Online Learning via Offline Greedy Algorithms: Applications in Market Design and Optimization

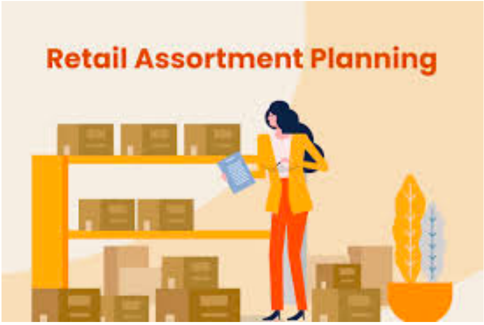
Negin Golrezaei, Sloan School of Management, MIT

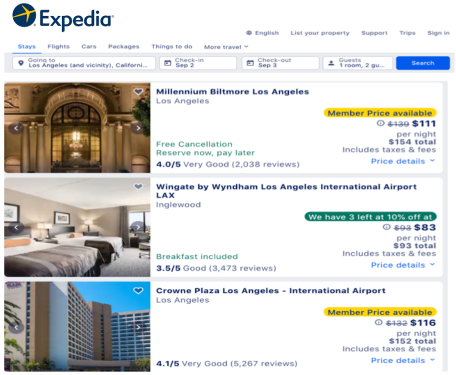
Joint work with R. Niazadeh, J. Wang, F. Susan, and A. Badanidiyuru

Simons Institute, Mathematics of Online Decision Making Workshop
Oct 29th, 2020

Decision-making in Marketplaces

Marketplaces have to make certain decisions repeatedly over time





Assortment planning: What items to offer to customers to maximize market share?

Product ranking: How to display products on online platforms?

Reserve price optimization: How to set reserve prices in auctions run to sell ads?

Challenges: Online decision making under uncertainty in a time-varying environment

Without uncertainty, the offline problem is NP-hard to solve

Research Questions

How to design learning algorithms for such combinatorial and time-varying environments?

Can we transform offline algorithms to online algorithms with sublinear (approximate) regret?

Yes, for a large class of offline problems that admit a robust greedy algorithm with a constant approximation factor

Use this problem to illustrate our technique

Assortment planning	**Product ranking**	**Reserve price optimization**
Greedy ✓	**Greedy ✓**	**Greedy ✓**

Preliminary: Offline Problem

- There are n products
- Our goal is to choose set S with $|S| \leq k$ that maximizes market share (probability of purchase)
  - $f(S) = \sum_{i \in S} \mathrm{Prob}(\text{i is purchased } |S)$ is the market share (demand) under set S
  - $f(\cdot)$ is a monotone submodular function under all random utility choice models
- We want to find

$$\boxed{S^* = \mathrm{argmax}_{|S| \leq k} f(S)}$$ Offline Problem

- The offline problem admits a greedy algorithm with $\gamma = 1 - 1/e$ approx. factor [Nemhauser et al., 1978]

Greedy Algorithm for the Offline Problem

Initialize $S^{(0)} = \{\}$
For subproblem $i \; = \; 1 \; to \; k$:

Greedily pick $z_i \in [n]$ such that

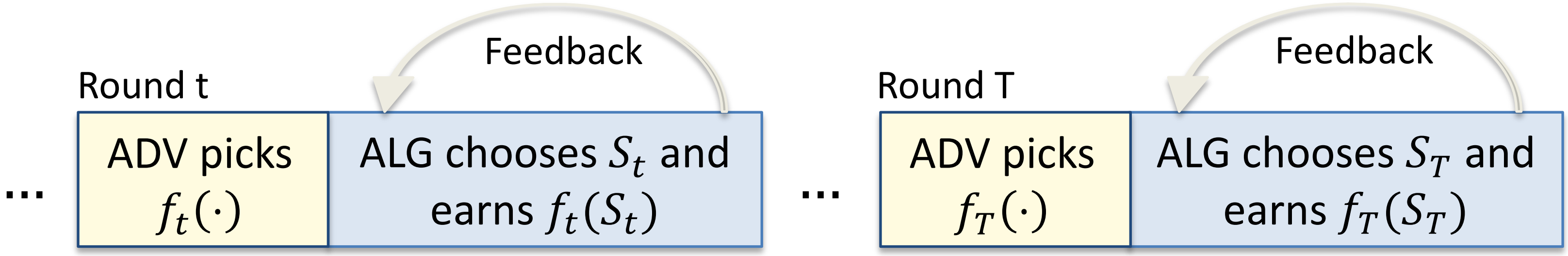
$$z_i \leftarrow argmax_{j \in [n]} \{ f\left(S^{(i-1)} \cup \{j\}\right) - f\left(S^{(i-1)}\right) \}$$

Choose a product with the maximum marginal market share

Set $S^{(i)} \leftarrow S^{(\boldsymbol{i-1})} \cup \{z_{\boldsymbol{i}}\}$
End
Return $S^{(k)}$

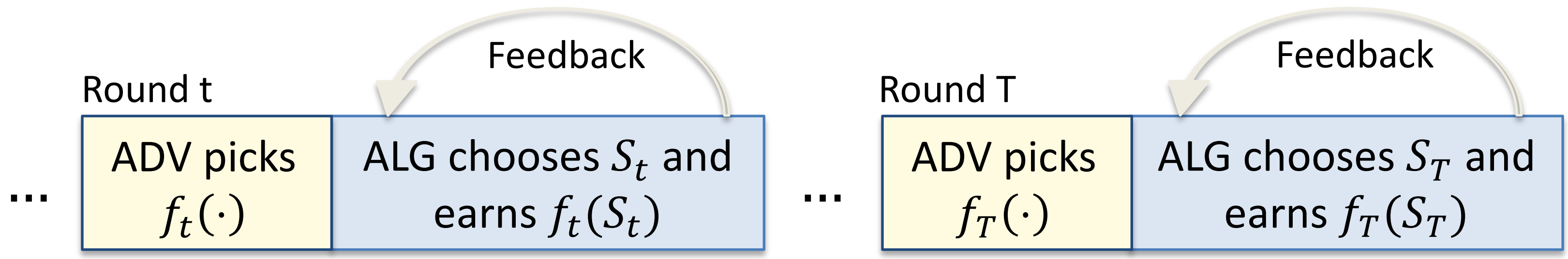
Greedy algorithm builds the solution <u>stage by stage</u>

Preliminary: Online Problem



- T periods
- In round t, nature (ADV) chooses a monotone submodular demand function $f_t(\cdot)$
- $f_t(\cdot)$ is unobservable to the decision-maker (ALG) at the time of the decision
- ALG chooses a set S_t and obtains market share (reward) of $f_t(S_t)$
- ALG gets feedback
  - Full information: ALG observes $f_t(\cdot)$ — Today's talk
  - Bandit: ALG only observes $f_t(S_t)$

Preliminary: Online Problem



Goal: minimize regret w.r.t. $\gamma \cdot \mathrm{OPT}$

$$\mathrm{OPT} = \max_{S,|S|\leq k} \sum_{t\in[T]} f_t(S)$$

Regret= $\gamma \cdot \mathrm{OPT} - \sum_{t\in[T]} f_t(S_t)$

Contributions and Main Results

- Design an efficient framework to transform offline greedy-based algorithm to a low-regret online algorithm via Blackwell approachability
  - For full information and bandit feedback structures

- $O(\sqrt{T})\ \gamma -$ regret for full information and $O(T^{2/3})\ \gamma -$ regret for bandit

- Maximizing monotone set submodular with cardinality constraints
  - **Full information**: our γ-regret bound $O\left(k\sqrt{T\log n}\right)$ *[Best prior bound $O\left(k\sqrt{T\,log\,n}\right)$ by Streeter and Golovin, 2008]*
  - **Bandit**: our bound $O\left(kn^{2/3}(\log n)^{1/3}\,T^{2/3}\right)$ *[Best prior bound $O(k^2(n\,log\,n)^{1/3}T^{2/3}(log\,T)^2)$ by Streeter and Golovin, 2008]*

Contributions and Main Results

- Our framework has a wide-range of applications

		Online Full-Information Setting		**Online Bandit Setting**	
Applications	γ	Our γ-Regret Bound	The Best Prior Bound	Our γ-Regret Bound	The Best Prior Bound
Product Ranking	1/2	$O(n\sqrt{T}\log n)$	-	$O(n^{5/3}T^{2/3}(\log n)^{1/3})$	-
Reserve Price Optimization	1/2	$O(n\sqrt{T}\log T)$	$O(n\sqrt{T}\log T)^{*}$	$O(n^{3/5}T^{4/5}(\log nT)^{1/3})$	-
Non-Monotone Set SM	1/2	$O(n\sqrt{T})$	$O(n\sqrt{T})^{\ddagger}$	$O(nT^{2/3})$	-
Non-Monotone Strong-DR SM	1/2	$O(n\sqrt{T}\log T)$	$\gamma = 1/4$, $O(T^{5/6})^{\S}$	$O(nT^{4/5}(\log T)^{1/3})$	$\gamma = \frac{1}{4}, O(T^{11/12})^{\S}$
Non-Monotone Weak-DR SM	1/2	$O(n\sqrt{T}\log T)$	-	$O(nT^{4/5}(\log T)^{1/3})$	-

$\sqrt{T}$ dependency

Discrete: $T^{\frac{2}{3}}$ dependency; **Continuous**: $T^{\frac{4}{5}}$ dependency

Bandit feedback structure captures more realistic scenarios; But, sparse results!

Related Work

Offline-to-online transformation for NP-hard combinatorial problems

Offline-to-online transformation

- Hazan and Koren, 2016 – **negative results** for general comb. problems
- Kalai and Vempala, 2005, Dudik et al., 2017 – learner can solve **offline problem efficiently**
- Kakade et al., 2009 – NP-hard problem amenable to approximation, **linear rewards**

Combinatorial learning

- Audibert et al., 2014 – exponentially weighted avg. forecaster for full-info setting, tight regret, **linear rewards**
- Bubeck et al., 2012, Hazan and Karnin, 2016 – efficient algorithm for the bandit setting, **linear rewards**

Our contribution

- NP-hard problems with **non-linear rewards**
- Both bandit and full-information settings
- Transform **offline greedy** algorithms to online

High level ideas and our algorithm

Revisiting the Greedy Algorithm

Subproblem 1

⋮

Subproblem i

$z_i \leftarrow argmax_{j \in [n]} \Delta f\left(S^{(i-1)}, j\right)$

⋮

Subproblem k

Greedy chooses product z_i that maximizes marginal market share $\Delta f\left(S^{(i-1)}, \cdot\right)$

$$\text{Payoff}\left(z_i, S^{(i-1)}, \Delta f\right) = \begin{pmatrix} \Delta f\left(S^{(i-1)}, z_i\right) - \Delta f\left(S^{(i-1)}, 1\right) \\ \Delta f\left(S^{(i-1)}, z_i\right) - \Delta f\left(S^{(i-1)}, 2\right) \\ \vdots \\ \Delta f\left(S^{(i-1)}, z_i\right) - \Delta f\left(S^{(i-1)}, n\right) \end{pmatrix} \geq \mathbf{0}$$

Issue: vector payoff is not linear in the greedy's decisions z_i!

$\Delta f(S, j) = f(S \cup \{j\}) - f(S)$ marginal market share of adding product j to set S

Revisiting the Greedy Algorithm

Subproblem 1

⋮

Subproblem i
returns distribution θ_i over n

⋮

Subproblem k

All products with positive mass in θ_i
maximizes marginal market share $\Delta f(S^{(i-1)},\cdot)$

$$\text{Payoff}(\theta_i, S^{(i-1)}, \Delta f) = \begin{pmatrix} \sum_{j\in[n]} \theta_{i,j} \Delta f(S^{(i-1)}, j) - \Delta f(S^{(i-1)}, 1) \\ \sum_{j\in[n]} \theta_{i,j} \Delta f(S^{(i-1)}, j) - \Delta f(S^{(i-1)}, 2) \\ \vdots \\ \sum_{j\in[n]} \theta_{i,j} \Delta f(S^{(i-1)}, j) - \Delta f(S^{(i-1)}, n) \end{pmatrix} \geq \mathbf{0}$$

Vector payoff is now LINEAR in the greedy's decisions θ_i!

$\sum_{j\in[n]} \theta_{i,j} \Delta f(S^{(i-1)}, j)$ is the expected value of marginal market share at the greedy solution θ_i

Greedy Algorithm is Robust to Local Errors

Errorless system: For every subproblem i and coordinate j, if we have

$$[\text{Payoff}(\theta_i, S^{(i-1)}, \Delta f)]_j \geq 0 \quad j \in [n]$$

we get $\gamma = \left(1 - \frac{1}{e}\right)$ approx. factor:

$$f\left(S^{(k)}\right) \geq \gamma \cdot f(S^*)$$

System with local errors: If θ_i is replaced by its noisy version $\widetilde{\theta}_i$ such that

$$[\text{Payoff}(\widetilde{\theta}_i, S^{(i-1)}, \Delta f)]_j + \epsilon \geq 0 \quad j \in [n]$$

we get

$$f\left(S^{(k)}\right) \geq \gamma f(S^*) - \epsilon\, k$$

Local errors do not propagate!

That Is Not All! Greedy is Extended Robust

Consider noisy run of the algorithm over T rounds. Then, if for every subproblem i

$$\left[\sum_{t\in[T]} \text{Payoff}\left(\widetilde{\theta_{i,t}}, S_t^{(i)}, \Delta f_t\right)\right]_j + \text{Error(T)} \geq 0 \qquad j \in [n]$$

we have

$$\sum_{t\in[T]} f_t(S_t) \geq \gamma \cdot \sum_{t\in[T]} f_t(S) - k \cdot \text{Error(T)} \qquad \forall S: |S| \leq k$$

If the aggregate error (over the T rounds) for every coordinate is small, the algorithm will still do well

- We say the greedy algorithm is extended robust
- Not every greedy algorithm has this property

Our High-level Idea

Transforming offline greedy algorithm to an online algorithm

Offline problem

⋮

Subproblem i
Returns distribution θ_i:
Payoff$\left(\theta_i, S^{(i-1)}, \Delta \boldsymbol{f}\right) \geq \mathbf{0}$

⋮

Online problem (round t)
$\boldsymbol{f_t}$ is unknown

⋮

Subproblem i
returns distribution $\widetilde{\theta_{i,t}}$: for $j \in [n]$
$\left[\sum_{\tau=1}^{t} \text{Payoff}\left(\widetilde{\theta_{i,t}}, S_\tau^{(i-1)}, \Delta f_\tau\right)\right]_j$ + Error(t)≥ 0

⋮

If we can keep Error(t) = O$(\sqrt{t})$, by the extended robustness property, we get **$\boldsymbol{O}(\sqrt{\boldsymbol{T}})$ $\boldsymbol{\gamma}$-regret**

Extended robustness :

$$\sum_{t\in[T]} f_t(S_t) \geq \gamma \cdot \sum_{t\in[T]} f_t(S) - k \cdot \text{Error(T)} \qquad \forall S: |S| \leq k$$

This is what ALG earns   This is the benchmark

Question: How to design an algorithm for each subproblem with Error(t) = $O(\sqrt{t})$?

Blackwell Approachability

Blackwell Sequential Games

Repeated two-player (P1 and P2) zero-sum game with vector-valued reward

Round t		...	Round T	
P1 plays x_t P2 plays y_t	P1 obtains $\mathrm{r}(\mathrm{x_t}, \mathrm{y_t})$ P2 obtains $-\mathrm{r}(\mathrm{x_t}, \mathrm{y_t})$	...	P1 plays x_T P2 plays y_T	P1 obtains $\mathrm{r}(\mathrm{x_T}, \mathrm{y_T})$ P2 obtains -$\mathrm{r}(\mathrm{x_T}, \mathrm{y_T})$

$r(\cdot,\cdot)$ is a vector-valued

Reward vector $r(x, y)$ is biaffine

Blackwell Game: P1 wants to approach a convex set S and P2 does not want this to happen

A convex and closed target set S is $g(T)$ −approachable if ∃ a P1 strategy such that for every P2 strategy:

$$d_\infty\left(\frac{1}{T}\sum_{t=1}^{T}\mathrm{r}(\mathrm{x_t}, \mathrm{y_t}), S\right) \le g(T)$$

We want g(T) to go to zero as $T \to \infty$

Average vector-valued reward

Not Every Target Set Is Approachable

Set S is approachable if for every P2 action y, there exists a P1 action x, such that r$(x, y) \in S$

S is approachable → S is $\mathrm{g(T)} = O\left(D(r)(\log d)^{1/2} T^{-1/2}\right)$ −approachable

- $D(r)$ is the diameter of the reward vector
- d is the dimension of the reward vector

For any approachable set, there is an algorithm *AlgB* with
$$\mathrm{g(T)} = O\left(D(r)(\log d)^{1/2} T^{-1/2}\right)$$

Revisiting our High-level Idea

Transforming offline greedy algorithm to an online algorithm

Offline problem

⋮

Subproblem i
Returns distribution θ_i:
Payoff$\left(\theta_i, S^{(i-1)}, \Delta \boldsymbol{f}\right) \geq \mathbf{0}$

⋮

Online problem (round t)
$\boldsymbol{f_t}$ is unknown

⋮

Subproblem i
returns distribution $\widetilde{\theta_{i,t}}$: for $j \in [n]$
$\left[\sum_{\tau=1}^{t} \text{Payoff}\left(\widetilde{\theta_{i,\tau}}, S_\tau^{(i-1)}, \Delta f_\tau\right)\right]_j + \text{Error(t)} \geq 0$

⋮

We let *AlgB* handle each subproblem $i \in [k]$

Blackwell Algorithms Handle Subproblems

- P1 is algorithm that returns $\widetilde{\theta_{i,t}}$
- P2 is the nature (ADV) that chooses

$\Delta f\left(S^{(i-1)},.\right)$

Subproblem i

returns distribution $\widetilde{\theta_{i,t}}$: for $j \in [n]$

$\left[\sum_{\tau=1}^{t} \text{Payoff}\left(\widetilde{\theta_{i,\tau}}, S_{\tau}^{(i-1)}, \Delta f_{\tau}\right)\right]_j$ + Error(t)≥ 0

- Per period payoff vector is biaffine

$$\text{Payoff}(\theta_i, S^{(i-1)}, \Delta f) = \begin{pmatrix} \sum_{j\in[n]} \theta_{i,j} \Delta f\left(S^{(i-1)}, j\right) - \Delta f\left(S^{(i-1)}, 1\right) \\ \sum_{j\in[n]} \theta_{i,j} \Delta f\left(S^{(i-1)}, j\right) - \Delta f\left(S^{(i-1)}, 2\right) \\ \vdots \\ \sum_{j\in[n]} \theta_{i,j} \Delta f\left(S^{(i-1)}, j\right) - \Delta f\left(S^{(i-1)}, n\right) \end{pmatrix} \geq \mathbf{0}$$

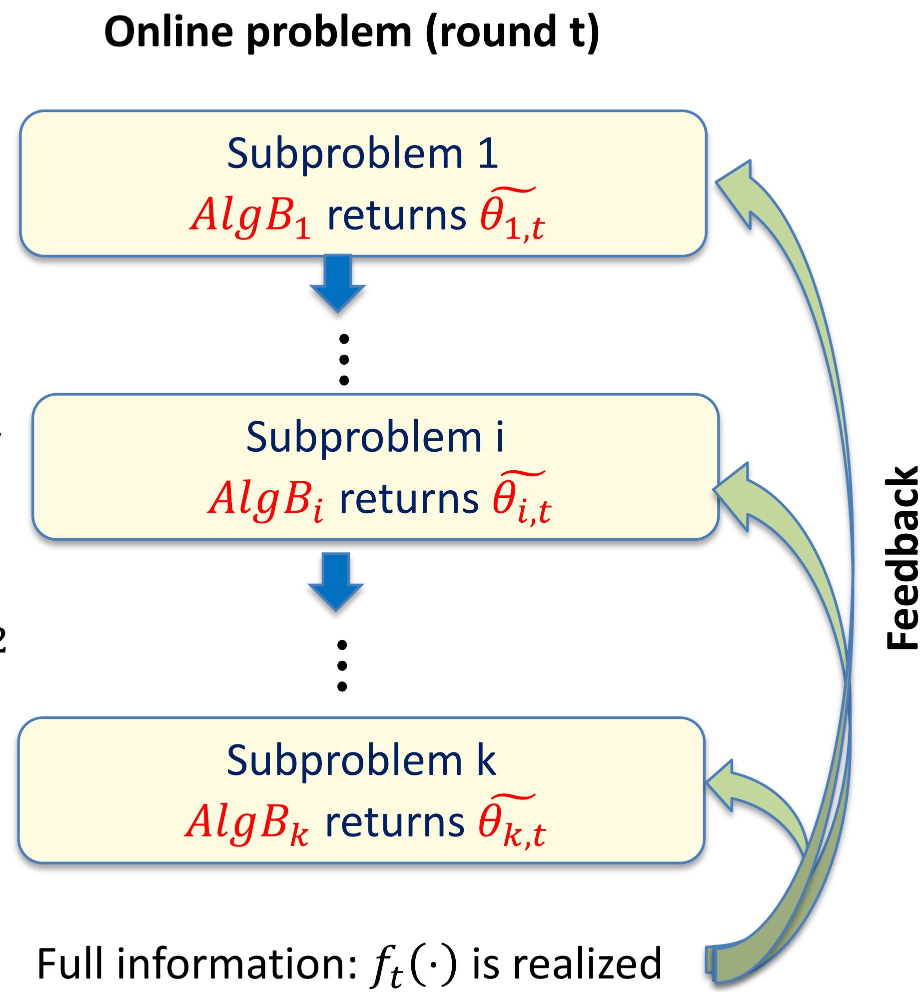
- Target set S is the positive orthant Payoff $\left(\widetilde{\theta_{i,t}}, S_t^{(i-1)}, \Delta f_t\right) \geq \mathbf{0}$ and is approachable
- We can approach set S with g(t) $=\ O\left(D(r)(\log d)^{1/2} t^{-1/2}\right) = \mathrm{O}(\log(n)^{1/2} t^{-1/2})$

Error(t) = $t^{1/2} \log(n)^{1/2}$

Blackwell Algorithms Coordination and Regret

- Blackwell algorithms coordinate with the help of the greedy algorithm
- Each *AlgB* gets updated independently
- Each *AlgB has* Error(T) = $T^{1/2}\log(n)^{1/2}$

Total regret = K $T^{1/2}\log(n)^{1/2}$



Full Information: Beyond Assortment Planning

Theorem 1 (Full-information offline-to-online transformation) Suppose that an offline algorithm

- is an extended robust approximation algorithm, and
- Blackwell reducible**.**

Then, in the full information setting, there exists an online algorithm that runs in polynomial time and satisfies:

$$\gamma - \text{regret} \leq O\left(kD(p)(\log d)^{1/2} T^{1/2}\right)$$

where k is the number of subproblems, d is the dimension of the payoffs, and $D(p)$ is the ℓ_∞ diameter of the vector payoff.

Blackwell reducible:

1) Defining bi-affine vector payoff for each subproblem
2) Defining an approachable target set for each subproblem

Maximizing Non-Monotone Submodular Functions

- Our framework has a wide-range of applications

		Online Full-Information Setting		Online Bandit Setting	
Applications	γ	Our γ-Regret Bound	The Best Prior Bound	Our γ-Regret Bound	The Best Prior Bound
Product Ranking	1/2	$O(n\sqrt{T\log n})$	-	$O(n^{5/3}T^{2/3}(\log n)^{1/3})$	-
Reserve Price Optimization	1/2	$O(n\sqrt{T\log T})$	$O(n\sqrt{T\log T})^{*}$	$O(n^{3/5}T^{4/5}(\log nT)^{1/3})$	-
Non-Monotone Set SM	1/2	$O(n\sqrt{T})$	$O(n\sqrt{T})^{‡}$	$O(nT^{2/3})$	-
Non-Monotone Strong-DR SM	1/2	$O(n\sqrt{T\log T})$	$\gamma = 1/4$, $O(T^{5/6})^{§}$	$O(nT^{4/5}(\log T)^{1/3})$	$\gamma = \frac{1}{4}$, $O(T^{11/12})^{§}$
Non-Monotone Weak-DR SM	1/2	$O(n\sqrt{T\log T})$	-	$O(nT^{4/5}(\log T)^{1/3})$	-

[*] Roughgarden and Wang, 2019; [‡] Roughgarden and Wang, 2018; [§] Thang and Srivastav, 2019

Takeaway

- Transform offline greedy algorithms to online ones using Blackwell approachability
  - Need the greedy algorithm to be extended robust and bandit Blackwell reducible
- For full information setting, our algorithm has $O\left(\sqrt{T}\right)$ γ −regret
- For Bandit setting, our algorithm has $O\left(T^{2/3}\right)$ γ −regret
- Our framework is flexible and can be applied to many applications
  - Product ranking optimization in online platforms
  - Reserve price optimization in auctions
  - Submodular maximization

Link to the paper: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3613756

Email: golrezae@mit.edu