

ADMM can also be extended to the setting when users have additional linear constraints

Organization

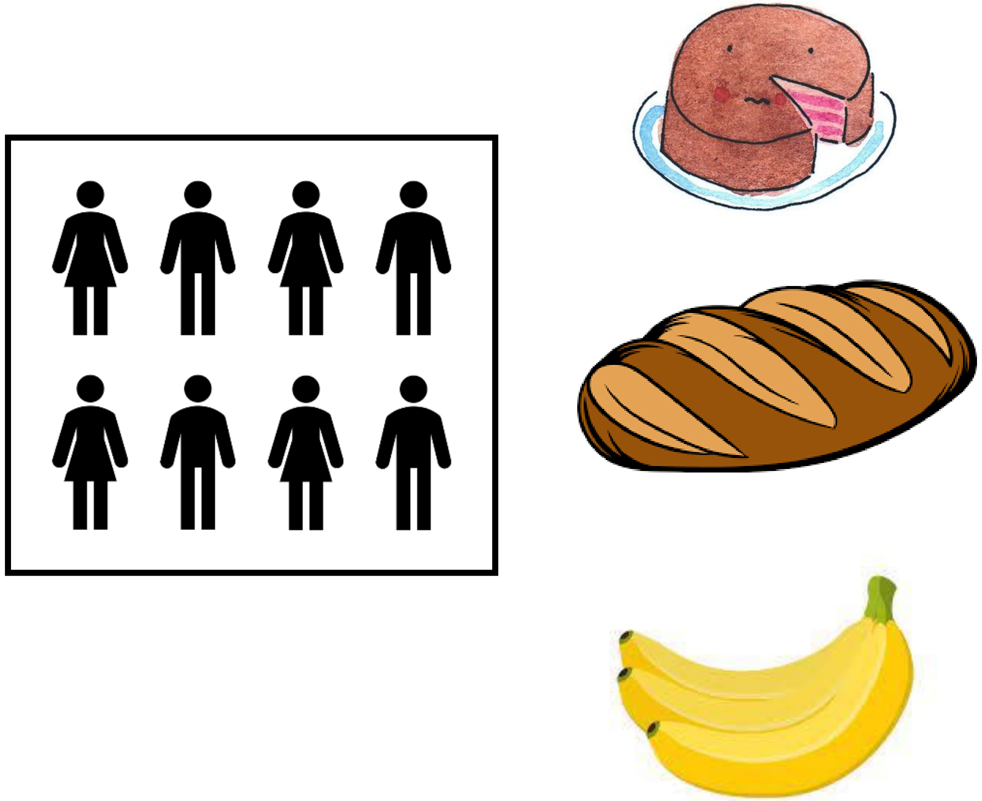
- Advantages of (Weighted) Geometric Mean Objective
- Distributed ADMM Algorithm for Fisher Markets (Simulated Market)
- Online Fisher Markets (Real Market)
- Conclusion

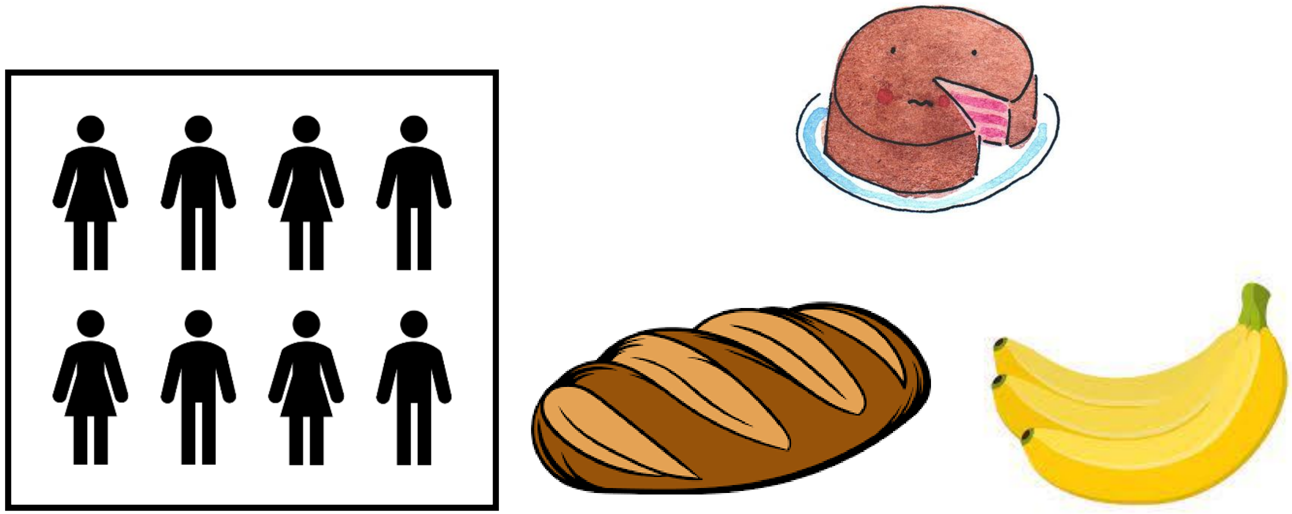
There are many settings wherein agents arrive into the market sequentially and decisions have to be made immediately

Agents obtain vaccines over time

Agents arrive over time to use public goods

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



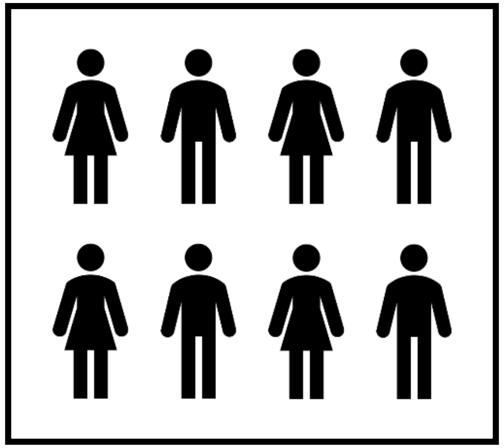


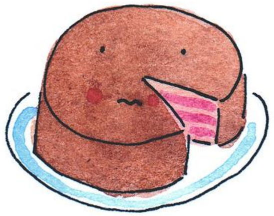
Prior Work: Goods Arrive Online
[Gorokh, Banerjee, Iyer, 2021]

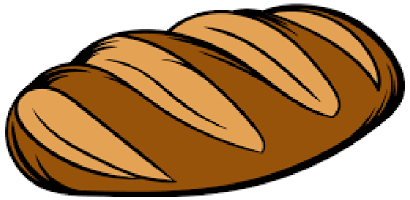
This Work: Agents arrive Online and an irrevocable allocation has to be made:
How much the objective value degraded from offline version?

The setting of agents arriving online has been studied in online linear programming (OLP)

Utility = $\sum_{j=1}^{m} u_{tj} x_{tj}$







Objective: Maximize $\sum_{t=1}^{n} \sum_{j=1}^{m} u_{tj} x_{tj}$
Subject to resource constraints

Performance of online algorithm measured with respect to regret from the offline linear objective
[Mehta et al. 2007], [Agrawal et al. 2010, 2014], [Kesselheim et al 2014]
[Li/Ye, 2019], [Li et al. 2020],

Online Linear Programming

- Traders come one by one sequentially, buy or sell, or combination, with a combinatorial order/bid $(\mathbf{a}_k, \pi_k)$
- The seller/market-maker has to make an order-fill decision as soon as an order arrives
- The seller/market-maker faces a dilemma:
  - **To accept or reject – this is the decision**
- Optimal Policy?
- The off-line problem can be an (0 1) linear program

$$
\begin{array}{ll}
\max & \sum_k \pi_k x_k \\
\text{s.t.} & \sum_k a_{ik} x_k \le b_i \quad \forall\, i \in S \\
 & 0 \le x_k \le 1 \quad \forall\, k \in N
\end{array}
$$

Off-Line LP

Regret 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 data.
- 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

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

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

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

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

Possibility Result on Regret

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

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

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

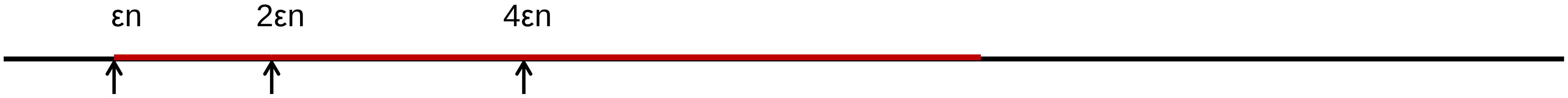
- **Learn "ideal" itemized-prices**
- **Use the prices to price each bid**
- **Accept if it is an over bid, and reject otherwise**

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.



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

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

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

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

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 $\boldsymbol{\pi}$

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

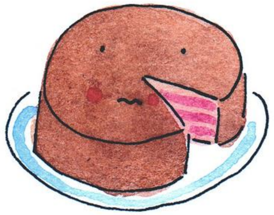
Violation of Capacity Constraint of good j

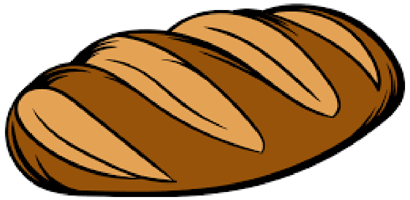
$$\boxed{V_n(\boldsymbol{\pi}) = ||\mathbb{E}[V(\boldsymbol{\pi})^+]||_2}$$

Norm of the expected constraint violation

Using the optimal expect prices, the capacity violation must be $\Omega(\sqrt{n})$, where n is the number of total agents

2 goods, each with a capacity of n

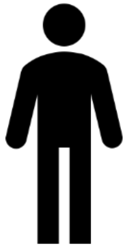


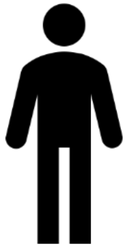


Two agent types specified by (Utility for Good 1, Utility for Good 2)

Type I: (1, 0)

Type II: (0, 1)





Arrival Probability = 0.5

Arrival Probability = 0.5

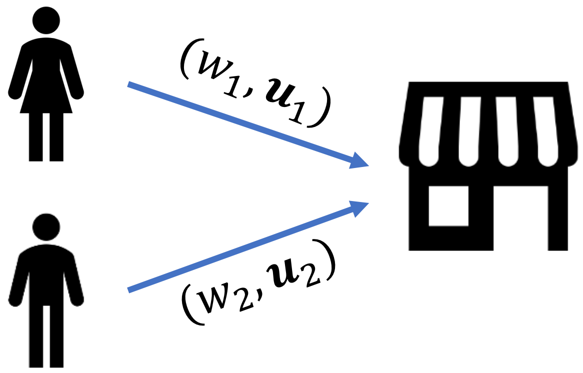
Expected Optimal Objective $\approx \boldsymbol{n\, log(2)}$
Since Type I users receive two units of good one, while type two receive two units of good two

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

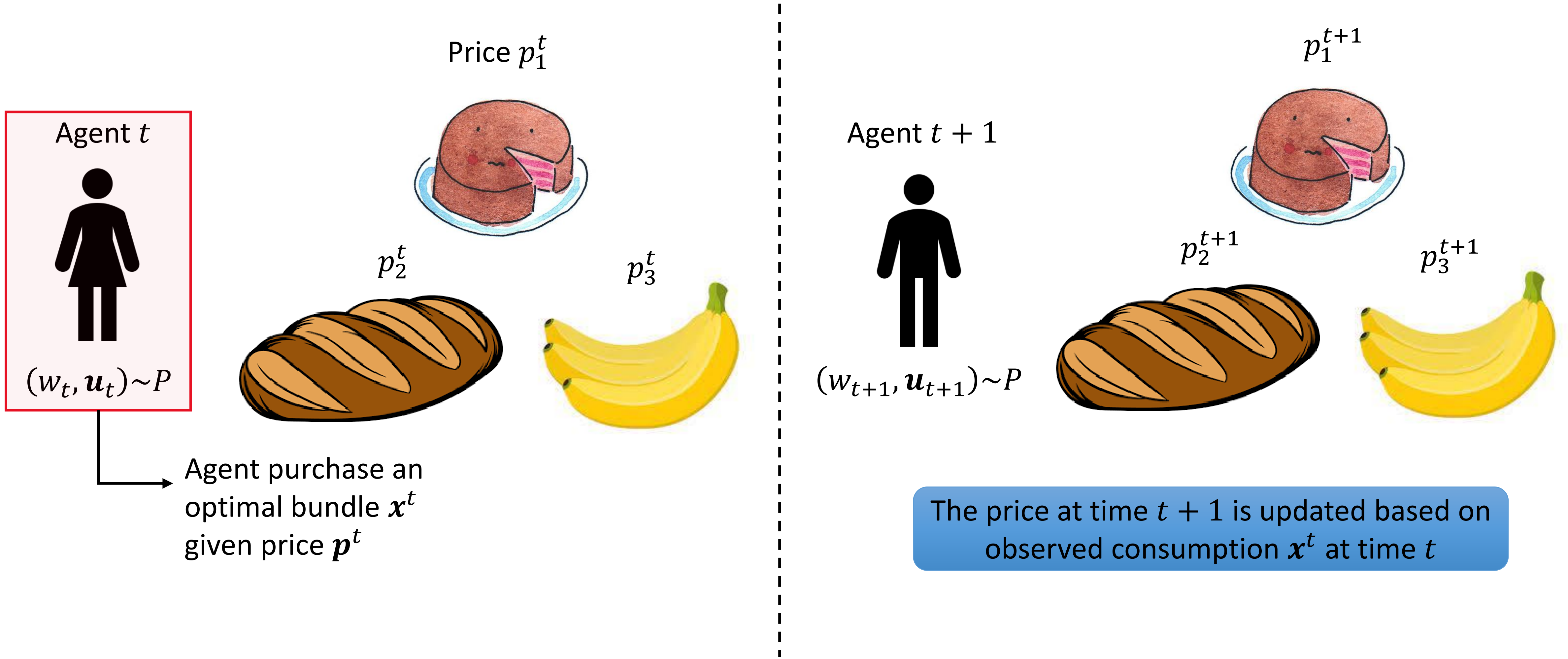
With parameters until user t arrives, we can solve the following primal problem

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

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 prices at each time they arrive



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

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

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

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

Difference between market share of each agent and goods purchased

We develop a privacy-preserving algorithm with 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

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

end

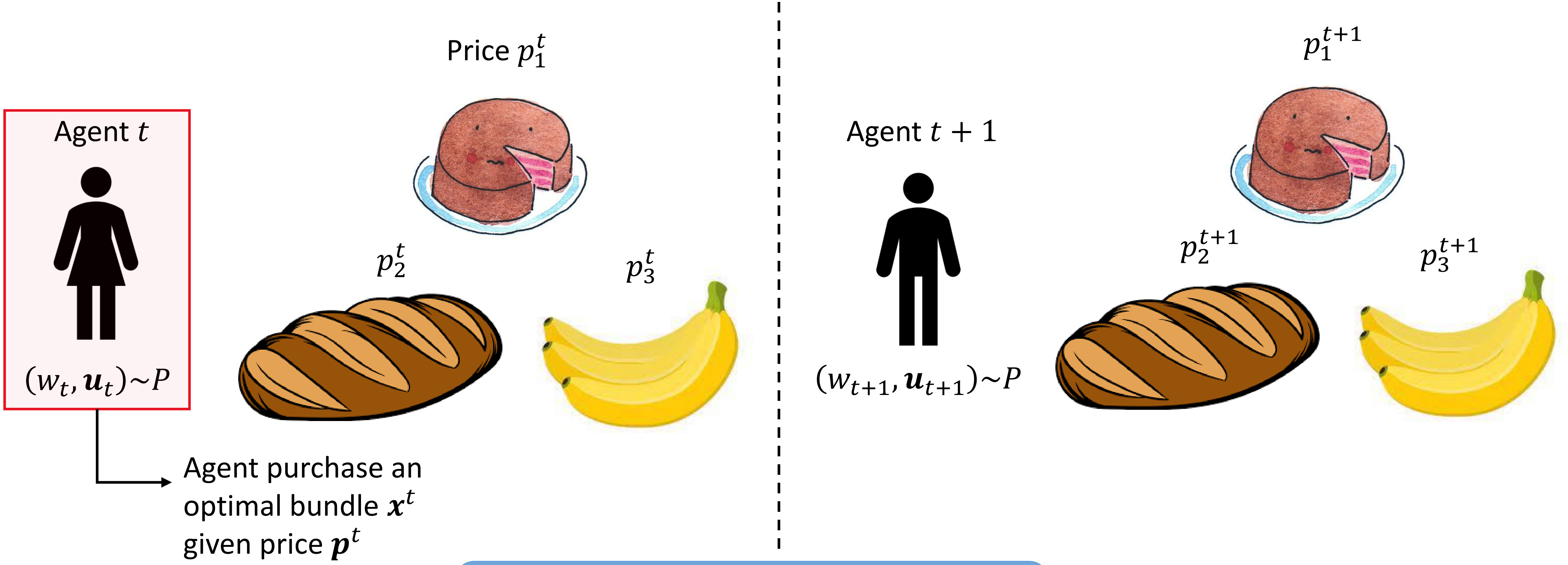
Difference between market share of each agent and goods purchased

Step-size: 1/\sqrt{n}

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

Theorem: 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.

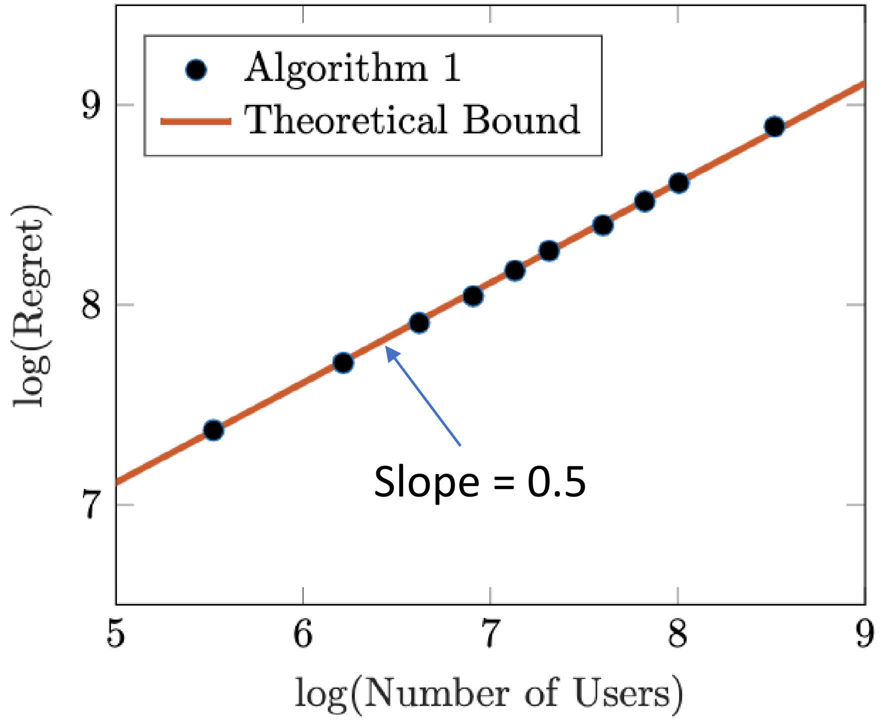
Again, the price of a good is increased if the arriving user purchase more than its market share of the good

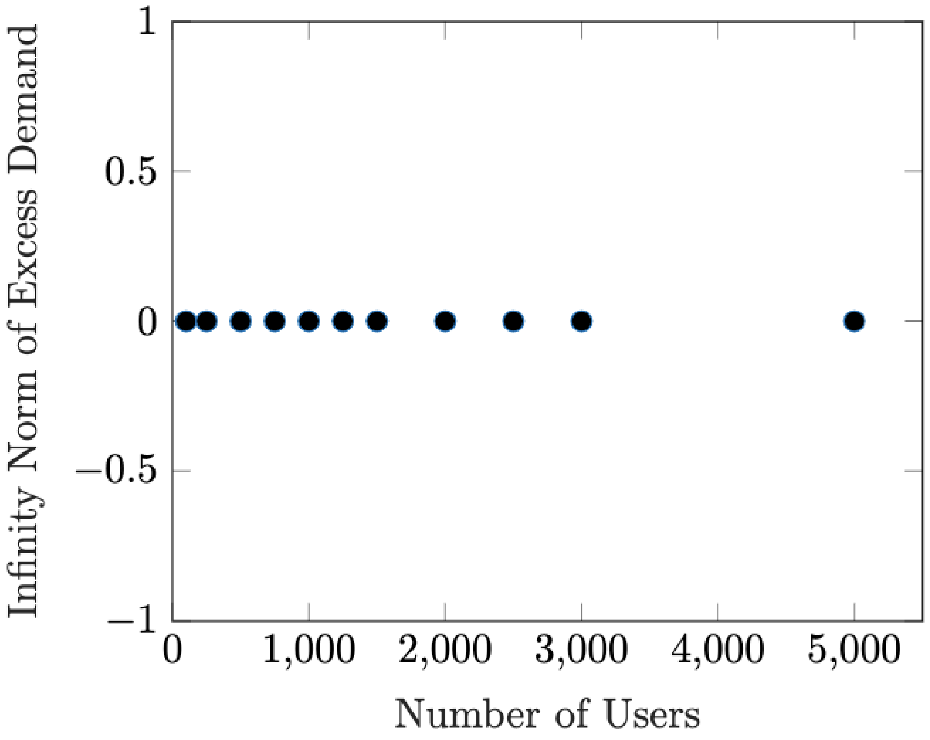


Increase Prices: $p_j^{t+1} > p_j^t$ if $x_j^{t+1} > \frac{c_j}{n}$

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

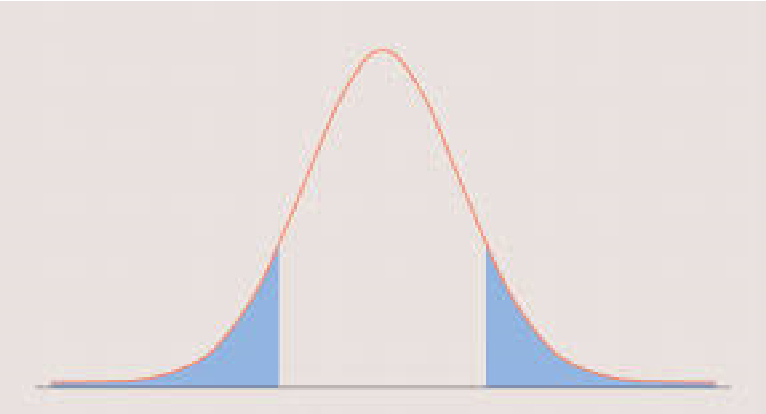
Our numerical results verify the obtained theoretical guarantee





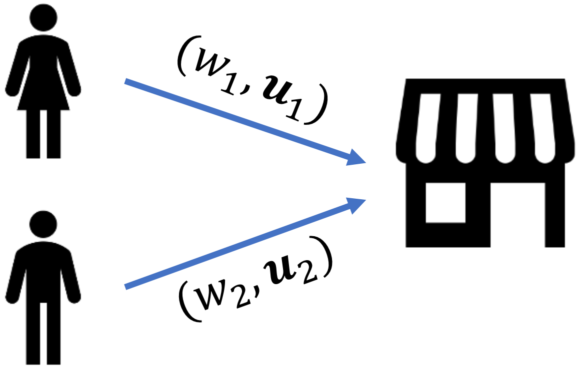
We also develop benchmarks that have access to more information to compare our algorithm's performance

Known Probability Distribution



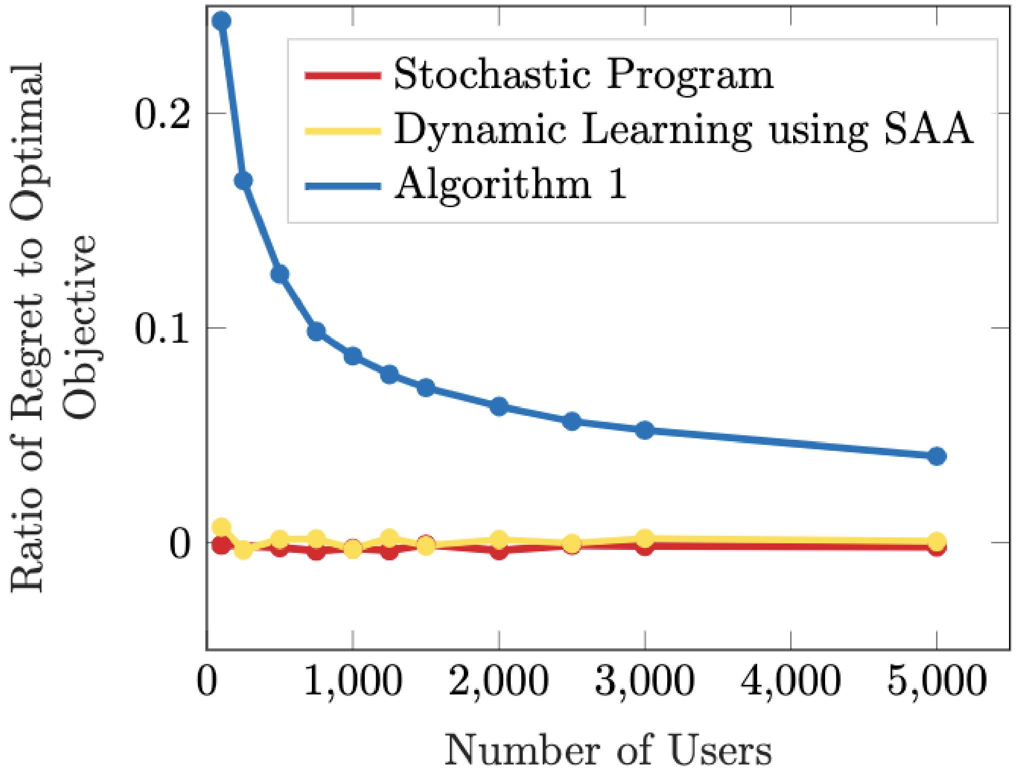
Benchmark 1: Set price based on solution of Stochastic Program

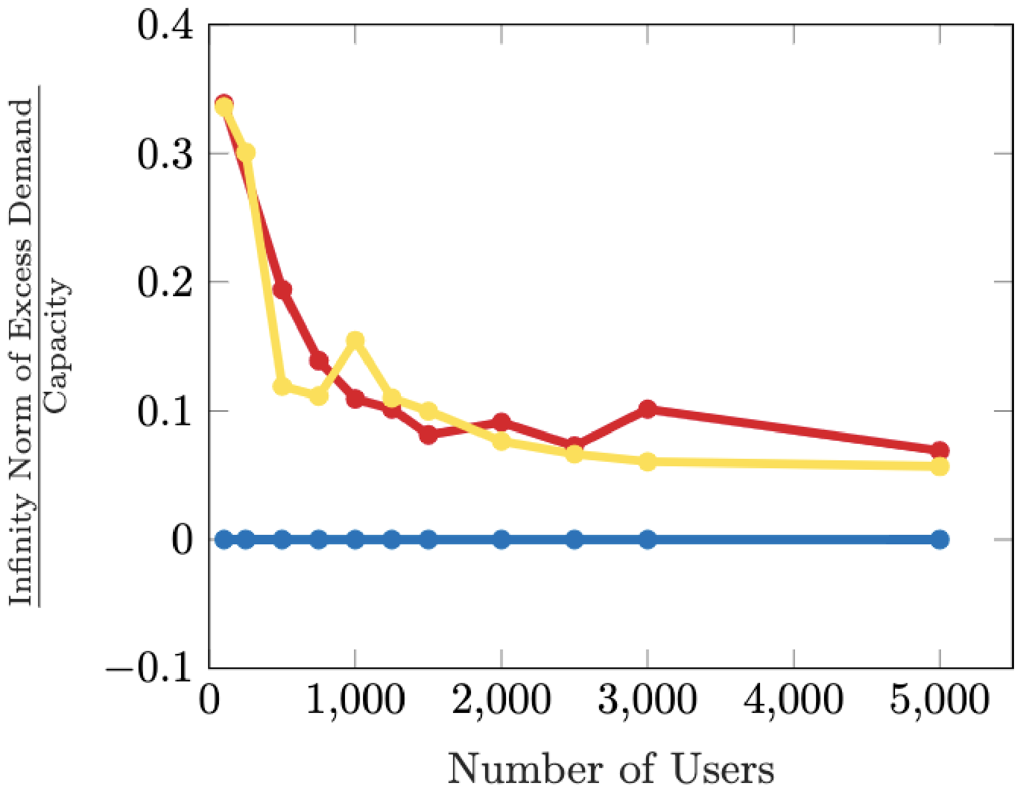
User parameters $(w, \boldsymbol{u})$ are revealed



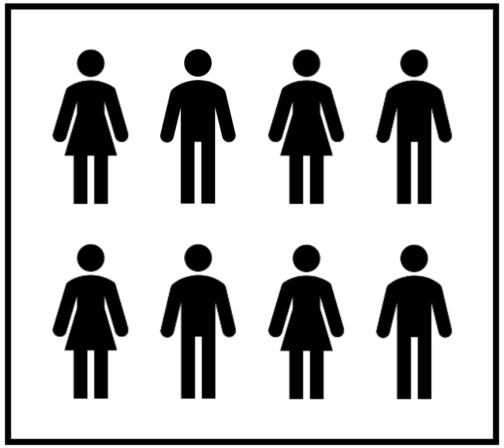
Benchmark 2: Set prices based on a sequence of dual problems using revealed parameters

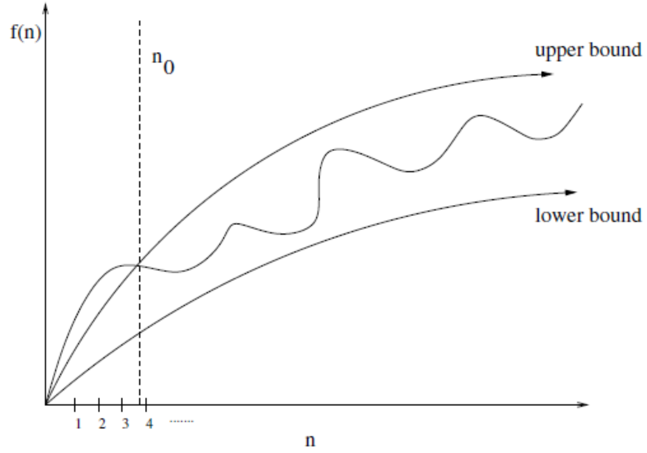
Our numerical results demonstrate a tradeoff between regret and constraint violation

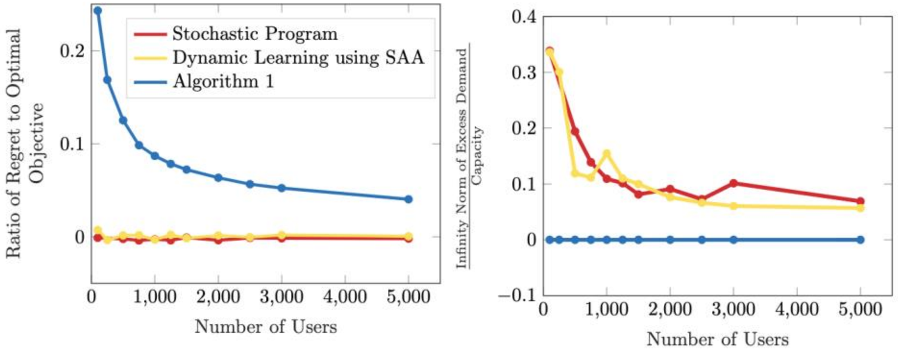




Summary: online algorithms are applicable to Fisher markets with geometric aggregation of social welfare with sub-linear regret guarantees







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)
- Conclusion/Takeaway

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

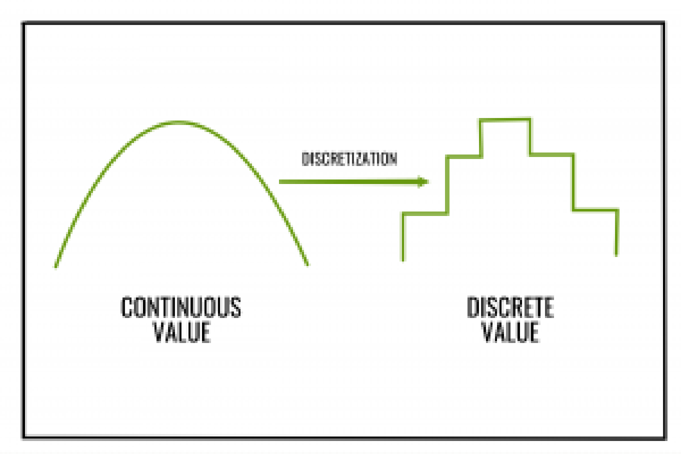
The weighted geometric average objective has several advantages including fairness, computational complexity, and the resulting allocation can be distributed using prices through Fisher markets

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

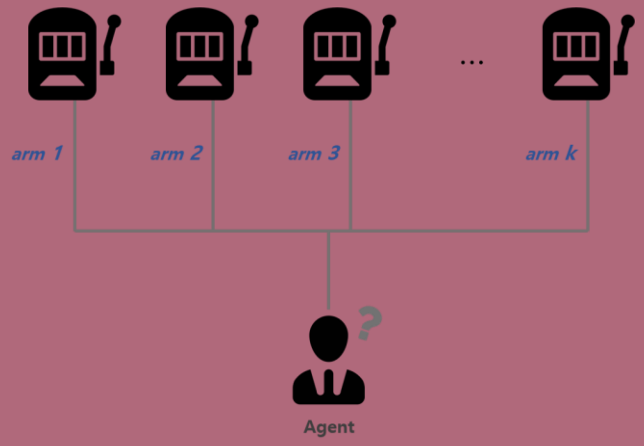
The corresponding allocations can be implemented in the online setting with a sublinear regret

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