Online Linear Programming: Applications and Extensions

Yinyu Ye

Stanford University and CUHK SZ (Sabbatical Leave)
(Currently Visiting CUHK and HK PolyU)

CUHK Business School Seminar
October 6, 2022
(Joint work with many...)
Table of Contents

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

\[
\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{n} r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^{n} a_{t} x_{t} \leq b, \\
& \quad 0 \leq x_{t} \leq 1, \quad \forall t = 1, \ldots, n.
\end{align*}
\]
Linear Programming

maximize \( \sum_{t=1}^{n} r_t x_t \)
subject to \( \sum_{t=1}^{n} a_t x_t \leq b, \)
\( 0 \leq x_t \leq 1, \quad \forall t = 1, \ldots, n. \)
Consider an auction/revenue-management problem:

<table>
<thead>
<tr>
<th></th>
<th>Bid 1 ((t = 1))</th>
<th>Bid 2 ((t = 2))</th>
<th>.....</th>
<th>Inventory ((b))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reward ((r_t))</td>
<td>$100</td>
<td>$30</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Decision</td>
<td>(x_1)</td>
<td>(x_2)</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Pants</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>100</td>
</tr>
<tr>
<td>Shoes</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>50</td>
</tr>
<tr>
<td>T-shirts</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>500</td>
</tr>
<tr>
<td>Jackets</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>200</td>
</tr>
<tr>
<td>Hats</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1000</td>
</tr>
</tbody>
</table>

where the decision for each customer/bidder is “accept” \((x_t = 1)\) or “reject” \((x_t = 0)\)
Offline vs. Online Linear Programming

\[ OPT(A, r) := \max_{x} \sum_{t=1}^{n} r_t x_t \]
subject to \[ \sum_{t=1}^{n} a_t x_t \leq b, \]
\[ x_t \in \{0, 1\} \quad (0 \leq x_t \leq 1), \quad \forall t = 1, \ldots, n. \]
Offline vs. Online Linear Programming

\[
\text{OPT}(A, r) := \max_x \sum_{t=1}^n r_t x_t \\
\text{subject to} \sum_{t=1}^n a_t x_t \leq b,
\]
\[
x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \ \forall t = 1, \ldots, n.
\]

\(r_t\): reward/revenue offered by the \(t\)-th customer/order
\(a_t \in R^m\): the bundle of resources requested by the \(t\)-th order
\(x_t\): acceptance or rejection decision to the \(t\)-th order
\(b \in R^m\): initially available budget/resource amounts

The objective \(\sum_{t=1}^n r_t x_t\): the total collected revenue.
Offline vs. Online Linear Programming

$$OPT(A, r) := \maximize_{x} \sum_{t=1}^{n} r_t x_t$$
subject to
$$\sum_{t=1}^{n} a_t x_t \leq b,$$
$$x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \ \forall t = 1, \ldots, n.$$  

- $r_t$: reward/revenue offered by the $t$-th customer/order
- $a_t \in R^m$: the bundle of resources requested by the $t$-th order
- $x_t$: acceptance or rejection decision to the $t$-th order
- $b \in R^m$: initially available budget/resource amounts

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

- We know only $b$ and $n$ at the start.
Offline vs. Online Linear Programming

\[ OPT(A, r) := \text{maximize} \sum_{t=1}^{n} r_t x_t \]
\[ \text{subject to} \quad \sum_{t=1}^{n} a_t x_t \leq b, \]
\[ x_t \in \{0, 1\} \quad (0 \leq x_t \leq 1), \quad \forall t = 1, \ldots, n. \]

- \( r_t \): reward/revenue offered by the \( t \)-th customer/order
- \( a_t \in \mathbb{R}^m \): the bundle of resources requested by the \( t \)-th order
- \( x_t \): acceptance or rejection decision to the \( t \)-th order
- \( b \in \mathbb{R}^m \): initially available budget/resource amounts

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

- We know only \( b \) and \( n \) at the start.
- the bidder data \((r_t, a_t)\) arrive sequentially.
Offline vs. Online Linear Programming

\[ \text{OPT}(A, r) := \maximize_{x} \sum_{t=1}^{n} r_t x_t \]
subject to \[ \sum_{t=1}^{n} a_t x_t \leq b, \]
\[ x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \ \forall t = 1, \ldots, n. \]

- \( r_t \): reward/revenue offered by the \( t \)-th customer/order
- \( a_t \in R^m \): the bundle of resources requested by the \( t \)-th order
- \( x_t \): acceptance or rejection decision to the \( t \)-th order
- \( b \in R^m \): initially available budget/resource amounts

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

- We know only \( b \) and \( n \) at the start.
- the bidder data \((r_t, a_t)\) arrive sequentially.
- an irrevocable decision must be made as soon as an order arrives (without knowing the future data).
Offline vs. Online Linear Programming

\[ \text{OPT}(A, r) := \maximize_{x} \sum_{t=1}^{n} r_t x_t \]
\[ \text{subject to } \sum_{t=1}^{n} a_t x_t \leq b, \]
\[ x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \ \forall t = 1, \ldots, n. \]

- \( r_t \): reward/revenue offered by the \( t \)-th customer/order
- \( a_t \in R^m \): the bundle of resources requested by the \( t \)-th order
- \( x_t \): acceptance or rejection decision to the \( t \)-th order
- \( b \in R^m \): initially available budget/resource amounts

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

- We know only \( b \) and \( n \) at the start.
- the bidder data \((r_t, a_t)\) 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.
The problem would be easy if there are “ideal itemized prices”:

<table>
<thead>
<tr>
<th>Bid (r_t)</th>
<th>Bid 1((t = 1))</th>
<th>Bid 2((t = 2))</th>
<th>……</th>
<th>Inventory((b))</th>
<th>(p^*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td>(x_1 = 0)</td>
<td>(x_2 = 1)</td>
<td>……</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pants</td>
<td>1</td>
<td>0</td>
<td>……</td>
<td>100</td>
<td>$45</td>
</tr>
<tr>
<td>Shoes</td>
<td>1</td>
<td>0</td>
<td>……</td>
<td>50</td>
<td>$45</td>
</tr>
<tr>
<td>T-shirts</td>
<td>0</td>
<td>1</td>
<td>……</td>
<td>500</td>
<td>$10</td>
</tr>
<tr>
<td>Jackets</td>
<td>0</td>
<td>0</td>
<td>……</td>
<td>200</td>
<td>$55</td>
</tr>
<tr>
<td>Hats</td>
<td>1</td>
<td>1</td>
<td>……</td>
<td>1000</td>
<td>$15</td>
</tr>
</tbody>
</table>

so that the online decision can be made by comparing the *reward* and “bundle cost” for each bid.
Primal and Dual Offline LPs

\[
\begin{align*}
\max & \quad r^\top x \\
\text{P:} & \quad \text{s.t.} \quad Ax \leq b \\
& \quad 0 \leq x \leq e
\end{align*}
\]

\[
\begin{align*}
\min & \quad b^\top p + e^\top s \\
\text{D:} & \quad \text{s.t.} \quad A^\top p + s \geq r \\
& \quad p \geq 0, s \geq 0
\end{align*}
\]

where the decision variables are \( x \in \mathbb{R}^n \), \( p \in \mathbb{R}^m \), \( s \in \mathbb{R}^n \), where \( e \) is the vector of all ones.
Primal and Dual Offline LPs

\[
\begin{align*}
\text{max} & \quad r^\top x & \quad \text{min} & \quad b^\top p + e^\top s \\
\text{P:} & \quad \text{s.t.} & \quad A x \leq b & \quad \text{D:} & \quad \text{s.t.} & \quad A^\top p + s \geq r \\
& & \quad 0 \leq x \leq e & & \quad p \geq 0, s \geq 0
\end{align*}
\]

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

Denote the primal/dual optimal solution as \( x^*, p^*, s^* \), then LP duality/complementarity theory tells that for \( t = 1, \ldots, n \),

\[
x_t^* = \begin{cases} 
1, & r_t > a_t^\top p^* \\
0, & r_t < a_t^\top p^*
\end{cases}
\]

(few \( x_t^* \) may take non-integer value when \( r_t = a_t^\top p^* \)).
Primal and Dual Offline LPs

\[
\begin{align*}
\text{max} \quad r^\top x \\
\text{s.t.} \quad Ax & \leq b \\
0 & \leq x \leq e
\end{align*}
\]

\[
\begin{align*}
\text{min} \quad b^\top p + e^\top s \\
\text{s.t.} \quad A^\top p + s & \geq r \\
p & \geq 0, s \geq 0
\end{align*}
\]

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

Denote the primal/dual optimal solution as \(x^*, p^*, s^*\), then LP duality/complementarity theory tells that for \(t = 1, \ldots, n\),

\[
x_t^* = \begin{cases} 
1, & r_t > a_t^\top p^* \\
0, & r_t < a_t^\top p^*
\end{cases}
\]

(few \(x_t^*\) may take non-integer value when \(r_t = a_t^\top p^*\)).

Online LP algorithms are based on learning \(p^*\) 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 $1 \leq t \leq \epsilon n$ and average allocation per bidder/buyer: $d = \frac{b}{n}$;
Simple Price-Learning Algorithm

We illustrate a simple Learning Algorithm:

- Set $x_t = 0$ for all $1 \leq t \leq \epsilon n$ and average allocation per bidder/buyer: $d = b/n$;
- Solve the $\epsilon$ portion of the problem

$$\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{\epsilon n} r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^{\epsilon n} a_{it} x_t \leq (\epsilon n) \cdot d_i \quad i = 1, \ldots, m \\
& \quad 0 \leq x_t \leq 1 \quad t = 1, \ldots, \epsilon n
\end{align*}$$

and get the optimal dual solution $\hat{p}$;
Simple Price-Learning Algorithm

We illustrate a simple Learning Algorithm:

- Set $x_t = 0$ for all $1 \leq t \leq \epsilon n$ and average allocation per bidder/buyer: $d = b/n$;
- Solve the $\epsilon$ portion of the problem

$$\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{\epsilon n} r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^{\epsilon n} a_{it} x_t \leq (\epsilon n) \cdot d_i \quad i = 1, \ldots, m \\
& \quad 0 \leq x_t \leq 1 \\
& \quad t = 1, \ldots, \epsilon n
\end{align*}$$

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

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

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) \((r_t, a_t)’s\) are i.i.d. from an unknown distribution
Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:
(a) \((r_t, a_t)\)'s are i.i.d. from an unknown distribution

Random Permutation (RP) Model:
(a') \((r_t, a_t)\)'s may be adversarially chosen but arrive in a random order (sample without replacement)

Early work assumes \(r_t; a_t\) (knapsack or one-sided market).

What are the necessary and sufficient conditions on the right-hand-side \(b\) to achieve \((1 + \epsilon)\)-competitive ratio of the expected total online reward over the optimal total offline reward \(OPT\) for all \((A; r)\)?

If the right-hand-side \(b = O(n)\), what is the best achievable sublinear gap or regret between the two?
Data/Model Assumptions for Analyses

**Stochastic Input (i.i.d) Model:**
(a) \((r_t, a_t)\)'s are i.i.d. from an unknown distribution

**Random Permutation (RP) Model:**
(a') \((r_t, a_t)\)'s may be adversarially chosen but arrive in a random order (sample without replacement)

**Both** assume boundedness:
(b) \(|r_t| \leq \bar{r} \) and \(\|a_t\|_\infty \leq \bar{a}\) for all \(t\)
(c) The right-hand-side \(b = n \cdot d(>0)\) in Regret Analysis.
Early work assumes \(r_t \geq 0, a_t \geq 0\) (knapsack or one-sited market).
Data/Model Assumptions for Analyses

**Stochastic Input (i.i.d) Model:**
(a) \((r_t, a_t)\)'s are i.i.d. from an unknown distribution

**Random Permutation (RP) Model:**
(a') \((r_t, a_t)\)'s may be adversarially chosen but arrive in a random order (sample without replacement)

**Both** assume boundedness:
(b) \(|r_t| \leq \bar{r}\) and \(\|a_t\|_\infty \leq \bar{a}\) for all \(t\)
(c) The right-hand-side \(b = n \cdot d(>0)\) in Regret Analysis.

Early work assumes \(r_t \geq 0, a_t \geq 0\) (knapsack or one-sited market).

- What are the necessary and sufficient conditions on the right-hand-side \(b\) to achieve \((1 - \epsilon)\)-competitive ratio of the expected total online reward over the optimal total offline reward \(OPT\) for all \((A, r)\)?
Data/Model Assumptions for Analyses

**Stochastic Input (i.i.d) Model:**
(a) \((r_t, a_t)\)'s are i.i.d. from an unknown distribution

**Random Permutation (RP) Model:**
(a') \((r_t, a_t)\)'s may be adversarially chosen but arrive in a random order (sample without replacement)

Both assume boundedness:
(b) \(|r_t| \leq \bar{r}\) and \(\|a_t\|_\infty \leq \bar{a}\) for all \(t\)
(c) The right-hand-side \(b = n \cdot d(> 0)\) in Regret Analysis.

Early work assumes \(r_t \geq 0, a_t \geq 0\) (knapsack or one-sited market).

- What are the necessary and sufficient conditions on the right-hand-side \(b\) to achieve \((1 - \epsilon)\)-competitive ratio of the expected total online reward over the optimal total offline reward \(OPT\) for all \((A, r)\)?
- If the right-hand-side \(b = O(n)\), what is the best achievable sublinear gap or regret between the two?
The conditions to design \((1 - \epsilon)\)-competitive online algorithms based on \(B = \min_i b_i\):

<table>
<thead>
<tr>
<th></th>
<th>Sufficient Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kleinberg (2005)</td>
<td>(B \geq \frac{1}{\epsilon^2}) for (m = 1)</td>
</tr>
<tr>
<td>Devanur et al (2009)</td>
<td>(\text{OPT} \geq \frac{m^2 \log n}{\epsilon^3})</td>
</tr>
<tr>
<td>Feldman et al (2010)</td>
<td>(B \geq \frac{m \log n}{\epsilon^3}) and (\text{OPT} \geq \frac{m \log n}{\epsilon})</td>
</tr>
<tr>
<td>Agrawal/Wang/Y (2010,14)</td>
<td>(B \geq \frac{m \log n}{\epsilon^2}) or (\text{OPT} \geq \frac{m^2 \log n}{\epsilon^2})</td>
</tr>
<tr>
<td>Molinaro/Ravi (2013)</td>
<td>(B \geq \frac{m^2 \log m}{\epsilon^2})</td>
</tr>
<tr>
<td>Kesselheim et al (2014)</td>
<td>(B \geq \frac{\log m}{\epsilon^2})</td>
</tr>
<tr>
<td>Gupta/Molinaro (2014)</td>
<td>(B \geq \frac{\log m}{\epsilon^2})</td>
</tr>
<tr>
<td>Agrawal/Devanur (2014)</td>
<td>(B \geq \frac{\log m}{\epsilon^2})</td>
</tr>
</tbody>
</table>
The conditions to design \((1 - \epsilon)\)-competitive online algorithms based on \(B = \min_i b_i\):

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

- Recent focuses are on dealing with:
  - two-sited markets/platforms, dual convergence, and regret analyses, and simple and fast algorithms,
  - online algorithm with interior-point LP solver,
  - extensions to bandit models and the Fisher market,
  - regret analysis with non i.i.d. input data.
Table of Contents

1. Online Linear Programming

2. Regret Analysis and 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

Let “offline” optimal solution be $x^*$ and “online” solution of $n$ orders be $x_n$, and

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

Let “offline” optimal solution be $x^*$ and “online” solution of $n$ orders be $x_n$, and

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

Then define

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

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

Let “offline” optimal solution be $x^*$ and “online” solution of $n$ orders be $x_n$, and

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

Then define

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

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 by early stopping such that the constraints are always satisfied at the end of the process.
Equivalent Form of the Dual Problem

Recall the dual problem

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

can be rewritten as

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

where $\left( \cdot \right)^+$ is the positive-part or ReLU function.
Equivalent Form of the Dual Problem

Recall the dual problem

\[
\min \ b^\top p + \sum_{t=1}^{n} s_t \quad \text{s.t. } s_t \geq r_t - a_t^\top p, \forall t; \quad p, s \geq 0
\]

can be rewritten as

\[
\min \ b^\top p + \sum_{t=1}^{n} \left( r_t - a_t^\top p \right)^+ \quad \text{s.t. } p \geq 0
\]

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

After normalizing the objective, it becomes

\[
\min_{p \geq 0} d^\top p + \frac{1}{n} \sum_{t=1}^{n} \left( r_t - a_t^\top p \right)^+
\]

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 Sample Dual $p_n^*$

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

Denote the $n$-sample SSA optimal solution by $p_n^*$. 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 $p^*$, there exists a constant $C$ such that

$$
\mathbb{E}\left\| p_n^* - p^* \right\|_2^2 \leq \frac{Cm \log \log n}{n}
$$

holds for all $n > m$. 
Theorem (Li & Y (2019, OR 2021))

Denote the $n$-sample SSA optimal solution by $p_n^*$. 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 $p^*$, there exists a constant $C$ such that

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

holds for all $n > m$.

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

$$
p_n^* \approx p^* + \frac{1}{\sqrt{n}} \cdot \text{Noise}
$$
Dual-Gradient Online Algorithm for Binary LP

LP-Solver Free Method:

1: Input: $d = b/n$ and initialize $p_1 = 0$
2: For $t = 1, 2, \ldots, n$
   
   $x_t = \begin{cases} 
   1, & \text{if } r_t > a_t^T p_t \\
   0, & \text{if } r_t \leq a_t^T p_t 
   \end{cases}$

3: Compute

   $\begin{cases} 
   p_{t+1} = p_t + \gamma_t (a_t x_t - d) \\
   p_{t+1} = p_{t+1}^+ 
   \end{cases}$

4: $x = (x_1, \ldots, x_n)$
Dual-Gradient Online Algorithm for Binary LP

LP-Solver Free Method:

1: Input: \( \mathbf{d} = \mathbf{b}/n \) and initialize \( \mathbf{p}_1 = 0 \)
2: For \( t = 1, 2, \ldots, n \)
   \[
   x_t = \begin{cases} 
   1, & \text{if } r_t > \mathbf{a}_t^\top \mathbf{p}_t \\
   0, & \text{if } r_t \leq \mathbf{a}_t^\top \mathbf{p}_t 
   \end{cases}
   \]
3: Compute
   \[
   \begin{cases} 
   \mathbf{p}_{t+1} = \mathbf{p}_t + \gamma_t (\mathbf{a}_t x_t - \mathbf{d}) \\
   \mathbf{p}_{t+1} = \mathbf{p}_{t+1}^+ 
   \end{cases}
   \]
4: \( \mathbf{x} = (x_1, \ldots, x_n) \)

Line 5 performs (projected) stochastic gradient descent in the dual, where step-size \( \gamma_t = \frac{1}{\sqrt{n}} \) or \( \gamma_t = \frac{1}{\sqrt{t}} \).
Dual-Gradient Online Algorithm for Binary LP

LP-Solver Free Method:

1: Input: \( \mathbf{d} = \mathbf{b}/n \) and initialize \( \mathbf{p}_1 = \mathbf{0} \)

2: For \( t = 1, 2, \ldots, n \)

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

3: Compute

\[
\begin{align*}
\mathbf{p}_{t+1} &= \mathbf{p}_t + \gamma_t (\mathbf{a}_t \mathbf{x}_t - \mathbf{d}) \\
\mathbf{p}_{t+1}^+ &= \mathbf{p}_{t+1} \\
\end{align*}
\]

4: \( \mathbf{x} = (x_1, \ldots, x_n) \)

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

This seems a classical online convex optimization algorithm, but the analysis is on \( \mathbf{r}^\top \mathbf{x} \) where \( \mathbf{x} \) is obtained onlinely.
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

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

under both the stochastic input and the random permutation models of two-sided data.

- $\tilde{O}$ 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 reward is in the range $O(mn)$.
- The algorithms runs in $nm$ times - the time to read in the data.
Adaptive Fast Online Algorithm for Binary LP

1. Initialize $b_1 = b$ and $p_1 = 0$

2. For $t = 1, 2, \ldots, n$
   
   $$x_t = \begin{cases} 
   1, & \text{if } r_t > a_t^\top p_t \\
   0, & \text{if } r_t \leq a_t^\top p_t
   \end{cases}$$

3. Compute
   
   $$p_{t+1} = p_t + \gamma_t \left( a_t x_t - \frac{1}{n-t+1} b_t \right)$$
   $$p_{t+1} = \max(p_{t+1}, 0)$$

4. Update remaining inventory: $b_{t+1} = b_t - a_t x_t$.

5. Return $x = (x_1, \ldots, x_n)$
Adaptive Fast Online Algorithm for Binary LP

1: Initialize $b_1 = b$ and $p_1 = 0$
2: For $t = 1, 2, \ldots, n$

$$x_t = \begin{cases} 
1, & \text{if } r_t > a_t^\top p_t \\
0, & \text{if } r_t \leq a_t^\top p_t
\end{cases}$$

3: Compute

$$p_{t+1} = p_t + \gamma_t \left( a_t x_t - \frac{1}{n-t+1} b_t \right)$$
$$p_{t+1} = p_{t+1} \lor 0$$

4: Update remaining inventory: $b_{t+1} = b_t - a_t x_t$.
5: Return $x = (x_1, \ldots, x_n)$

**Only Difference:** The average allocation vector $b/n$ in Step 3 is adaptively replaced 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 Algorithm as a Pre-Solver for the Offline LP Solver Development

More precisely, the fast online LP solution can be interpreted as a presolver and establish a “score” of how likely a variable is to be optimal basic (nonzero).

We run online algorithm to obtain $\hat{x}$, set a threshold $\varepsilon$ and select the columns in $\mathbb{I}_{\{\hat{x} > \varepsilon\}}$ in the column-generation scheme. For a benchmark LP problem in the Mittelmann’s Simplex Benchmark, this reduces solution time from hundreds to 8 seconds (or 3 seconds by IPM).

This technique has been adopted in the emerging LP solver COPT - one of the state of art LP solvers nowadays.
Fast Algorithm as a Pre-Solver for the Offline LP Solver Development

More precisely, the fast online LP solution can be interpreted as a presolver and establish a “score” of how likely a variable is to be optimal basic (nonzero).

We run online algorithm to obtain $\hat{x}$, set a threshold $\varepsilon$ and select the columns in $\mathbb{I}_{\{x>\varepsilon\}}$ in the column-generation scheme. For a benchmark LP problem in the Mittelmann’s Simplex Benchmark, this reduces solution time from hundreds to 8 seconds (or 3 seconds by IPM).

This technique has been adopted in the emerging LP solver COPT - one of the state of art LP solvers nowadays.

Are other types of data learn-able?
Regenerative Data of Different Scales

Figure: 1) Simulated Regenerative Data; 2) Soybean price (years); 3) Coffee Price (years); 4) TSLA (15 seconds)
Theorem (Regenerative Dual Convergence)

Suppose $a_t$ follows an i.i.d process and $r_j$ follows a regenerative process with bounded regenerative time, and under the same boundedness and non-degeneracy assumptions as in the i.i.d Dual Convergence Theorem, there exists a constant $C$ such that

$$
\mathbb{E} \left[ \| p^*_n - p^* \|_2^2 \right] \leq \frac{Cm \log m \log \log n}{n}
$$

holds for all $n \geq \max\{m, 3\}$, $m \geq 2$. Additionally,

$$
\mathbb{E} \left[ \| p^*_n - p^* \|_2 \right] \leq C \sqrt{\frac{m \log m \log \log n}{n}}
$$

.
Regrets for Online Algorithms

Since the regenerative data has the same dual convergence rate, we can show the regrets are as well bounded by the same order:

**Theorem (Regenerative Regret by Using Optimal Stochastic Prices)**

*With the online policy $\pi_1$ specified by Algorithm 1 with regenerative data,*

$$\Delta_n \leq O(\sqrt{n})$$

**Theorem (Regenerative Regret by LP Learning)**

*With the online policy $\pi_2$ specified by Algorithm 2 with regenerative data,*

$$\Delta_n \leq O(\sqrt{n \log n})$$
Table of Contents

1. Online Linear Programming

2. Regret Analysis and 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 “Solution-Uniqueness” Assumption in Online LP Algorithm

A Common Assumption: the learning target, solution of the offline LP problem, is unique or non-generate.
A “Solution-Uniqueness” Assumption in Online LP Algorithm

A Common Assumption: the learning target, solution of the offline LP problem, is *unique* or *non-generate*. Let $T$ bidders (changed from $n$ as in the literature) bidders have a finite types, $i = 1, \ldots, K$, with $\mathbb{P}((r_t, a_t) = (\mu_i, c_i)) = p_i$ (unknown to the decision maker). Then, the offline problem reduces to:

$$\max \sum_{i=1}^{K} p_i \mu_i y_i \quad \text{s.t.} \quad \sum_{i=1}^{K} p_i c_i y_i \leq b / T, \quad y_i \in [0, 1]$$

where $y_i$ is the acceptance rate/probability for customer type $i$ (some are zeros or “nonbasic”!)
A “Solution-Uniqueness” Assumption in Online LP Algorithm

A Common Assumption: the learning target, solution of the offline LP problem, is unique or non-generate.

Let \( T \) bidders (changed from \( n \) as in the literature) bidders have a finite types, \( i = 1, \ldots, K \), with \( \mathbb{P}((r_t, a_t) = (\mu_i, c_i)) = p_i \) (unknown to the decision maker). Then, the offline problem reduces to:

\[
\max \sum_{i=1}^K p_i \mu_i y_i \quad \text{s.t.} \quad \sum_{i=1}^K p_i c_i y_i \leq b/T, \quad y_i \in [0,1]
\]

where \( y_i \) is the acceptance rate/probability for customer type \( i \) (some are zeros or “nonbasic”!)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Regret Bound</th>
<th>Key Assumption(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vera et al. (2019)</td>
<td>Hindsight</td>
<td>Bounded</td>
</tr>
<tr>
<td>Bumpensanti and Wang (2020)</td>
<td>Hindsight</td>
<td>Bounded</td>
</tr>
<tr>
<td>Asadpour et al. (2019)</td>
<td>Full flex.</td>
<td>Bounded</td>
</tr>
</tbody>
</table>
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 $\hat{p}_j$

$$\max \sum_{i=1}^K \hat{p}_i \mu_i y_i \quad \text{s.t.} \quad \sum_{i=1}^K \hat{p}_i c_i y_i \leq b/T, \quad y_i \in [0, 1],$$

since the sample and offline LP may be degenerate or with multiple optimal solutions - a common property for real-life LP problems.
The key in Chen et al. (2021) paper is to use the interior-point algorithm for solving the sample LPs with sample proportion $\hat{p}_j$

$$\max \sum_{i=1}^{K} \hat{p}_i \mu_i y_i \quad \text{s.t.} \quad \sum_{i=1}^{K} \hat{p}_i c_i y_i \leq b / T, \quad y_i \in [0, 1],$$

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 tend to accept mainly the first half (or the second half of the orders), which is unfair or unideal such as Adwords application.
Time Fairness: The algorithm may tend 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 tend 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 maintaining the efficiency in objective value, all individuals/groups get a fairer allocation share?
Fairness Desiderata: Time and Individual

**Time Fairness:** The algorithm may tend 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 $y^*$, the fair offline optimal solution of the LP problem

$$\max \sum_{i=1}^{K} p_i \mu_i y_i, \quad \text{s.t.} \quad \sum_{i=1}^{K} p_i c_i y_i \leq b/T, \quad y_i \in [0, 1]$$

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

$$\max \sum_{i=1}^{K} p_i \mu_i y_i, \quad \text{s.t.} \quad \sum_{i=1}^{K} p_i c_i y_i \leq b/T, \quad y_i \in [0, 1]$$

as the \textit{analytical center} of the optimal solution set, which represents an “average” of all the corner optimal solutions.

Let $y_t$ be allocation solution at time $t$ which encodes the accepting rates/probabilities under algorithm $\pi$. Then we define the \textit{cumulative unfairness} of the online algorithm $\pi$ as

$$UF_T(\pi) = \mathbb{E} \left[ \sum_{t=1}^{T} \| y_t - y^* \|_2^2 \right].$$
Fairer Solution for the Offline Problem

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

$$\max \sum_{i=1}^{K} p_i \mu_i y_i, \quad \text{s.t.} \quad \sum_{i=1}^{K} p_i c_i y_i \leq \mathbf{b}/T, \quad y_i \in [0, 1]$$

as the \textit{analytical center} of the optimal solution set, which represents an “average” of all the corner optimal solutions.

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

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

This definition is consistent with the definition of so-called \textit{fair} classifiers/regressors in machine learning.
Our Result

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

\[ UF_T(\pi) = O(\log T) \text{ and } \text{Reg}_T(\pi) = \text{Bounded w.r.t } T \]
Our Result

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

$$\text{UF}_T(\pi) = O(\log T) \text{ and } \text{Reg}_T(\pi) = \text{Bounded w.r.t } T$$

Key ideas in algorithm design:
- At each time $t$, we use interior-point method to obtain the analytic-center solution $y_t$ of sampled LPs, and it is necessary to achieve the performance under non-uniqueness assumption while maintain fairness.
- We also adaptively 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.
Our Result

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

$$\text{UF}_T(\pi) = O(\log T) \text{ and } \text{Reg}_T(\pi) = \text{Bounded w.r.t } T$$

Key ideas in algorithm design:

- At each time $t$, we use interior-point method to obtain the analytic-center solution $y_t$ of sampled LPs, and it is necessary to achieve the performance under non-uniqueness assumption while maintain fairness.

- We also adaptively 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.

An advantage of interior-point method over simplex method!
# Table of Contents

1. Online Linear Programming

2. Regret Analysis and 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 $t$, the decision maker decides an arm (/customer/order) among $K$ arms to play/sell and then observe $(\hat{r}_t, \hat{c}_t)$.
Bandits with Knapsacks

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

Horizon: $T$ time periods (known a priori)
Bandits with Knapsacks

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

Horizon: $T$ time periods (known a priori)

Bandits: $K$ arms, where each arm $i$ with an unknown mean reward $\mu_i$. 
Bandits with Knapsacks

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

**Horizon:** $T$ time periods (known a priori)

**Bandits:** $K$ arms, where each arm $i$ with an unknown mean reward $\mu_i$.

**Knapsacks:** $m$ types of resources with a known total resource capacity $b \in \mathbb{R}^m$, and the pull of each arm requires an unknown resource bundle.
Bandits with Knapsacks

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

**Horizon:** $T$ time periods (known a priori)

**Bandits:** $K$ arms, where each arm $i$ with an unknown mean reward $\mu_i$.

**Knapsacks:** $m$ types of resources with a known total resource capacity $b \in \mathbb{R}^m$, and the pull of each arm requires an unknown resource bundle.

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

$$\mathbb{E}[\hat{r}_t | i] = \mu_i, \quad \mathbb{E}[\hat{c}_t | i] = c_i.$$
Bandits with Knapsacks

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

Horizon: $T$ time periods (known a priori)

Bandits: $K$ arms, where each arm $i$ with an unknown mean reward $\mu_i$.

Knapsacks: $m$ types of resources with a known total resource capacity $b \in \mathbb{R}^m$, and the pull of each arm requires an unknown resource bundle.

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

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

Goal: Select a subset of winning/optimal arms to pull in order 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 $\mu = (\mu_1, ..., \mu_K)$ and mean resource-cost $(c_1, ..., c_K)$ of arms, consider the following deterministic offline LP,

$$
\max_x \sum_{i=1}^{K} \mu_i x_i \quad \text{s.t.} \quad \sum_{i=1}^{K} c_i x_i \leq b, \ x_i \geq 0, \ i \in [k]
$$

Here $x_i$ represents the optimal times of playing $i$-th arm if everything is deterministic and known – only $m$ of them positive (basic).
Offline Linear Program (LP) and Regret

With mean reward \( \mu = (\mu_1, \ldots, \mu_K) \) and mean resource-cost \((c_1, \ldots, c_K)\) of arms, consider the following deterministic offline LP,

\[
\max_x \sum_{i=1}^{K} \mu_i x_i \quad \text{s.t.} \quad \sum_{i=1}^{K} c_i x_i \leq b, \ x_i \geq 0, \ i \in [k]
\]

Here \( x_i \) represents the optimal times of playing \( i \)-th arm if everything is deterministic and known – only \( m \) of them positive (basic).

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

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

where \( \mathcal{P} \) encapsulates the parameters related to the underlying data distribution.
## Literature and Our Result

<table>
<thead>
<tr>
<th>$P$-Independent</th>
<th>Paper</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$-Independent</td>
<td>Badanidiyuru et. al. (13)</td>
<td>$O(\text{poly}(m, k) \cdot \sqrt{T})$</td>
</tr>
<tr>
<td>$P$-Independent</td>
<td>Agrawal and Devanur (14)</td>
<td>$O(\text{poly}(m, k) \cdot \sqrt{T})$</td>
</tr>
<tr>
<td>$P$-Dependent</td>
<td>Flajolet and Jaillet (15)</td>
<td>$\tilde{O}(2^{m+k} \log T)$</td>
</tr>
<tr>
<td>$P$-Dependent</td>
<td>Sankararaman and Slivkins (20)</td>
<td>$\tilde{O}(k \log T)$ for $m = 1$</td>
</tr>
<tr>
<td>$P$-Dependent</td>
<td>Li, Sun &amp; Y (21)</td>
<td>$\tilde{O}(m^4 + k \log T)$</td>
</tr>
</tbody>
</table>

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

**Primal**: \[ \begin{align*}
\text{max} & \quad \mu^\top x \\
\text{s.t.} & \quad Cx \leq b, \; x \geq 0
\end{align*} \]

**Dual**: \[ \begin{align*}
\text{min} & \quad b^\top y \\
\text{s.t.} & \quad C^\top y \geq \mu, \; y \geq 0
\end{align*} \]

Denote \( x^* \in R^K \) and \( y^* \in R^m \) as optimal solutions.
Define reduced cost (profit) for \( i \)-th arm \( \Delta_i := c_i^\top y^* - \mu_i \) and the “nonbasic” 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:

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

- \( b^{(t)} \): remaining resources 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 $\Delta_i$. 

Task II: Make sure no valuable resources $b_j(\tau)$ left unused, which corresponds to the second component in the regret bound $\mathbb{E}[b_j(\tau)]$.(Recall is the time that one of the resources is exhausted.)

Task II is often overlooked in the existing BwK literature.
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 $\Delta_i$.

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

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

Recall $\tau$ 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 $\Delta_i$.

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

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

Recall $\tau$ 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 $i$:

$$OPT_i := \max \mu^\top x$$

s.t. $Cx \leq b$, $x_i = 0$, $x \geq 0$.

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 $i$:

  \[
  OPT_i := \max \mu^\top x \\
  \text{s.t.} \quad Cx \leq b, \quad x_i = 0, \quad x \geq 0.
  \]

  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 $t' = O\left(\frac{k \log T}{\sigma^2 \delta^2}\right)$ 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.
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:

- \(\sigma\) 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:

- \( \sigma \) is the minimum singular value of the sub-matrix of the constraint matrix \( C \) that corresponds to the optimal basis.
- \( \delta \) measures the difficulty of identifying optimal basic variables:
  \[ \min \{ \min \{x^*_i | x^*_i > 0\}, \min \{OPT - OPT_i | x^*_i > 0\}, \min \{\Delta_i | x^*_i = 0\} \}. \]
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:

- \( \sigma \) is the minimum singular value of the sub-matrix of the constraint matrix \( C \) that corresponds to the optimal basis.
- \( \delta \) measures the difficulty of identifying optimal basic variables:
  \[ \min \{ \min \{ x_i^* | x_i^* > 0 \}, \min \{ OPT - OPT_i | x_i^* > 0 \}, \min \{ \Delta_i | x_i^* = 0 \} \} \]

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. Regret Analysis and Fast Algorithms for (Binary) Online Linear Programming
3. A Fairer Online Interior-Point LP Algorithm
4. Online Bandits with Knapsacks
5. Online Fisher Markets
The Fisher Social Optimization Problem

\[
\max_{x_i}'s \quad \sum_{i \in B} w_i \log(u_i^T x_i)
\]

s.t. \[\sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G, \quad x_{ij} \geq 0, \quad \forall i, j,\]

\(u_i\): linear utility coefficients of buyer \(i\), \(c_j\): capacity of good \(j\).
The Fisher Social Optimization Problem

\[ \max_{x_i's} \quad \sum_{i \in B} w_i \log(u_i^T x_i) \]

s.t. \[ \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G, \quad x_{ij} \geq 0, \quad \forall i, j, \]

\( u_i \): linear utility coefficients of buyer \( i \), \( c_j \): capacity of good \( j \).

**Theorem (Eisenberg and Gale (1959))**

*Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector to clear the market.*
The Fisher Social Optimization Problem

\[ \max_{x_i's} \sum_{i \in B} w_i \log(u_i^T x_i) \]

s.t. \( \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G, \quad x_{ij} \geq 0, \quad \forall i, j, \)

\( u_i \): linear utility coefficients of buyer \( i \), \( c_j \): capacity of good \( j \).

Theorem (Eisenberg and Gale (1959))

Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector to clear the market.

Now, consider the online setting: \( n \) buyers/agents arrive Online and an irrevocable allocation-bundle \( x_i \) has to be made on time (Agrawal/Devanur 2014; Lu et al. 2020).
The Fisher Social Optimization Problem

\[
\max_{x_i's} \quad \sum_{i \in B} w_i \log(u_i^T x_i)
\]
\[
\text{s.t.} \quad \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G, \quad x_{ij} \geq 0, \quad \forall i, j,
\]

\(u_i\): linear utility coefficients of buyer \(i\), \(c_j\): capacity of good \(j\).

Theorem (Eisenberg and Gale (1959))

Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector to clear the market.

Now, consider the online setting: \(n\) buyers/agents arrive Online and an irrevocable allocation-bundle \(x_i\) has to be made on time (Agrawal/Devanur 2014; Lu et al. 2020).

Questions: Could the algorithm be implemented while protecting privacy by a price-posting mechanism? How much would the aggregated social welfare be deteriorated from the offline setting? May the market be cleared?
Let “offline” optimal solution be $\mathbf{x}_i^*$ and “online” solution be $\mathbf{x}_i$, and

$$R_n^* = \sum_{i=1}^{n} w_i \log(u_i^T \mathbf{x}_i^*), \quad R_n = \sum_{i=1}^{n} w_i \log(u_i^T \mathbf{x}_i)$$
Regret Analysis and Model

Let “offline” optimal solution be \( x^*_i \) and “online” solution be \( x_i \), and

\[
R^*_n = \sum_{i=1}^{n} w_i \log(u^T_i x^*_i), \quad R_n = \sum_{i=1}^{n} w_i \log(u^T_i x_i)
\]

Then define

\[
\Delta_n = \sup \mathbb{E} [R^*_n - R_n], \quad \nu(x) = \sup \mathbb{E} \left[ \| (Ax - b)^+ \|_2 \right]
\]

where the expectation is taken with respect to i.i.d distribution, and the \text{sup} operator is over all permissible distributions and admissible data.
Regret Analysis and Model

Let “offline” optimal solution be \( x^*_i \) and “online” solution be \( x_i \), and

\[
R^*_n = \sum_{i=1}^{n} w_i \log(u_i^T x^*_i), \quad R_n = \sum_{i=1}^{n} w_i \log(u_i^T x_i)
\]

Then define

\[
\Delta_n = \sup \mathbb{E} [R^*_n - R_n], \quad \nu(x) = \sup \mathbb{E} \left[ \| (Ax - b)^+ \|_2 \right]
\]

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

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

\[
\frac{\left( \prod_i (u_i^T x^*_i)^{w_i} \right)^{1/n}}{\left( \prod_i (u_i^T x_i)^{w_i} \right)^{1/n}} \leq e^{o(n)/n}.
\]
Online Fisher Markets: Price-Posting Mechanism

Each agent $i$, with budget $w_i$, purchases an optimal bundle $x^t_i$ given price $p^t$

How to setup $p^t$ for each good before buyer $t$ comes so that the social welfare is maximized and capacity constraint violation is minimized for total $n$ buyers?
There is an adaptive price-policy (path-dependent price vector) such that the market is cleared and the expected optimal social value

\[ n \log(2) - 1 \leq \mathbb{E}[R_n] = \mathbb{E}[R^*_n] \leq n \log(2). \]

However, for any static pricing-policy, even using the expected optimal equilibrium price-vector, either the expected regret or constraint violation is at least \( \Omega \sqrt{n} \).
One may apply a similar primal price-learning algorithm, that is, solve the aggregated social problem based on arrived \( \epsilon \) portion of buyers:

\[
\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{\epsilon n} w_t \log(u_t^T x_t) \\
\text{subject to} & \quad \sum_{t=1}^{\epsilon n} x_t \leq \epsilon c_j, \quad j = 1, \ldots, m \\
& \quad 0 \leq x_t.
\end{align*}
\]

One can set an initial positive price vector \( p^1 \) and determine allocation \( x_t \) as the optimal solution for the individual maximization problem under price vector \( p^t \).
Simple Price-Learning Algorithm

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

$$\text{maximize}_x \sum_{t=1}^{\epsilon n} w_t \log(u_t^T x_t)$$
subject to
$$\sum_{t=1}^{\epsilon n} x_t \leq \epsilon c_j, \quad j = 1, \ldots, m$$
$$0 \leq x_t.$$ 

One can set an initial positive price vector $p^1$ and determine allocation $x_t$ as the optimal solution for the individual maximization problem under price vector $p^t$.

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 $\epsilon$ portion of buyers:

$$\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{\epsilon n} w_t \log(u_t^T x_t) \\
\text{subject to} & \quad \sum_{t=1}^{\epsilon n} x_t \leq \epsilon c_j, \quad j = 1, \ldots, m \\
& \quad 0 \leq x_t.
\end{align*}$$

One can set an initial positive price vector $p^1$ and determine allocation $x_t$ as the optimal solution for the individual maximization problem under price vector $p^t$.

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

Could the prices be updated in a privacy-preserving manner?
A Privacy-Preserving Algorithm

Consider the dual 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

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

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 $p^1 = e$, and for $t = 1, 2, \ldots, n$
2. Let $x_t$ be the individual optimal bundle solution under price vector $p^t$.
3. Update prices
   \[ p_{t+1} = p_t - \gamma_t (d - x_t) \]
   \[ p_{t+1}^+ = p_{t+1} \]
4. $x = (x_1, \ldots, x_n)$

Again, line 3 performs (projected) stochastic gradient step.
1: Initialize $p^1 = e$, and for $t = 1, 2, \ldots, n$
2: Let $x_t$ be the individual optimal bundle solution under price vector $p^t$.
3: Update prices
   \[ p_{t+1} = p_t - \gamma_t (d - x_t) \]
   \[ p_{t+1}^+ = p_{t+1} \]
4: $x = (x_1, \ldots, x_n)$

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 $v(x) \leq O(\sqrt{n})$, where $n$ is the number of arriving buyers.
Takeaways and Open Problems

- **Learning-while-doing (taking actions)** is common in today’s decision making.
- The Off-line and On-line Regret measures the learning efficiency.
- Could more non-stationary data be learned with sub-linear regret?
- Could learning/decision be based on past data together with future prediction?
- Overall, **Linear Programming** continues to play a big role in online learning and decisioning.
Takeaways and Open Problems

- Learning-while-doing (taking actions) is common in today’s decision making
- The Off-line and On-line Regret measures the learning efficiency
- Could more non-stationary data be learned with sub-linear regret?
- Could learning/decision be based on past data together with future prediction?
- Overall, Linear Programming continues to play a big role in online learning and decisioning.

Long Live Linear Programming!