Online Linear Programming: Applications and Extensions

Yinyu Ye

Department of Management Science and Engineering
Institute of Computational and Mathematical Engineering
Stanford University, Stanford

ADSE Seminar Series
June 20, 2022
(Joint work with many...)

Table of Contents

1. Online Linear Programming
2. Fast Algorithms for (Binary) Online Linear Programming
3. A Fairer Online Interior-Point LP Algorithm
4. Online Bandits with Knapsacks
5. Online Fisher Markets

Offline and Online Linear Programming

$$\begin{array}{ll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} r_t x_t \\ \text{subject to} & \sum_{t=1}^{n} \mathbf{a}_t x_t \leq \mathbf{b}, \\ & x_t \in \{0,1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n. \end{array}$$

Offline and Online Linear Programming

$$
\begin{array}{ll}
\text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} r_t x_t \\
\text{subject to} & \sum_{t=1}^{n} \mathbf{a}_t x_t \leq \mathbf{b}, \\
 & x_t \in \{0,1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n.
\end{array}
$$

r_t: reward/revenue offered by the t-th customer/order

$\mathbf{a}_t \in R^m$: the bundle of resources requested by the t-th order

x_t: acceptance or rejection decision to the t-th order

$\mathbf{b} \in R^m$: initially available budget/resource amounts

The objective $\sum_{t=1}^{n} r_t x_t$: the total collected revenue.

Offline and Online Linear Programming

$$
\begin{array}{ll}
\text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} r_t x_t \\
\text{subject to} & \sum_{t=1}^{n} \mathbf{a}_t x_t \leq \mathbf{b}, \\
 & x_t \in \{0,1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n.
\end{array}
$$

r_t: reward/revenue offered by the t-th customer/order

$\mathbf{a}_t \in R^m$: the bundle of resources requested by the t-th order

x_t: acceptance or rejection decision to the t-th order

$\mathbf{b} \in R^m$: initially available budget/resource amounts

The objective $\sum_{t=1}^{n} r_t x_t$: the total collected revenue.

- We know only $\mathbf{b}$ and n at the start.

Offline and Online Linear Programming

$$\begin{array}{ll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} r_t x_t \\ \text{subject to} & \sum_{t=1}^{n} \mathbf{a}_t x_t \leq \mathbf{b}, \\ & x_t \in \{0,1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n. \end{array}$$

r_t: reward/revenue offered by the t-th customer/order

$\mathbf{a}_t \in R^m$: the bundle of resources requested by the t-th order

x_t: acceptance or rejection decision to the t-th order

$\mathbf{b} \in R^m$: initially available budget/resource amounts

The objective $\sum_{t=1}^{n} r_t x_t$: the total collected revenue.

- We know only $\mathbf{b}$ and n at the start.
- the bidder data $(r_t, \mathbf{a}_t)$ arrive sequentially.

Offline and Online Linear Programming

$$\begin{array}{ll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} r_t x_t \\ \text{subject to} & \sum_{t=1}^{n} \mathbf{a}_t x_t \leq \mathbf{b}, \\ & x_t \in \{0,1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n. \end{array}$$

r_t: reward/revenue offered by the t-th customer/order

$\mathbf{a}_t \in R^m$: the bundle of resources requested by the t-th order

x_t: acceptance or rejection decision to the t-th order

$\mathbf{b} \in R^m$: initially available budget/resource amounts

The objective $\sum_{t=1}^{n} r_t x_t$: the total collected revenue.

- We know only $\mathbf{b}$ and n at the start.
- the bidder data $(r_t, \mathbf{a}_t)$ arrive sequentially.
- an irrevocable decision must be made as soon as an order arrives (without knowing the future data).

Offline and Online Linear Programming

$$\begin{array}{ll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} r_t x_t \\ \text{subject to} & \sum_{t=1}^{n} \mathbf{a}_t x_t \leq \mathbf{b}, \\ & x_t \in \{0,1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n. \end{array}$$

r_t: reward/revenue offered by the t-th customer/order

$\mathbf{a}_t \in R^m$: the bundle of resources requested by the t-th order

x_t: acceptance or rejection decision to the t-th order

$\mathbf{b} \in R^m$: initially available budget/resource amounts

The objective $\sum_{t=1}^{n} r_t x_t$: the total collected revenue.

- We know only $\mathbf{b}$ and n at the start.
- the bidder data $(r_t, \mathbf{a}_t)$ arrive sequentially.
- an irrevocable decision must be made as soon as an order arrives (without knowing the future data).
- Conform to resource capacity constraints at the end.

A Toy Example

Consider an auction problem:

	Bid 1($t = 1$)	Bid 2($t = 2$)	.....	Inventory($\mathbf{b}$)
Reward(r_t)	\$100	\$30	...	
Decision	x_1	x_2	...	
Pants	1	0	...	100
Shoes	1	0	...	50
T-shirts	0	1	...	500
Jackets	0	0	...	200
Hats	1	1	...	1000

where the decision for each customer/bidder is "accept" ($x_t = 1$) or "reject" ($x_t = 0$)

Price Mechanism for OLP I

The problem would be easy if there are "ideal prices":

	Bid 1($t = 1$)	Bid 2($t = 2$)	.....	Inventory($\mathbf{b}$)	$\mathbf{p}^*$
Bid(r_t)	\$100	\$30	...		
Decision	x_1	x_2	...		
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

so that the online decision can be made by comparing the reward and "bundle cost" for each bid.

Primal and Dual Offline LPs

$$P: \begin{array}{ll} \max & \mathbf{r}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{e} \end{array} \qquad D: \begin{array}{ll} \min & \mathbf{b}^\top \mathbf{p} + \mathbf{e}^\top \mathbf{s} \\ \text{s.t.} & A^\top \mathbf{p} + \mathbf{s} \geq \mathbf{r} \\ & \mathbf{p} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0} \end{array}$$

where the decision variables are $\mathbf{x} \in R^n$, $\mathbf{p} \in R^m$, $\mathbf{s} \in R^n$ ($\mathbf{e}$ vector of all ones).

Primal and Dual Offline LPs

$$
P: \begin{array}{ll} \max & \mathbf{r}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \le \mathbf{b} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{e} \end{array}
\qquad
D: \begin{array}{ll} \min & \mathbf{b}^\top \mathbf{p} + \mathbf{e}^\top \mathbf{s} \\ \text{s.t.} & A^\top \mathbf{p} + \mathbf{s} \ge \mathbf{r} \\ & \mathbf{p} \ge \mathbf{0}, \mathbf{s} \ge \mathbf{0} \end{array}
$$

where the decision variables are $\mathbf{x} \in R^n$, $\mathbf{p} \in R^m$, $\mathbf{s} \in R^n$ ($\mathbf{e}$ vector of all ones).

Denote the primal/dual optimal solution as $\mathbf{x}^*$, $\mathbf{p}^*$, $\mathbf{s}^*$, then LP duality/complementarity theory tells that for $t = 1, ..., n$,

$$
x_t^* = \begin{cases} 1, & r_t > \mathbf{a}_t^\top \mathbf{p}^* \\ 0, & r_t < \mathbf{a}_t^\top \mathbf{p}^* \end{cases}
$$

(x_t^* may take non-integer value when $r_t = \mathbf{a}_t^\top \mathbf{p}^*$).

Primal and Dual Offline LPs

$$P: \begin{array}{ll} \max & \mathbf{r}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \le \mathbf{b} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{e} \end{array} \qquad D: \begin{array}{ll} \min & \mathbf{b}^\top \mathbf{p} + \mathbf{e}^\top \mathbf{s} \\ \text{s.t.} & A^\top \mathbf{p} + \mathbf{s} \ge \mathbf{r} \\ & \mathbf{p} \ge \mathbf{0}, \mathbf{s} \ge \mathbf{0} \end{array}$$

where the decision variables are $\mathbf{x} \in R^n$, $\mathbf{p} \in R^m$, $\mathbf{s} \in R^n$ ($\mathbf{e}$ vector of all ones).

Denote the primal/dual optimal solution as $\mathbf{x}^*$, $\mathbf{p}^*$, $\mathbf{s}^*$, then LP duality/complementarity theory tells that for $t = 1, ..., n$,

$$x_t^* = \begin{cases} 1, & r_t > \mathbf{a}_t^\top \mathbf{p}^* \\ 0, & r_t < \mathbf{a}_t^\top \mathbf{p}^* \end{cases}$$

(x_t^* may take non-integer value when $r_t = \mathbf{a}_t^\top \mathbf{p}^*$).

Most online LP algorithms are based on learning $\mathbf{p}^*$ by dynamically solving small sample-sized LPs based on revealed data.

Simple Price-Learning Algorithm

We illustrate a simple Learning Algorithm:

- Set $x_t = 0$ for all $1 \leq t \leq \epsilon n$;

Simple Price-Learning Algorithm

We illustrate a simple Learning Algorithm:

- Set $x_t = 0$ for all $1 \le t \le \epsilon n$;
- Solve the ϵ portion of the problem

$$
\begin{array}{lll}
\text{maximize}_{\mathbf{x}} & \sum_{t=1}^{\epsilon n} r_t x_t & \\
\text{subject to} & \sum_{t=1}^{\epsilon n} a_{it} x_t \le \frac{\epsilon n}{n} b_i & i = 1, ..., m \\
 & 0 \le x_t \le 1 & t = 1, ..., \epsilon n
\end{array}
$$

and get the optimal dual solution $\hat{\mathbf{p}}$;

Simple Price-Learning Algorithm

We illustrate a simple Learning Algorithm:

- Set $x_t = 0$ for all $1 \leq t \leq \epsilon n$;
- Solve the ϵ portion of the problem

$$
\begin{array}{lll}
\text{maximize}_{\mathbf{x}} & \sum_{t=1}^{\epsilon n} r_t x_t & \\
\text{subject to} & \sum_{t=1}^{\epsilon n} a_{it} x_t \leq \frac{\epsilon n}{n} b_i & i = 1, ..., m \\
& 0 \leq x_t \leq 1 & t = 1, ..., \epsilon n
\end{array}
$$

  and get the optimal dual solution $\hat{\mathbf{p}}$;
- Determine the future allocation x_t as:

$$
x_t = \begin{cases} 0 & \text{if } r_t \leq \hat{\mathbf{p}}^T \mathbf{a}_t \\ 1 & \text{if } r_t > \hat{\mathbf{p}}^T \mathbf{a}_t \end{cases}
$$

One may update the prices periodically and/or set $x_t = 0$ as soon as a resource is exhausted.

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

(a) $(r_t, \mathbf{a}_t)$'s are i.i.d. from an unknown distribution

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

(a) $(r_t, \mathbf{a}_t)$'s are i.i.d. from an unknown distribution

Random Permutation (RP) Model:

(a') $(r_t, \mathbf{a}_t)$'s may be adversarially chosen but arrive in a random order (sample without replacement)

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

(a) $(r_t, \mathbf{a}_t)$'s are i.i.d. from an unknown distribution

Random Permutation (RP) Model:

(a') $(r_t, \mathbf{a}_t)$'s may be adversarially chosen but arrive in a random order (sample without replacement)

Both assume boundedness:

(b) $|r_t| \leq \bar{r}$ and $\|\mathbf{a}_t\|_\infty \leq \bar{a}$ for all t

(c) The right-hand-side $\mathbf{b} = n \cdot \mathbf{d}(> \mathbf{0})$.

All early works also assume $r_t \geq 0, \mathbf{a}_t \geq 0$ (one-sited market).

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

(a) $(r_t, \mathbf{a}_t)$'s are i.i.d. from an unknown distribution

Random Permutation (RP) Model:

(a') $(r_t, \mathbf{a}_t)$'s may be adversarially chosen but arrive in a random order (sample without replacement)

Both assume boundedness:

(b) $|r_t| \leq \bar{r}$ and $\|\mathbf{a}_t\|_\infty \leq \bar{a}$ for all t

(c) The right-hand-side $\mathbf{b} = n \cdot \mathbf{d}(> \mathbf{0})$.

All early works also assume $r_t \geq 0, \mathbf{a}_t \geq 0$ (one-sited market).

- What are the necessary and sufficient assumptions on the right-hand-side $\mathbf{b}$ to achieve $(1-\epsilon)$-competitive ratio of the expected online reward over the optimal offline reword?

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

(a) $(r_t, \mathbf{a}_t)$'s are i.i.d. from an unknown distribution

Random Permutation (RP) Model:

(a') $(r_t, \mathbf{a}_t)$'s may be adversarially chosen but arrive in a random order (sample without replacement)

Both assume boundedness:

(b) $|r_t| \leq \bar{r}$ and $\|\mathbf{a}_t\|_\infty \leq \bar{a}$ for all t

(c) The right-hand-side $\mathbf{b} = n \cdot \mathbf{d}(> \mathbf{0})$.

All early works also assume $r_t \geq 0, \mathbf{a}_t \geq 0$ (one-sited market).

- What are the necessary and sufficient assumptions on the right-hand-side $\mathbf{b}$ to achieve $(1 - \epsilon)$-competitive ratio of the expected online reward over the optimal offline reword?
- If the right-hand-side $\mathbf{b}$ (such as $\mathbf{b} = O(n)$), what is the best achievable gap or regret between the two?

Competitive Ratio Summary of One-Sited Market

The journey to design $(1-\epsilon)$-competitive online algorithms against benchmark OPT-Optimal Offline Objective Value where $B = \min_i b_i$:

	Sufficient Condition
Kleinberg (2005)	$B \geq \frac{1}{\epsilon^2}$, for $m = 1$
Devanur et al (2009)	$OPT \geq \frac{m^2 \log n}{\epsilon^3}$
Feldman et al (2010)	$B \geq \frac{m \log n}{\epsilon^3}$ and $OPT \geq \frac{m \log n}{\epsilon}$
Agrawal/Wang/Y (2010,14)	$B \geq \frac{m \log n}{\epsilon^2}$ or $OPT \geq \frac{m^2 \log n}{\epsilon^2}$
Molinaro/Ravi (2013)	$B \geq \frac{m^2 \log m}{\epsilon^2}$
Kesselheim et al (2014)	$B \geq \frac{\log m}{\epsilon^2}$
Gupta/Molinaro (2014)	$B \geq \frac{\log m}{\epsilon^2}$
Agrawal/Devanur (2014)	$B \geq \frac{\log m}{\epsilon^2}$

Competitive Ratio Summary of One-Sited Market

The journey to design $(1-\epsilon)$-competitive online algorithms against benchmark OPT-Optimal Offline Objective Value where $B = \min_i b_i$:

	Sufficient Condition
Kleinberg (2005)	$B \geq \frac{1}{\epsilon^2}$, for $m = 1$
Devanur et al (2009)	$OPT \geq \frac{m^2 \log n}{\epsilon^3}$
Feldman et al (2010)	$B \geq \frac{m \log n}{\epsilon^3}$ and $OPT \geq \frac{m \log n}{\epsilon}$
Agrawal/Wang/Y (2010,14)	$B \geq \frac{m \log n}{\epsilon^2}$ or $OPT \geq \frac{m^2 \log n}{\epsilon^2}$
Molinaro/Ravi (2013)	$B \geq \frac{m^2 \log m}{\epsilon^2}$
Kesselheim et al (2014)	$B \geq \frac{\log m}{\epsilon^2}$
Gupta/Molinaro (2014)	$B \geq \frac{\log m}{\epsilon^2}$
Agrawal/Devanur (2014)	$B \geq \frac{\log m}{\epsilon^2}$
	Necessary Condition
Agrawal/Wang/Y (2010,14)	$B \geq \frac{\log m}{\epsilon^2}$

Remarks

- The optimal online algorithms have been designed for the competitive ratio analyses and for one-sited market and random permutation data model!

Remarks

- The optimal online algorithms have been designed for the competitive ratio analyses and for one-sited market and random permutation data model!
- The key difference between OLP and Online Convex Optimization with Constraints (OCOwC):
  - Online LP problem employs a stronger benchmark where the decision variables are allowed to take different values at each time period
  - OCOwC (Mahdavi et al., 2012; Yu et al., 2017; Yuan and Lamperski, 2018) and OCO problems usually consider a stationary benchmark where the the decision variables are required to be the same at each time period.

Remarks

- The optimal online algorithms have been designed for the competitive ratio analyses and for one-sited market and random permutation data model!
- The key difference between OLP and Online Convex Optimization with Constraints (OCOwC):
  - Online LP problem employs a stronger benchmark where the decision variables are allowed to take different values at each time period
  - OCOwC (Mahdavi et al., 2012; Yu et al., 2017; Yuan and Lamperski, 2018) and OCO problems usually consider a stationary benchmark where the the decision variables are required to be the same at each time period.
- Recent focuses are on dealing with two-sited markets/platforms, regret analyses, simple and fast algorithms, interior-point online algorithm, extension to bandit models and the Fisher market.

Table of Contents

1 Online Linear Programming

2 Fast Algorithms for (Binary) Online Linear Programming

3 A Fairer Online Interior-Point LP Algorithm

4 Online Bandits with Knapsacks

5 Online Fisher Markets

Regret Analysis and Model

Let "offline" optimal solution be $\mathbf{x}^*$ and "online" solution of n orders be $\mathbf{x}_n$, and

$$R_n^* = \sum_{j=1}^{n} r_j x_j^*, \quad R_n = \sum_{j=1}^{n} r_j x_j.$$

Regret Analysis and Model

Let "offline" optimal solution be $\mathbf{x}^*$ and "online" solution of n orders be $\mathbf{x}_n$, and

$$R_n^* = \sum_{j=1}^{n} r_j x_j^*, \quad R_n = \sum_{j=1}^{n} r_j x_j.$$

Then define

$$\Delta_n = \sup \mathbb{E}\left[R_n^* - R_n\right], \quad v(\mathbf{x}) = \sup \mathbb{E}\left[\|\left(A\mathbf{x} - \mathbf{b}\right)^+\|_2\right]$$

where the expectation is taken with respect to i.i.d distribution or random permutation, and the sup operator is over all permissible distributions and admissible data.

Regret Analysis and Model

Let "offline" optimal solution be $\mathbf{x}^*$ and "online" solution of n orders be $\mathbf{x}_n$, and

$$R_n^* = \sum_{j=1}^{n} r_j x_j^*, \quad R_n = \sum_{j=1}^{n} r_j x_j.$$

Then define

$$\Delta_n = \sup \mathbb{E}\left[R_n^* - R_n\right], \quad v(\mathbf{x}) = \sup \mathbb{E}\left[\|\left(A\mathbf{x} - \mathbf{b}\right)^+\|_2\right]$$

where the expectation is taken with respect to i.i.d distribution or random permutation, and the sup operator is over all permissible distributions and admissible data.

Remark: A bi-criteria performance measure, but one can easily modify the algorithms such that the constraints are always satisfied at the end.

Equivalent Form of the Dual Problem

Recall the dual problem

$$\min \ \mathbf{b}^\top \mathbf{p} + \sum_{t=1}^{n} s_t \quad \text{s.t. } s_t \geq r_t - \mathbf{a}_t^\top \mathbf{p}, \forall t; \quad \mathbf{p}, \mathbf{s} \geq \mathbf{0}$$

can be rewritten as

$$\min \ \mathbf{b}^\top \mathbf{p} + \sum_{t=1}^{n} \left(r_t - \mathbf{a}_t^\top \mathbf{p} \right)^+ \quad \text{s.t. } \ \mathbf{p} \geq \mathbf{0}$$

where $(\cdot)^+$ is the positive-part or ReLU function.

Equivalent Form of the Dual Problem

Recall the dual problem

$$\min\ \mathbf{b}^\top \mathbf{p} + \sum_{t=1}^{n} s_t \quad \text{s.t. } s_t \geq r_t - \mathbf{a}_t^\top \mathbf{p}, \forall t; \quad \mathbf{p}, \mathbf{s} \geq \mathbf{0}$$

can be rewritten as

$$\min\ \mathbf{b}^\top \mathbf{p} + \sum_{t=1}^{n} \left(r_t - \mathbf{a}_t^\top \mathbf{p} \right)^+ \quad \text{s.t. } \ \mathbf{p} \geq \mathbf{0}$$

where $(\cdot)^+$ is the positive-part or ReLU function.

After normalizing the objective, it becomes

$$\min_{\mathbf{p} \geq \mathbf{0}} \mathbf{d}^\top \mathbf{p} + \frac{1}{n} \sum_{t=1}^{n} \left(r_t - \mathbf{a}_t^\top \mathbf{p} \right)^+$$

which can be viewed as a simple-sample-average (SSA) (with n sample points) of a stochastic optimization problem under an i.i.d distribution.

Convergence of $\mathbf{p}_n^*$

Theorem (Li & Y (2019, OR 2021))

Denote the n-sample SSA optimal solution by $\mathbf{p}_n^*$. Then, for the stochastic input model under moderate conditions that guarantee a local strong convexity of the underlying stochastic program $f(p)$ around its optimal solution $\mathbf{p}^*$, there exists a constant C such that

$$\mathbb{E}\|\mathbf{p}_n^* - \mathbf{p}^*\|_2^2 \leq \frac{Cm \log \log n}{n}$$

holds for all $n > m$.

Convergence of $\mathbf{p}_n^*$

Theorem (Li & Y (2019, OR 2021))

Denote the n-sample SSA optimal solution by $\mathbf{p}_n^*$. Then, for the stochastic input model under moderate conditions that guarantee a local strong convexity of the underlying stochastic program $f(p)$ around its optimal solution $\mathbf{p}^*$, there exists a constant C such that

$$\mathbb{E}\|\mathbf{p}_n^* - \mathbf{p}^*\|_2^2 \leq \frac{Cm \log \log n}{n}$$

holds for all $n > m$.

This is L_2 convergence for the dual optimal solution. Heuristically,

$$\mathbf{p}_n^* \approx \mathbf{p}^* + \frac{1}{\sqrt{n}} \cdot \textbf{Noise}$$

Dual-Gradient Online Algorithm for Binary LP

1: Input: $\mathbf{d} = \mathbf{b}/n$

2: Initialize $\mathbf{p}_1 = \mathbf{0}$

3: For $t = 1, 2, ..., n$

4:

$$x_t = \begin{cases} 1, & \text{if } r_t > \mathbf{a}_t^\top \mathbf{p}_t \\ 0, & \text{if } r_t \leq \mathbf{a}_t^\top \mathbf{p}_t \end{cases}$$

5: Compute

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \gamma_t \left(\mathbf{a}_t x_t - \mathbf{d} \right)$$
$$\mathbf{p}_{t+1} = \mathbf{p}_{t+1} \vee \mathbf{0}$$

6: $\mathbf{x} = (x_1, ..., x_n)$

Dual-Gradient Online Algorithm for Binary LP

1: Input: $\mathbf{d} = \mathbf{b}/n$

2: Initialize $\mathbf{p}_1 = \mathbf{0}$

3: For $t = 1, 2, ..., n$

4:
$$x_t = \begin{cases} 1, & \text{if } r_t > \mathbf{a}_t^\top \mathbf{p}_t \\ 0, & \text{if } r_t \le \mathbf{a}_t^\top \mathbf{p}_t \end{cases}$$

5: Compute
$$\mathbf{p}_{t+1} = \mathbf{p}_t + \gamma_t \left(\mathbf{a}_t x_t - \mathbf{d} \right)$$
$$\mathbf{p}_{t+1} = \mathbf{p}_{t+1} \vee \mathbf{0}$$

6: $\mathbf{x} = (x_1, ..., x_n)$

Line 5 performs (projected) stochastic gradient descent in the dual, where step-size $\gamma_t = \frac{1}{\sqrt{n}}$ or $\gamma_t = \frac{1}{\sqrt{t}}$.

Performance Analysis

Theorem (Li, Sun & Y (2020, NeurIPS))

With step size $\gamma_t = 1/\sqrt{n}$, the regret and expected constraint violation of the algorithm satisfy

$$\mathbb{E}[R_n^* - R_n] \leq \tilde{O}(m\sqrt{n}), \quad \mathbb{E}\left[v(\mathbf{x})\right] \leq \tilde{O}(m\sqrt{n}).$$

under both the stochastic input and the random permutation models.

- $\tilde{O}$ omits the logarithm terms and the constants related to $(\bar{a}, \bar{r})$, but the algorithm does not require any prior knowledge on the constants.
- The optimal offline value is in the range $O(mn)$.
- The algorithms runs in nm times - the time to read in the data.
- It can be implemented by posting prices: customers decide and keep r_t's private.

Adaptive Fast Online Algorithm for Binary LP

Initialize $\mathbf{b}_1 = \mathbf{b}$ and $\mathbf{p}_1 = \mathbf{0}$

For $t = 1, 2, ..., n$

$$x_t = \begin{cases} 1, & \text{if } r_t > \mathbf{a}_t^\top \mathbf{p}_t \\ 0, & \text{if } r_t \le \mathbf{a}_t^\top \mathbf{p}_t \end{cases}$$

Compute

$$\begin{aligned} \mathbf{p}_{t+1} &= \mathbf{p}_t + \gamma_t \left(\mathbf{a}_t x_t - \frac{1}{n-t+1} \mathbf{b}_t \right) \\ \mathbf{p}_{t+1} &= \mathbf{p}_{t+1} \vee \mathbf{0} \end{aligned}$$

Update remaining inventory: $\mathbf{b}_{t+1} = \mathbf{b}_t - \mathbf{a}_t x_t$.

Return $\mathbf{x} = (x_1, ..., x_n)$

Adaptive Fast Online Algorithm for Binary LP

1: Initialize $\mathbf{b}_1 = \mathbf{b}$ and $\mathbf{p}_1 = \mathbf{0}$

2: For $t = 1, 2, ..., n$

3:

$$x_t = \begin{cases} 1, & \text{if } r_t > \mathbf{a}_t^\top \mathbf{p}_t \\ 0, & \text{if } r_t \leq \mathbf{a}_t^\top \mathbf{p}_t \end{cases}$$

4: Compute

$$\begin{aligned} \mathbf{p}_{t+1} &= \mathbf{p}_t + \gamma_t \left(\mathbf{a}_t x_t - \tfrac{1}{n-t+1} \mathbf{b}_t \right) \\ \mathbf{p}_{t+1} &= \mathbf{p}_{t+1} \vee \mathbf{0} \end{aligned}$$

5: Update remaining inventory: $\mathbf{b}_{t+1} = \mathbf{b}_t - \mathbf{a}_t x_t$.

6: Return $\mathbf{x} = (x_1, ..., x_n)$

The average inventory vector is adaptively adjusted based on the previous realizations/decisions – this is a non-stationary approach.

Nonadaptive vs. Adaptive

The first resource (sequential) usages in 10 runs of the algorithms.

Nonadaptive vs. Adaptive

The first resource (sequential) usages in 10 runs of the algorithms.

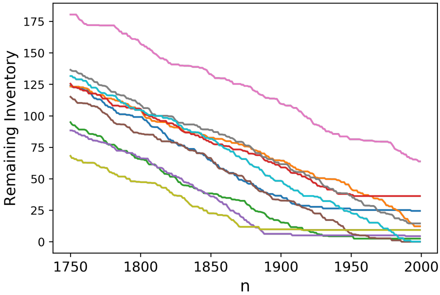


Figure: Nonadaptive

Nonadaptive vs. Adaptive

The first resource (sequential) usages in 10 runs of the algorithms.

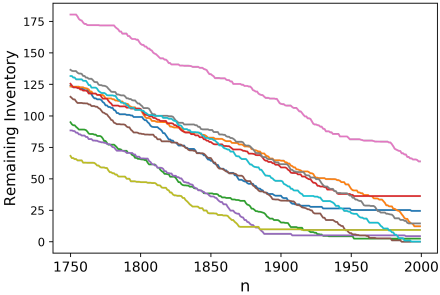


Figure: Nonadaptive

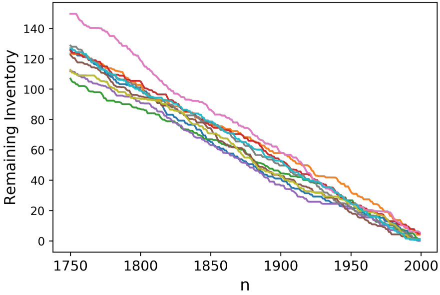


Figure: Adaptive

Fast Online LP Algorithm for Solving Offline LPs?

A crucial assumption is that the right-hand-side $\mathbf{b} = n\mathbf{d}$ scales linearly with n. Is there a remedy for this case where we do not want to compromise the computational efficiency of simple online algorithm?

Fast Online LP Algorithm for Solving Offline LPs?

A crucial assumption is that the right-hand-side $\mathbf{b} = n\mathbf{d}$ scales linearly with n. Is there a remedy for this case where we do not want to compromise the computational efficiency of simple online algorithm?

Consider a "Replicated" LP from the original LP

$$\begin{aligned} \max \ & \sum_{t=1}^{n}\sum_{h=1}^{k} r_t x_{th} \\ \text{s.t.} \ & \sum_{t=1}^{n}\sum_{h=1}^{k} \mathbf{a}_t x_{th} \le k\mathbf{b}, \ 0 \le x_t \le 1, \quad t = 1, ..., n. \end{aligned}$$

Algorithm: Solve the new LP with Simple Online Algorithm and use $x_t = \frac{1}{k}(x_{t1} + ... + x_{tk})$ as the solution to the original LP.

Fast Online LP Algorithm for Solving Offline LPs?

A crucial assumption is that the right-hand-side $\mathbf{b} = n\mathbf{d}$ scales linearly with n. Is there a remedy for this case where we do not want to compromise the computational efficiency of simple online algorithm?

Consider a "Replicated" LP from the original LP

$$\begin{aligned} \max \quad & \sum_{t=1}^{n} \sum_{h=1}^{k} r_t x_{th} \\ \text{s.t.} \quad & \sum_{t=1}^{n} \sum_{h=1}^{k} \mathbf{a}_t x_{th} \leq k\mathbf{b}, \; 0 \leq x_t \leq 1, \quad t = 1, ..., n. \end{aligned}$$

Algorithm: Solve the new LP with Simple Online Algorithm and use $x_t = \frac{1}{k}(x_{t1} + ... + x_{tk})$ as the solution to the original LP.
The algorithm runs in $O(kmn)$ times.

Performance of the Variable-Replicating Algorithm

Proposition (Gao, Sun, Ye & Y (2021))

Under the random permutation model, the variable-replicating algorithm finds a solution for the original LP that achieves at least $(1-\mathcal{O}(\varepsilon))OPT$ *with the constraint violation bounded by* $(1+\mathcal{O}(\varepsilon))B$ *where* $B = \min_{i=1,\ldots,m} b_i$*, if* $\sqrt{k}B^2 \geq \frac{n^{3/2}\log kn}{\varepsilon}$ *and* $\sqrt{k}B \geq \frac{m\sqrt{n}}{\varepsilon}$ *for any* $\varepsilon > 0$*. Moreover, if* $kn \geq m$*, the relative constraint violation can be bounded by* $(1+\mathcal{O}(\frac{\varepsilon}{\sqrt{m}}))$*.*

The proof comes from a direct application of performance analyses of the Simple Online Algorithm

Performance of the Variable-Replicating Algorithm

Proposition (Gao, Sun, Ye & Y (2021))

Under the random permutation model, the variable-replicating algorithm finds a solution for the original LP that achieves at least $(1-\mathcal{O}(\varepsilon))OPT$ *with the constraint violation bounded by* $(1+\mathcal{O}(\varepsilon))B$ *where* $B = \min_{i=1,\ldots,m} b_i$*, if* $\sqrt{k}B^2 \geq \frac{n^{3/2}\log kn}{\varepsilon}$ *and* $\sqrt{k}B \geq \frac{m\sqrt{n}}{\varepsilon}$ *for any* $\varepsilon > 0$*. Moreover, if* $kn \geq m$*, the relative constraint violation can be bounded by* $(1+\mathcal{O}(\frac{\varepsilon}{\sqrt{m}}))$.

The proof comes from a direct application of performance analyses of the Simple Online Algorithm

Takeaway: k times more computation cost for a $\sqrt{k}$ factor improvement in regret performance.

Multi-knapsack Problem Instances - Binary LP

Benchmark dataset of Chu & Beasley implementation

		V.R. Alg.	Gurobi
$m = 5, n = 500, k = 50$	Time	0.000	0.211
	Cmpt. Ratio	88.2%	95.3%
$m = 5, n = 500, k = 1000$	Time	0.007	0.211
	Cmpt. Ratio	89.2%	95.3%
$m = 8, n = 10^3, k = 50$	Time	0.004	3.800
	Cmpt. Ratio	89.9%	99.0%
$m = 8, n = 10^3, k = 1000$	Time	0.077	3.800
	Cmpt. Ratio	95.6%	99.0%
$m = 64, n = 10^4, k = 50$	Time	0.013	> 60
	Cmpt. Ratio	90.3%	98.7%
$m = 64, n = 10^4, k = 1000$	Time	0.223	> 60
	Cmpt. Ratio	96.4%	98.7%

Fast Online Algorithm as Pre-Classifier for LP

The key combinatorial task of LP is the partition of all variables into optimal basic (with positive values) and optimal nonbasic (with zero values) variables.

Fast Online Algorithm as Pre-Classifier for LP

The key combinatorial task of LP is the partition of all variables into optimal basic (with positive values) and optimal nonbasic (with zero values) variables.

In LP, a column generation techniques is popularly used when $n >> m$:

- Constructed a Restricted Master Problem (RMP) defined by a small subset of variables of the original problem
- Solve RMP and reselect initially unselected variables into RMP

Ideally, the initial RMP would already contain the set of $O(m)$ optimal basic variables and there is no need (or less) to do reselect!

Fast Online Algorithm as Pre-Classifier for LP

The key combinatorial task of LP is the partition of all variables into optimal basic (with positive values) and optimal nonbasic (with zero values) variables.

In LP, a column generation techniques is popularly used when $n >> m$:

- Constructed a Restricted Master Problem (RMP) defined by a small subset of variables of the original problem
- Solve RMP and reselect initially unselected variables into RMP

Ideally, the initial RMP would already contain the set of $O(m)$ optimal basic variables and there is no need (or less) to do reselect!

This is precisely where the fast online LP algorithm does a good job - classify variables being positive or zero at an optimal solution in a short time.

Implementation in LP Solvers

More precisely, the fast online LP solution can be interpreted as a "score" of how likely a variable is to be optimal basic.

We run online algorithm to obtain $\hat{\mathbf{x}}$, set a threshold ε and select the columns in $\mathbb{I}_{\{\hat{\mathbf{x}}>\varepsilon\}}$. For benchmark LP problems that have more columns than rows (such as **rail4284**, **s82**, and **scpm1** from the Mittelmann's Simplex Benchmark), the online solution identifies more than 90% of the primal optimal basis on average.

This technique has been adopted in the emerging LP solver COPT - a new state of art LP solver.

Table of Contents

1 Online Linear Programming

2 Fast Algorithms for (Binary) Online Linear Programming

3 A Fairer Online Interior-Point LP Algorithm

4 Online Bandits with Knapsacks

5 Online Fisher Markets

A “Fairer” Online LP Algorithm

Recall the online LP formulation (changing n to T as in the literature)

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

A "Fairer" Online LP Algorithm

Recall the online LP formulation (changing n to T as in the literature)

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

A finite-type assumption: $\mathbb{P}((r_t, \mathbf{a}_t) = (\mu_j, \mathbf{c}_j)) = p_j$ (unknown to the decision maker) for $j = 1, ..., J$. The offline problem with the knowledge:

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

where y_j is the acceptance probability for each customer type j.

A "Fairer" Online LP Algorithm

Recall the online LP formulation (changing n to T as in the literature)

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

A finite-type assumption: $\mathbb{P}((r_t, \mathbf{a}_t) = (\mu_j, \mathbf{c}_j)) = p_j$ (unknown to the decision maker) for $j = 1, ..., J$. The offline problem with the knowledge:

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

where y_j is the acceptance probability for each customer type j.

	Benchmark	Regret Bound	Key Assumption(s)
Jasin and Kumar (2012)	Fluid	Bounded	Nondeg., distrib. known
Jasin (2015)	Fluid	$\tilde{O}(\log T)$	Nondeg.
Vera et al. (2019)	Hindsight	Bounded	Distrib. known
Bumpensanti and Wang (2020)	Hindsight	Bounded	Distrib. known
Asadpour et al. (2019)	Full flex.	Bounded	Long-chain, ξ-Hall condition
Chen, Li & Y (2021)	Fluid	Bounded	Partial Nondeg.

Behavior of the Simplex and Interior-Point

The key in Chen et al. (2021) paper is to use the interior-point algorithm for solving the sample LPs with sample proportion $\hat{p}_j$

$$\max \sum_{j=1}^{J} \hat{p}_j \mu_j y_j \quad \text{s.t.} \quad \sum_{j=1}^{J} \hat{p}_j \mathbf{c}_j y_j \leq \mathbf{b}/T, \quad y_j \in [0, 1],$$

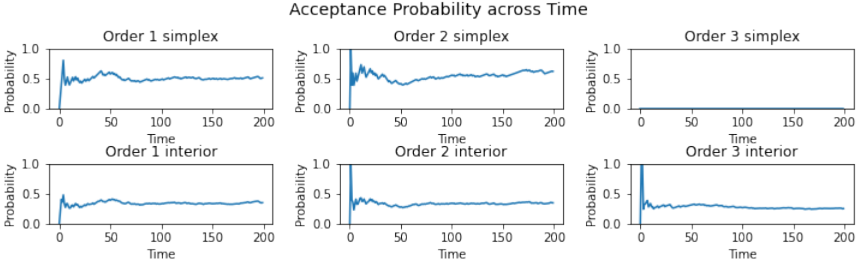
since the sample and offline LP may be degenerate or with multiple optimal solutions - a common property for real-life LP problems.

Behavior of the Simplex and Interior-Point

The key in Chen et al. (2021) paper is to use the interior-point algorithm for solving the sample LPs with sample proportion $\hat{p}_j$

$$\max \sum_{j=1}^{J} \hat{p}_j \mu_j y_j \quad \text{s.t.} \quad \sum_{j=1}^{J} \hat{p}_j \mathbf{c}_j y_j \le \mathbf{b}/T, \quad y_j \in [0,1],$$

since the sample and offline LP may be degenerate or with multiple optimal solutions - a common property for real-life LP problems.



Fairness Desiderata: Time and Individual

Time Fairness: The algorithm may tends to accept mainly the first half (or the second half of the orders), which is unfair or unideal such as Adwords application.

Fairness Desiderata: Time and Individual

Time Fairness: The algorithm may tends to accept mainly the first half (or the second half of the orders), which is unfair or unideal such as Adwords application.

Individual Fairness: For certain customer types there exist **multiple** optimal allocation rules. Unfortunately, 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 the online algorithm/allocation-rule.

Fairness Desiderata: Time and Individual

Time Fairness: The algorithm may tends to accept mainly the first half (or the second half of the orders), which is unfair or unideal such as Adwords application.

Individual Fairness: For certain customer types there exist **multiple** optimal allocation rules. Unfortunately, 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 the online algorithm/allocation-rule.

But these individuals/groups could have different sensitive features, such as demographic, race, and gender, and areas in Hospital Admission and Hotel/Flight booking application.

Fairness Desiderata: Time and Individual

Time Fairness: The algorithm may tends to accept mainly the first half (or the second half of the orders), which is unfair or unideal such as Adwords application.

Individual Fairness: For certain customer types there exist **multiple** optimal allocation rules. Unfortunately, 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 the online algorithm/allocation-rule.

But these individuals/groups could have different sensitive features, such as demographic, race, and gender, and areas in Hospital Admission and Hotel/Flight booking application.

Could we design an online algorithm/allocation-rule such as, while maintain the efficiency in objective value, all individual/groups get a fairer allocation shares?

Fairer Solution for the Offline Problem

We define $\mathbf{y}^*$, the fair offline optimal solution of the LP problem

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

as the analytical center of the optimal solution set, which represents an "average" of all the corner optimal solutions.

Fairer Solution for the Offline Problem

We define $\mathbf{y}^*$, the fair offline optimal solution of the LP problem

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

as the analytical center of the optimal solution set, which represents an "average" of all the corner optimal solutions.

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

$$\mathrm{UF}_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{y}^*\|_2^2\right].$$

Fairer Solution for the Offline Problem

We define $\mathbf{y}^*$, the fair offline optimal solution of the LP problem

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

as the analytical center of the optimal solution set, which represents an "average" of all the corner optimal solutions.

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

$$\mathrm{UF}_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{y}^*\|_2^2\right].$$

This definition is consistent with the definition of fair classifiers/regressors in fair machine learning.

Our Result

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

$$\mathrm{UF}_T(\pi) = O(\log T)$$

$$\mathrm{Reg}_T(\pi) = \text{ Bounded w.r.t } T$$

Our Result

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

$$\mathrm{UF}_T(\pi) = O(\log T)$$

$$\mathrm{Reg}_T(\pi) = \text{ Bounded w.r.t } T$$

Key ideas in algorithm design:

- At each time t, we use interior-point method to obtain the sample analytic-center solution $\mathbf{y}_t$, and it is necessary to achieve the performance under weak non-degeneracy assumption and maintain fairness.
- We also adjust the right-hand-side of the LP constraints properly to ensure (i) the depletion of binding resources and (ii) non-binding resources not affecting the fairness.

The use of interior-point method also relaxes a non-degeneracy assumption in previous analysis

Table of Contents

1 Online Linear Programming

2 Fast Algorithms for (Binary) Online Linear Programming

3 A Fairer Online Interior-Point LP Algorithm

4 Online Bandits with Knapsacks

5 Online Fisher Markets

Bandits with Knapsacks

Reverse the order of decisions and observations in online LP setting: in each time t, decide a customer/order/arm among k arms to sell/play and then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Bandits with Knapsacks

Reverse the order of decisions and observations in online LP setting: in each time t, decide a customer/order/arm among k arms to sell/play and then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

Bandits with Knapsacks

Reverse the order of decisions and observations in online LP setting: in each time t, decide a customer/order/arm among k arms to sell/play and then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

Bandits: k arms, where each arm i with an unknown mean reward μ_i,.

Bandits with Knapsacks

Reverse the order of decisions and observations in online LP setting: in each time t, decide a customer/order/arm among k arms to sell/play and then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

Bandits: k arms, where each arm i with an unknown mean reward μ_i,.

Knapsacks: m types of resources. The total resource capacity $\mathbf{b} \in \mathbb{R}^m$. Each arm i with an unknown mean resource consumption $\mathbf{c}_i \in \mathbb{R}^m$.

Bandits with Knapsacks

Reverse the order of decisions and observations in online LP setting: in each time t, decide a customer/order/arm among k arms to sell/play and then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

Bandits: k arms, where each arm i with an unknown mean reward μ_i,.

Knapsacks: m types of resources. The total resource capacity $\mathbf{b} \in \mathbb{R}^m$. Each arm i with an unknown mean resource consumption $\mathbf{c}_i \in \mathbb{R}^m$.

At each time $t \in [T]$, an arm i is selected to pull. The realized reward $\hat{r}_t$ and resources cost $\hat{\mathbf{c}}_t$ satisfying

$$\mathbb{E}[\hat{r}_t | i] = \mu_i, \quad \mathbb{E}[\hat{\mathbf{c}}_t | i] = \mathbf{c}_i.$$

Bandits with Knapsacks

Reverse the order of decisions and observations in online LP setting: in each time t, decide a customer/order/arm among k arms to sell/play and then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

Bandits: k arms, where each arm i with an unknown mean reward μ_i,.

Knapsacks: m types of resources. The total resource capacity $\mathbf{b} \in \mathbb{R}^m$. Each arm i with an unknown mean resource consumption $\mathbf{c}_i \in \mathbb{R}^m$.

At each time $t \in [T]$, an arm i is selected to pull. The realized reward $\hat{r}_t$ and resources cost $\hat{\mathbf{c}}_t$ satisfying

$$\mathbb{E}[\hat{r}_t | i] = \mu_i, \quad \mathbb{E}[\hat{\mathbf{c}}_t | i] = \mathbf{c}_i.$$

Goal: Select a subset of winning/optimal arms to maximize the total reward subject to the resource capacity constraints - pro-actively explore arms and exploit learned data.

Offline Linear Program (LP) and Regret

With mean reward $\boldsymbol{\mu} = (\mu_1, ..., \mu_k)$ and mean resource-cost $(\mathbf{c}_1, ..., \mathbf{c}_k)$ of arms, consider the following deterministic offline LP,

$$\max_{\mathbf{x}} \sum_{i=1}^{k} \mu_i x_i \quad \text{s.t.} \quad \sum_{i=1}^{k} \mathbf{c}_i x_i \leq \mathbf{b}, x_i \geq \mathbf{0}, i \in [k]$$

Here x_i represents the optimal fractional number of playing i-th arm if everything is deterministic and known

Offline Linear Program (LP) and Regret

With mean reward $\boldsymbol{\mu} = (\mu_1, ..., \mu_k)$ and mean resource-cost $(\mathbf{c}_1, ..., \mathbf{c}_k)$ of arms, consider the following deterministic offline LP,

$$\max_{\mathbf{x}} \sum_{i=1}^{k} \mu_i x_i \quad \text{s.t.} \quad \sum_{i=1}^{k} \mathbf{c}_i x_i \leq \mathbf{b}, x_i \geq \mathbf{0}, i \in [k]$$

Here x_i represents the optimal fractional number of playing i-th arm if everything is deterministic and known

Denote its optimal value as OPT (the benchmark) and let τ be the stopping time as soon as one of the resources is depleted. Then the problem-dependent regret

$$Regret(\mathcal{P}) = OPT - \mathbb{E}\left[\sum_{t=1}^{\tau} r_t\right],$$

where $\mathcal{P}$ encapsulates the parameters related to the underlying data distribution.

Literature and Our Result

	Paper	Result
$\mathcal{P}$-Independent	Badanidiyuru et. al. (13) Agrawal and Devanur (14)	$O(poly(m, k) \cdot \sqrt{T})$
$\mathcal{P}$-Dependent	Flajolet and Jaillet (15) Sankararaman and Slivkins (20) Li, Sun & Y (21)	$\tilde{O}(2^{m+k} \log T)$ $\tilde{O}(k \log T)$ for $m = 1$ $\tilde{O}\left(m^4 + k \log T\right)$

The problem-dependent bounds all involve parameters related to the non-degeneracy and the reduced cost of the underlying LP, while our work has the mildest assumption and requires no prior knowledge of these parameters.

Dual LP and Reduced Cost

$$
\begin{array}{llll}
\textit{Primal}: & \max & \mu^\top \mathbf{x} & \\
& \text{s.t.} & C\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0} &
\end{array}
\qquad
\begin{array}{llll}
\textit{Dual}: & \min & \mathbf{b}^\top \mathbf{y} \\
& \text{s.t.} & C^\top \mathbf{y} \ge \mu, \mathbf{y} \ge \mathbf{0}
\end{array}
$$

Denote $\mathbf{x}^* \in R^k$ and $\mathbf{y}^* \in R^m$ as optimal solutions

Define reduced cost (profit) for i-th arm $\Delta_i := \mathbf{c}_i^\top \mathbf{y}^* - \mu_i$ and the non-basic variable set $\mathcal{I}' = \{i : \Delta_i > 0\}$.

Proposition (Li, Sun & Y (2021, ICML))

The regret of a BwK algorithm has the following upper bound:

$$Regret(\mathcal{P}) \le \sum_{i \in \mathcal{I}'} \Delta_i \mathbb{E}[n_i(\tau)] + \mathbb{E}[\mathbf{b}^{(\tau)}]^\top \mathbf{y}^*$$

- $\mathbf{b}^{(t)}$: remaining resource at time t
- $n_i(t)$: the number of times that i-th (non-optimal) arm is played up to time t

Implications of the Regret Upper Bound

Two tasks to accomplish to reduce the regret:

Task I: Control the number of plays $n_i(\tau)$ for non-optimal arms $i \in \mathcal{I}'$ which corresponds to the first component in the regret bound

$$\sum_{i \in \mathcal{I}'} \Delta_i \mathbb{E}[n_i(\tau)]$$

Playing each non-optimal arm will induce a cost/waste of Δ_i.

Implications of the Regret Upper Bound

Two tasks to accomplish to reduce the regret:

Task I: Control the number of plays $n_i(\tau)$ for non-optimal arms $i \in \mathcal{I}'$ which corresponds to the first component in the regret bound

$$\sum_{i \in \mathcal{I}'} \Delta_i \mathbb{E}[n_i(\tau)]$$

Playing each non-optimal arm will induce a cost/waste of Δ_i.

Task II: Make sure no valuable resources $\mathbf{b}_j^{(\tau)}$ left unused, which corresponds to the second component in the regret bound

$$\mathbb{E}[\mathbf{b}^{(\tau)}]^\top \mathbf{y}^*$$

Recall τ is the time that one of the resources is exhausted.

Implications of the Regret Upper Bound

Two tasks to accomplish to reduce the regret:

Task I: Control the number of plays $n_i(\tau)$ for non-optimal arms $i \in \mathcal{I}'$ which corresponds to the first component in the regret bound

$$\sum_{i \in \mathcal{I}'} \Delta_i \mathbb{E}[n_i(\tau)]$$

Playing each non-optimal arm will induce a cost/waste of Δ_i.

Task II: Make sure no valuable resources $\mathbf{b}_j^{(\tau)}$ left unused, which corresponds to the second component in the regret bound

$$\mathbb{E}[\mathbf{b}^{(\tau)}]^\top \mathbf{y}^*$$

Recall τ is the time that one of the resources is exhausted.

Task II is often overlooked in the existing BwK literature.

Our Approach: A Two-Phase Algorithm

- Phase I: Identify the optimal arms with as fewer number of plays as possible by designing an "importance score" for arm i:

$$
\begin{aligned}
OPT_i := \max \quad & \boldsymbol{\mu}^\top \mathbf{x} \\
\text{s.t.} \quad & C\mathbf{x} \le \mathbf{b},\ x_i = 0, \mathbf{x} \ge \mathbf{0}.
\end{aligned}
$$

  Implication: A larger value of $OPT - OPT_i \Rightarrow x_i$ important and likely to represent an optimal arm. Our algorithm then maintains upper confidence bound (UCB)/lower confidence bound (LCB) to estimate OPT and OPT_i based are samples.

Our Approach: A Two-Phase Algorithm

- Phase I: Identify the optimal arms with as fewer number of plays as possible by designing an "importance score" for arm i:

$$
\begin{aligned}
OPT_i := \max \quad & \boldsymbol{\mu}^\top \mathbf{x} \\
\text{s.t.} \quad & C\mathbf{x} \leq \mathbf{b}, \; x_i = 0, \mathbf{x} \geq \mathbf{0}.
\end{aligned}
$$

  Implication: A larger value of $OPT - OPT_i \Rightarrow x_i$ important and likely to represent an optimal arm. Our algorithm then maintains upper confidence bound (UCB)/lower confidence bound (LCB) to estimate OPT and OPT_i based are samples.

  After $t' = O(\frac{k \log T}{\sigma^2 \delta^2})$ times of Phase I, the non-optimal arm variables are identified as set $\mathcal{I}'$ and they would be removed from further consideration, and then we start

- Phase II: Use the remaining arms to exhaust the resource through an adaptive procedure such that no valuable resources are wasted.

Adaptive Algorithm for filling the knapsacks:

For $t = t' + 1, ..., T$

1 Solve the UCB-LP and denote its optimal solution as $\tilde{\mathbf{x}}$

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{i=1}^{k} \left(\hat{\mu}_i(t) + \sqrt{\frac{2 \log T}{n_i(t)}} \right) x_i \\ \text{s.t.} \quad & \sum_{i=1}^{k} \left(\hat{\mathbf{c}}_i(t) - \sqrt{\frac{2 \log T}{n_i(t)}} \right) x_i \leq \mathbf{b}^{(t-1)} \\ & \mathbf{x} \geq \mathbf{0}, x_i = 0 \text{ for } i \in \mathcal{I}' \end{aligned}$$

2 Normalize $\tilde{\mathbf{x}}$ into a probability and play an arm accordingly

3 Update the knapsack process $\mathbf{b}^{(t)}$ (remaining resource)

Combining the Two Phases

Proposition (Li, Sun & Y 2021, ICML)

The regret of our two-phase algorithm is bounded by

$$O\left(\frac{m^4}{\sigma^2\delta^2} + \frac{k\log T}{\delta^2}\right).$$

Combining the Two Phases

Proposition (Li, Sun & Y 2021, ICML)

The regret of our two-phase algorithm is bounded by

$$O\left(\frac{m^4}{\sigma^2\delta^2} + \frac{k \log T}{\delta^2}\right).$$

Here the problem-dependent conditional numbers of the deterministic BwK LP problem are:

- σ is the minimum singular value of the sub-matrix of the constraint matrix C that corresponds to the optimal basis.

Combining the Two Phases

Proposition (Li, Sun & Y 2021, ICML)

The regret of our two-phase algorithm is bounded by

$$O\left(\frac{m^4}{\sigma^2\delta^2} + \frac{k \log T}{\delta^2}\right).$$

Here the problem-dependent conditional numbers of the deterministic BwK LP problem are:

- σ is the minimum singular value of the sub-matrix of the constraint matrix C that corresponds to the optimal basis.
- δ measures the difficulty of identifying optimal basic variables:

$$\min\left\{\min\{x_i^*|x_i^*>0\}, \min\{OPT - OPT_i|x_i^*>0\}, \min\{\Delta_i|x_i^*=0\}\right\}.$$

Combining the Two Phases

Proposition (Li, Sun & Y 2021, ICML)

The regret of our two-phase algorithm is bounded by

$$O\left(\frac{m^4}{\sigma^2\delta^2} + \frac{k \log T}{\delta^2}\right).$$

Here the problem-dependent conditional numbers of the deterministic BwK LP problem are:

- σ is the minimum singular value of the sub-matrix of the constraint matrix C that corresponds to the optimal basis.
- δ measures the difficulty of identifying optimal basic variables:

$$\min\left\{\min\{x_i^*|x_i^* > 0\}, \min\{OPT - OPT_i|x_i^* > 0\}, \min\{\Delta_i|x_i^* = 0\}\right\}.$$

These condition numbers generalize the optimality gap for the original (unconstrained) multi-armed bandits (Lai and Robbins (1985), Auer et al. (2002)).

Table of Contents

1 Online Linear Programming

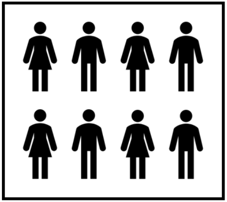
2 Fast Algorithms for (Binary) Online Linear Programming

3 A Fairer Online Interior-Point LP Algorithm

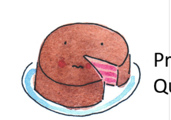
4 Online Bandits with Knapsacks

5 Online Fisher Markets

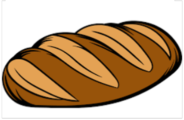
Fisher Markets for Resource-Allocation



Each agent i , with budget w_i, purchases an optimal bundle $\boldsymbol{x}_i$ given price $\boldsymbol{p}$



Price p_1
Quant.: c_1



p_2 , c_2

p_3 , c_3

How to setup "prices" for each good so that goods can be wholly allocated while keep each individual buyer/agent satisfied?

The Model: Fisher's Equilibrium Price

Buyer $i \in B$'s optimization problem for given prices p_j, $j \in G$.

$$\begin{array}{rl} \max & \mathbf{u}_i^T \mathbf{x}_i := \sum_{j \in G} u_{ij} x_{ij} \\ \text{s.t.} & \mathbf{p}^T \mathbf{x}_i := \sum_{j \in G} p_j x_{ij} \le w_i, \\ & x_{ij} \ge 0, \quad \forall j, \end{array}$$

Assume that the given amount of each good is c_j. The equilibrium price vector is the one that for all $j \in G$

$$\sum_{i \in B} x^*(\mathbf{p})_{ij} = c_j$$

where $\mathbf{x}^*(\mathbf{p})$ is a maximizer of the utility maximization problem for every buyer i.

The Aggregated Social Optimization Problem

$$
\begin{aligned}
\max \quad & \textstyle\sum_{i\in B} w_i \log(\mathbf{u}_i^T \mathbf{x}_i) \\
\text{s.t.} \quad & \textstyle\sum_{i\in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\
& x_{ij} \geq 0, \quad \forall i, j,
\end{aligned}
$$

The Aggregated Social Optimization Problem

$$\begin{array}{rl} \max & \sum_{i \in B} w_i \log(\mathbf{u}_i^T \mathbf{x}_i) \\ \text{s.t.} & \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\ & x_{ij} \geq 0, \quad \forall i, j, \end{array}$$

Theorem (Eisenberg and Gale (1959))

Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector.

The Aggregated Social Optimization Problem

$$\begin{array}{rl} \max & \sum_{i \in B} w_i \log(\mathbf{u}_i^T \mathbf{x}_i) \\ \text{s.t.} & \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\ & x_{ij} \geq 0, \quad \forall i, j, \end{array}$$

Theorem (Eisenberg and Gale (1959))

Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector.

Now, consider the online setting: buyers/agents arrive Online and an irrevocable allocation has to be made.

The Aggregated Social Optimization Problem

$$\begin{array}{rl} \max & \sum_{i \in B} w_i \log(\mathbf{u}_i^T \mathbf{x}_i) \\ \text{s.t.} & \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\ & x_{ij} \geq 0, \quad \forall i, j, \end{array}$$

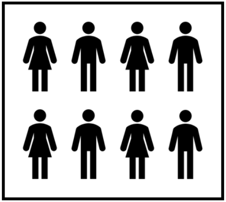
Theorem (Eisenberg and Gale (1959))

Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector.

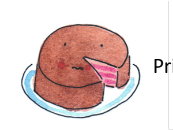
Now, consider the online setting: buyers/agents arrive Online and an irrevocable allocation has to be made.

Question: how much would the aggregated social welfare be deteriorated from the offline setting? Could the algorithm be implemented by protecting privacy?

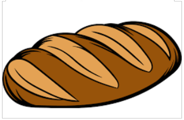
Online Fisher Markets



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



Price p_1^t



p_2^t

p_3^t

How to setup $\mathbf{p}^t$ for each good before buyer t comes so that the social welfare is maximized and capacity constraint violation is minimized?

Regret Analysis and Model

Let "offline" optimal solution be $\mathbf{x}^*$ and "online" solution of n orders be $\mathbf{x}$, and

$$R_n^* = \sum_{j=1}^{n}\sum_{i=1}^{n} w_i \log(\mathbf{u}_i^T \mathbf{x}_i^*), \quad R_n = \sum_{i=1}^{n} w_i \log(\mathbf{u}_i^T \mathbf{x}_i)$$

Regret Analysis and Model

Let "offline" optimal solution be $\mathbf{x}^*$ and "online" solution of n orders be $\mathbf{x}$, and

$$R_n^* = \sum_{j=1}^{n} \sum_{i=1}^{n} w_i \log(\mathbf{u}_i^T \mathbf{x}_i^*), \quad R_n = \sum_{i=1}^{n} w_i \log(\mathbf{u}_i^T \mathbf{x}_i)$$

Then define

$$\Delta_n = \sup \mathbb{E}\left[R_n^* - R_n\right], \quad v(\mathbf{x}) = \sup \mathbb{E}\left[\| (A\mathbf{x} - \mathbf{b})^+ \|_2\right]$$

where the expectation is taken with respect to i.i.d distribution, and the sup operator is over all permissible distributions and admissible data.

Regret Analysis and Model

Let "offline" optimal solution be $\mathbf{x}^*$ and "online" solution of n orders be $\mathbf{x}$, and

$$R_n^* = \sum_{j=1}^{n} \sum_{i=1}^{n} w_i \log(\mathbf{u}_i^T \mathbf{x}_i^*), \quad R_n = \sum_{i=1}^{n} w_i \log(\mathbf{u}_i^T \mathbf{x}_i)$$

Then define

$$\Delta_n = \sup \mathbb{E}\left[R_n^* - R_n\right], \quad v(\mathbf{x}) = \sup \mathbb{E}\left[\| (A\mathbf{x} - \mathbf{b})^+ \|_2\right]$$

where the expectation is taken with respect to i.i.d distribution, and the sup operator is over all permissible distributions and admissible data.

Remark: Again this a bi-criteria performance measure and, if $\Delta_n \leq o(n)$ (sublinear), then

$$\frac{(\prod_i (\mathbf{u}_i^T \mathbf{x}_i^*)^{w_i})^{1/n}}{(\prod_i (\mathbf{u}_i^T \mathbf{x}_i)^{w_i})^{1/n}} \leq e^{o(n)/n}.$$

Simple Price-Learning Algorithm

One may apply a similar primal price-learning algorithm, that is, solve the aggregated social problem based on arrived ϵ portion of buyers:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{\epsilon n} w_t \log(\mathbf{u}_t^T \mathbf{x}_t) & \\ \text{subject to} & \sum_{t=1}^{\epsilon n} \mathbf{x}_t \leq \epsilon c_j, & j = 1, ..., m \\ & 0 \leq x_t. & \end{array}$$

One can set an initial positive price vector $\mathbf{p}^1$ and determine allocation $\mathbf{x}_t$ as the optimal solution for the individual maximization problem under price vector $\mathbf{p}^t$.

Simple Price-Learning Algorithm

One may apply a similar primal price-learning algorithm, that is, solve the aggregated social problem based on arrived ϵ portion of buyers:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{\epsilon n} w_t \log(\mathbf{u}_t^T \mathbf{x}_t) & \\ \text{subject to} & \sum_{t=1}^{\epsilon n} \mathbf{x}_t \leq \epsilon c_j, & j = 1, ..., m \\ & 0 \leq x_t. & \end{array}$$

One can set an initial positive price vector $\mathbf{p}^1$ and determine allocation $\mathbf{x}_t$ as the optimal solution for the individual maximization problem under price vector $\mathbf{p}^t$.

The price update needs to have full information of each buyer, which could be private!

Simple Price-Learning Algorithm

One may apply a similar primal price-learning algorithm, that is, solve the aggregated social problem based on arrived ϵ portion of buyers:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{\epsilon n} w_t \log(\mathbf{u}_t^T \mathbf{x}_t) & \\ \text{subject to} & \sum_{t=1}^{\epsilon n} \mathbf{x}_t \leq \epsilon c_j, & j = 1, ..., m \\ & 0 \leq x_t. & \end{array}$$

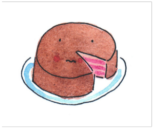
One can set an initial positive price vector $\mathbf{p}^1$ and determine allocation $\mathbf{x}_t$ as the optimal solution for the individual maximization problem under price vector $\mathbf{p}^t$.

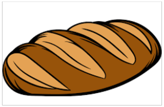
The price update needs to have full information of each buyer, which could be private!

Is there an online algorithm that relies on only $\mathbf{x}_t$?

Negative Results: Two-Good Example

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)

Arrival Probability = 0.5

Type II: (0, 1)

Arrival Probability = 0.5

Theorem (Jelota & Y (2022))

The expect optimal social value

$$n\log(2) - 1 \le \mathbb{E}[R_n^*] \le n\log(2).$$

For any static pricing policy, either the expected regret or constraint violation is $\Omega\sqrt{n}$. Even using the optimal expected equilibrium prices Even using the optimal expect equilibrium prices for online allocation,

$$\mathbb{E}[\|(A\mathbf{x} - \mathbf{b})^+\|_2] \ge \sqrt{n}.$$

Consider the Dual of the Fisher Market

$$\min \ \mathbf{c}^\top \mathbf{p} - \sum_{t=1}^{n} w_t \log\left(\min_j \frac{p_j}{u_{tj}}\right) + \sum_{t=1}^{n} w_t(\log(w_t) - 1).$$

It can be, after removing the fixed part, equivalently rewritten as

$$\min \ (\frac{1}{n}\mathbf{c})^\top \mathbf{p} - \frac{1}{n}\sum_{t=1}^{n} w_t \log\left(\min_j \frac{p_j}{u_{tj}}\right)$$

which can be viewed as a simple-sample-average (SSA) (with n buyers) of a stochastic optimization problem under an i.i.d distribution, where $\mathbf{d} := \frac{1}{n}\mathbf{c}$ is the average resource allocation to each buyer.

Dual-Gradient Online Algorithm for Fisher-Markets

1: Initialize $\mathbf{p}^1 = \epsilon\mathbf{e}$, and for $t = 1, 2, ..., n$

2: Let $\mathbf{x}_t$ be the individual optimal bundle solution for price vector $\mathbf{p}^t$.

3: Update prices

$$\mathbf{p}_{t+1} = \mathbf{p}_t - \gamma_t \left(\mathbf{d} - \mathbf{x}_t\right)$$
$$\mathbf{p}_{t+1} = \mathbf{p}_{t+1} \vee \mathbf{0}$$

4: $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$

Again, line 3 performs (projected) stochastic gradient step.

Dual-Gradient Online Algorithm for Fisher-Markets

1: Initialize $\mathbf{p}^1 = \epsilon\mathbf{e}$, and for $t = 1, 2, ..., n$
2: Let $\mathbf{x}_t$ be the individual optimal bundle solution for price vector $\mathbf{p}^t$.
3: Update prices

$$\mathbf{p}_{t+1} = \mathbf{p}_t - \gamma_t (\mathbf{d} - \mathbf{x}_t)$$
$$\mathbf{p}_{t+1} = \mathbf{p}_{t+1} \vee \mathbf{0}$$

4: $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$

Again, line 3 performs (projected) stochastic gradient step.

Theorem (Jelota & Y (2022))

Under i.i.d. budget and utility parameters and when good capacities are $O(n)$, the algorithm achieves an expected regret $\Delta_n \leq O(\sqrt{n})$ and the expected constraint violation $v(\mathbf{x}) \leq O(\sqrt{n})$, where n is the number of arriving buyers.

Takeaways and Open Problems

- Geometrically aggregated welfare optimization is as easy as linear programming and more desirable in many social/economical settings.
- Tight upper and lower regret bounds for geometrically aggregated online social optimization?
- Geometrical aggregation to Bandit/MDP and other learning?
- Extensions to non-divisible goods for Fisher markets?
- Linear Programming continues to play a big role in online learning and decisioning.
- Could non-stationary data be learned with sub-linear regret?

Takeaways and Open Problems

- Geometrically aggregated welfare optimization is as easy as linear programming and more desirable in many social/economical settings.
- Tight upper and lower regret bounds for geometrically aggregated online social optimization?
- Geometrical aggregation to Bandit/MDP and other learning?
- Extensions to non-divisible goods for Fisher markets?
- Linear Programming continues to play a big role in online learning and decisioning.
- Could non-stationary data be learned with sub-linear regret?

OLP provides a data-driven and adaptive-learning policy/mechanism for decision making in real time...