<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Online Linear Programming</td>
</tr>
<tr>
<td>2</td>
<td>Fast Algorithms for (Binary) Online Linear Programming</td>
</tr>
<tr>
<td>3</td>
<td>A Fairer Online Interior-Point LP Algorithm</td>
</tr>
<tr>
<td>4</td>
<td>Online Bandits with Knapsacks</td>
</tr>
<tr>
<td>5</td>
<td>Online Fisher Markets</td>
</tr>
</tbody>
</table>
Offline and Online Linear Programming

\[
\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{n} r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^{n} a_t x_t \leq b, \\
& \quad x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \ \forall t = 1, \ldots, n.
\end{align*}
\]
Offline and Online Linear Programming

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

- \(r_t\): 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 and Online Linear Programming

\[
\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^{n} r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^{n} a_t x_t \leq b, \\
& \quad x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, \ldots, n.
\end{align*}
\]

- \(r_t\): 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 and Online Linear Programming

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.
Offline and Online Linear Programming

\[
\begin{align*}
\text{maximize}_{x} & \quad \sum_{t=1}^{n} r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^{n} a_t x_t \leq b, \\
& \quad x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \ \forall t = 1, \ldots, n.
\end{align*}
\]

- \(r_t\): 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 \textit{sequentially}.
- an \textit{irrevocable decision} must be made as soon as an order arrives (without knowing the future data).
Offline and Online Linear Programming

\[
\begin{align*}
\text{maximize}_x & \quad \sum_{t=1}^n r_t x_t \\
\text{subject to} & \quad \sum_{t=1}^n a_t x_t \leq b, \\
& \quad x_t \in \{0, 1\} \ (0 \leq x_t \leq 1), \quad \forall t = 1, ..., n.
\end{align*}
\]

- \( r_t \): 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.
A Toy Example

Consider an auction problem:

<table>
<thead>
<tr>
<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>
</tr>
<tr>
<td>Decision</td>
<td>(x_1)</td>
<td>(x_2)</td>
<td>...</td>
</tr>
<tr>
<td>Pants</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>Shoes</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>T-shirts</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>Jackets</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>Hats</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

where the decision for each customer/bidder is “accept” \((x_t = 1)\) or “reject” \((x_t = 0)\)
The problem would be easy if there are “ideal prices”:

<table>
<thead>
<tr>
<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>Bid ($r_t$)</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>

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

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

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

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

$$\max \quad r^\top x$$
\[ P : \text{s.t.} \quad Ax \leq b \]
\[ 0 \leq x \leq e \]

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

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

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

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

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

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

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

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

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

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

Most 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 $1 \leq t \leq \epsilon n$;
Simple Price-Learning Algorithm

We illustrate a simple Learning Algorithm:

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

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

and get the optimal dual solution $\hat{p}$;
We illustrate a simple Learning Algorithm:

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

$$\text{maximize}_x \sum_{t=1}^{\epsilon n} r_t x_t$$
$$\text{subject to } \sum_{t=1}^{\epsilon n} a_{it} x_t \leq \frac{\epsilon n}{n} b_i \quad i = 1, \ldots, m$$
$$0 \leq x_t \leq 1 \quad t = 1, \ldots, \epsilon n$$

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.
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)

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

If the right-hand-side \(b\) (such as \(b = O(n^d)\)) what is the best achievable 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 \(b = n \cdot d(>0)\).

All early works also assume \(r_t \geq 0, a_t \geq 0\) (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 \(b = n \cdot d(>0)\).

All early works also assume \(r_t \geq 0, a_t \geq 0\) (one-sited market).

What are the **necessary and sufficient** assumptions on the right-hand-side \(b\) to achieve \((1 - \epsilon)\)-competitive ratio of the expected online reward over the optimal offline reward?
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 \(b = n \cdot d(> 0)\).

All early works also assume \(r_t \geq 0, a_t \geq 0\) (one-sited market).

- What are the **necessary and sufficient** assumptions on the right-hand-side \(b\) to achieve \((1 - \epsilon)\)-competitive ratio of the expected online reward over the optimal offline reward?
- If the right-hand-side \(b\) (such as \(b = O(n)\)), what is the best achievable gap or regret between the two?
The journey to design \((1 - \epsilon)\)-competitive online algorithms against benchmark OPT—Optimal Offline Objective Value where \(B = \min_{i} b_i\):

<table>
<thead>
<tr>
<th>Authors</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>(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 (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 (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 journey to design \((1 - \epsilon)\)-competitive online algorithms against benchmark \(OPT\) - Optimal Offline Objective Value where \(B = \min_i b_i\):

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

- 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 decision variables are required to be the same at each time period.
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 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 $x^*$ 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 and Model

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

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

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 $x^*$ 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

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

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.

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

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

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.
Equivalent Form of the Dual Problem

Recall the dual problem

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

can be rewritten as

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

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 $p^*_n$

**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

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

holds for all $n > m$. 
Convergence of $p_n^*$

**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

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

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

1: Input: $d = b/n$
2: Initialize $p_1 = 0$
3: For $t = 1, 2, ..., n$
4:
   $x_t = \begin{cases} 
   1, & \text{if } r_t > a_t^T p_t \\
   0, & \text{if } r_t \leq a_t^T p_t 
   \end{cases}$
5: Compute
   
   \[ p_{t+1} = p_t + \gamma_t (a_t x_t - d) \]
   \[ p_{t+1} = p_{t+1} \lor 0 \]
6: $x = (x_1, ..., x_n)$
Dual-Gradient Online Algorithm for Binary LP

1: Input: \( \mathbf{d} = \mathbf{b}/n \)
2: Initialize \( \mathbf{p}_1 = 0 \)
3: For \( t = 1, 2, \ldots, n \)
4: 

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

5: Compute

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

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

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

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

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

1: Initialize $b_1 = b$ and $p_1 = 0$
2: For $t = 1, 2, \ldots, n$
3: \[ x_t = \begin{cases} 1, & \text{if } r_t > a_t^\top p_t \\ 0, & \text{if } r_t \leq a_t^\top p_t \end{cases} \]
4: Compute
\[ p_{t+1} = p_t + \gamma_t \left( a_t x_t - \frac{1}{n-t+1} b_t \right) \]
\[ p_{t+1} = p_{t+1} \lor 0 \]
5: Update remaining inventory: $b_{t+1} = b_t - a_t x_t$.
6: 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 $t = 1, 2, ..., n$
3: 
   \[ x_t = \begin{cases} 
   1, & \text{if } r_t > a_t^T p_t \\
   0, & \text{if } r_t \leq a_t^T p_t 
   \end{cases} \]
4: Compute
   \[ p_{t+1} = p_t + \gamma_t (a_t x_t - \frac{1}{n-t+1} b_t) \]
   \[ p_{t+1} = p_{t+1} \lor 0 \]
5: Update remaining inventory: $b_{t+1} = b_t - a_t x_t$.
6: Return $x = (x_1, ..., x_n)$

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
A crucial assumption is that the right-hand-side $b = nd$ 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

$$\begin{align*}
\max & \sum_{t=1}^{n} \sum_{h=1}^{k} r_{th} x_{th} \\
\text{s.t.} & \sum_{t=1}^{n} \sum_{h=1}^{k} a_{th} x_{th} \leq k \mathbf{b}, \quad 0 \leq x_{t} \leq 1, \quad t = 1, \ldots, n.
\end{align*}$$

Algorithm: Solve the new LP with Simple Online Algorithm and use $x_{t} = \frac{1}{k} (x_{t1} + \ldots + x_{tk})$ as the solution to the original LP.
A crucial assumption is that the right-hand-side $b = nd$ 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

$$\begin{align*}
&\text{max} \quad \sum_{t=1}^{n} \sum_{h=1}^{k} r_t x_{th} \\
&\text{s.t.} \quad \sum_{t=1}^{n} \sum_{h=1}^{k} a_t x_{th} \leq k b, \quad 0 \leq x_t \leq 1, \quad t = 1, \ldots, n.
\end{align*}$$

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

<table>
<thead>
<tr>
<th></th>
<th>V.R. Alg.</th>
<th>Gurobi</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 5, n = 500, k = 50$</td>
<td>0.000</td>
<td>0.211</td>
</tr>
<tr>
<td></td>
<td>88.2%</td>
<td>95.3%</td>
</tr>
<tr>
<td>$m = 5, n = 500, k = 1000$</td>
<td>0.007</td>
<td>0.211</td>
</tr>
<tr>
<td></td>
<td>89.2%</td>
<td>95.3%</td>
</tr>
<tr>
<td>$m = 8, n = 10^3, k = 50$</td>
<td>0.004</td>
<td>3.800</td>
</tr>
<tr>
<td></td>
<td>89.9%</td>
<td>99.0%</td>
</tr>
<tr>
<td>$m = 8, n = 10^3, k = 1000$</td>
<td>0.077</td>
<td>3.800</td>
</tr>
<tr>
<td></td>
<td>95.6%</td>
<td>99.0%</td>
</tr>
<tr>
<td>$m = 64, n = 10^4, k = 50$</td>
<td>0.013</td>
<td>&gt; 60</td>
</tr>
<tr>
<td></td>
<td>90.3%</td>
<td>98.7%</td>
</tr>
<tr>
<td>$m = 64, n = 10^4, k = 1000$</td>
<td>0.223</td>
<td>&gt; 60</td>
</tr>
<tr>
<td></td>
<td>96.4%</td>
<td>98.7%</td>
</tr>
</tbody>
</table>
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.
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 $n \gg m$:

- 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!
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 $n \gg m$:

- 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.
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 $\hat{x}$, set a threshold $\varepsilon$ and select the columns in $I_{\{\hat{x} > \varepsilon\}}$. 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.
Recall the online LP formulation (changing $n$ to $T$ as in the literature)

$$\max \sum_{t=1}^{T} r_t x_t \quad \text{s.t.} \quad \sum_{t=1}^{T} a_t x_t \leq b, \quad x_t \in [0, 1]$$
Recall the online LP formulation (changing $n$ to $T$ as in the literature)

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

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

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

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

A "Fairer" Online LP Algorithm
A “Fairer” Online LP Algorithm

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

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

A finite-type assumption: \( \mathbb{P}((r_t, a_t) = (\mu_j, c_j)) = p_j \) (unknown to the decision maker) for \( j = 1, \ldots, J \). The offline problem with the knowledge:

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

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

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

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

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 $\hat{p}_j$

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

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 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.
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.
**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?
We define $y^*$, the **fair** offline optimal solution of the LP problem

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

as the **analytical center** of the optimal solution set, which represents an “average” of all the corner optimal solutions.
We define $y^*$, the fair offline optimal solution of the LP problem
\[
\max \sum_{j=1}^J p_j \mu_j y_j, \quad \text{s.t.} \quad \sum_{j=1}^J p_j c_j y_j \leq b / T, \quad y_j \in [0, 1]
\]
as the analytical center of the optimal solution set, which represents an “average” of all the corner optimal solutions.

Let $y_t$ be allocation rule at time $t$ which encodes the accepting probabilities under algorithm $\pi$. Then we define the 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 $y^*$, the fair offline optimal solution of the LP problem

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

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

Let $y_t$ be allocation rule at time $t$ which encodes the accepting probabilities under algorithm $\pi$. Then we define the cumulative unfairness of the online algorithm $\pi$ as

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

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

\[ UF_T(\pi) = O(\log T) \]

\[ Reg_T(\pi) = \text{Bounded w.r.t } T \]
Our Result

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

\[ \text{UF}_T(\pi) = O(\log T) \]

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

Key ideas in algorithm design:

- At each time \( t \), we use \textit{interior-point method} to obtain the sample analytic-center solution \( y_t \), 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.
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 $(\hat{r}_t, \hat{c}_t)$.
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 \((\hat{r}_t, \hat{c}_t)\).

**Horizon:** \( T \) time periods (\( T \) known a priori)
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 $(\hat{r}_t, \hat{c}_t)$.

**Horizon:** $T$ time periods ($T$ 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 $t$, decide a customer/order/arm among $k$ arms to sell/play and then observe $(\hat{r}_t, \hat{c}_t)$.

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

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

**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 $(\hat{r}_t, \hat{c}_t)$.

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

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

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

At each time $t \in [T]$, an arm $i$ is selected to pull. The realized reward $\hat{r}_t$ and resources cost $\hat{c}_t$ 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 $t$, decide a customer/order/arm among $k$ arms to sell/play and then observe $(\hat{r}_t, \hat{c}_t)$.

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

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

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

At each time $t \in [T]$, an arm $i$ is selected to pull. The realized reward $\hat{r}_t$ and resources cost $\hat{c}_t$ 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 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 $\mu = (\mu_1, \ldots, \mu_k)$ 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 fractional number of playing $i$-th arm if everything is deterministic and known.
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 fractional number of playing $i$-th arm if everything is deterministic and known.

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></th>
<th>Paper</th>
<th>Result</th>
</tr>
</thead>
</table>
| $\mathcal{P}$-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 $m = 1$  
$\tilde{O}(m^4 + k \log T)$ |

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

\textit{Primal}: \quad \max \quad \mu^\top x \quad \text{s.t.} \quad Cx \leq b, \; x \geq 0

\textit{Dual}: \quad \min \quad b^\top y \quad \text{s.t.} \quad C^\top y \geq \mu, \; y \geq 0

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 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:

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

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

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

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

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.

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.
Phase II: Exhausting the Binding Resources

Adaptive Algorithm for filling the knapsacks:

For $t = t' + 1, \ldots, T$

1. Solve the UCB-LP and denote its optimal solution as $	ilde{x}$

$$\max_x \sum_{i=1}^k \left( \hat{\mu}_i(t) + \sqrt{\frac{2 \log T}{n_i(t)}} \right) x_i$$

s.t. $$\sum_{i=1}^k \left( \hat{c}_i(t) - \sqrt{\frac{2 \log T}{n_i(t)}} \right) x_i \leq b^{(t-1)}$$

$$x \geq 0, \ x_i = 0 \text{ for } i \in \mathcal{I}'$$

2. Normalize $	ilde{x}$ into a probability and play an arm accordingly

3. Update the knapsack process $b^{(t)}$ (remaining resource)
Combining the Two Phases

Proposition (Li, Sun & Y 2021, ICML)

The regret of our two-phase algorithm is bounded by

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

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:

\[
\begin{align*}
\min f_{\text{min}} & \quad \min f_{\text{min}} \quad \min f_{\text{min}} \\
& \quad x_i > 0 & \quad g \\
& \quad \min f_{\text{OPT}} \quad \min f_{\text{OPT}} \quad \min f_{\text{OPT}} \\
& \quad x_i > 0 & \quad g \\
& \quad \min f_{\Delta} \quad \min f_{\Delta} \quad \min f_{\Delta} \\
& \quad x_i = 0 & \quad g
\end{align*}
\]

These condition numbers generalize the optimality gap for the original (unconstrained) multi-armed bandits (Lai and Robbins (1985), Auer et al. (2002)).
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.
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:
  \[ \min \left\{ \min \{x_i^* | x_i^* > 0\}, \min \{OPT - OPT_i | x_i^* > 0\}, \min \{\Delta_i | x_i^* = 0\} \right\}. \]
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:

- $\sigma$ 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:

$$\min \{ \min \{ x_i^* | x_i^* > 0 \}, \min \{ \text{OPT} - \text{OPT}_i | x_i^* > 0 \}, \min \{ \Delta_i | x_i^* = 0 \} \}.$$

These condition numbers generalize the optimality gap for the original (unconstrained) multi-armed bandits (Lai and Robbins (1985), Auer et al. (2002)).
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Online Linear Programming</td>
</tr>
<tr>
<td>2</td>
<td>Fast Algorithms for (Binary) Online Linear Programming</td>
</tr>
<tr>
<td>3</td>
<td>A Fairer Online Interior-Point LP Algorithm</td>
</tr>
<tr>
<td>4</td>
<td>Online Bandits with Knapsacks</td>
</tr>
<tr>
<td>5</td>
<td>Online Fisher Markets</td>
</tr>
</tbody>
</table>
Fisher Markets for Resource-Allocation

Each agent $i$, with budget $w_i$, purchases an optimal bundle $x_i$ given price $p$

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 $p_j, j \in G$.

\[
\begin{align*}
\max \quad & u_i^T x_i := \sum_{j \in G} u_{ij} x_{ij} \\
\text{s.t.} \quad & p^T x_i := \sum_{j \in G} p_j x_{ij} \leq w_i, \\
& x_{ij} \geq 0, \quad \forall j,
\end{align*}
\]

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

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

where $x^*(p)$ is a maximizer of the utility maximization problem for every buyer $i$. 
The Aggregated Social Optimization Problem

\[
\begin{align*}
\text{max} & \quad \sum_{i \in B} w_i \log (u_i^T x_i) \\
\text{s.t.} & \quad \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\
& \quad x_{ij} \geq 0, \quad \forall i, j,
\end{align*}
\]
The Aggregated Social Optimization Problem

max \( \sum_{i \in B} w_i \log(u_i^T x_i) \)

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

\( x_{ij} \geq 0, \quad \forall i, j, \)

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

*Optimal dual (Lagrange) multiplier vector of equality constraints is an equilibrium price vector.*
The Aggregated Social Optimization Problem

\[
\begin{align*}
\max & \quad \sum_{i \in B} w_i \log(u_i^T x_i) \\
\text{s.t.} & \quad \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\
& \quad x_{ij} \geq 0, \quad \forall i, j,
\end{align*}
\]

**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

\[
\begin{align*}
\text{max} & \quad \sum_{i \in B} w_i \log(u_i^T x_i) \\
\text{s.t.} & \quad \sum_{i \in B} x_{ij} = (\leq) c_j, \quad \forall j \in G \\
& \quad x_{ij} \geq 0, \quad \forall i, j,
\end{align*}
\]

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?
Each agent $i$, with budget $w_i$, purchases an optimal bundle $x_i^t$ 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?
Let “offline” optimal solution be $x^*$ and “online” solution of $n$ orders be $x$, and

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

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

$$R_n^* = \sum_{j=1}^{n} \sum_{i=1}^{n} w_i \log(u_i^T x_i^*), \quad R_n = \sum_{i=1}^{n} w_i \log(u_i^T x_i)$$

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 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 \( x^* \) and “online” solution of \( n \) orders be \( x \), and

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

Then define

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

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

\[
\frac{\left( \prod_i (u_i^T x_i^*) w_i \right)^{1/n}}{\left( \prod_i (u_i^T x_i)^{w_i} \right)^{1/n}} \leq e^{o(n)/n}.
\]
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 $p^1_t$ 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:

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

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!
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!

Is there an online algorithm that relies on only $x_t$?
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)
  - Type II: (0, 1)
  - Arrival Probability = 0.5

Theorem (Jelota & Y (2022))

The expect optimal social value

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

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

$$\mathbb{E}[\| (Ax - b)^+ \|_2] \geq \sqrt{n}.$$
Consider the Dual of the Fisher Market

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

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

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

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 \( d := \frac{1}{n} c \) 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
   \[
   \mathbf{p}_{t+1} = \mathbf{p}_t - \gamma_t (\mathbf{d} - \mathbf{x}_t)
   \]
   \[
   \mathbf{p}_{t+1} = \mathbf{p}_{t+1} \lor 0
   \]
4. $\mathbf{x} = (x_1, ..., x_n)$

Again, line 3 performs (projected) stochastic gradient step.
Dual-Gradient Online Algorithm for Fisher-Markets

1: Initialize $p^1 = \epsilon e$, and for $t = 1, 2, \ldots, n$
2: Let $x_t$ be the individual optimal bundle solution for price vector $p^t$.
3: Update prices
   \[ p_{t+1} = p_t - \gamma_t (d - x_t) \]
   \[ p_{t+1} = p_{t+1} \lor 0 \]
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

- 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...**