Online Learning via Linear Programming

Yinyu Ye

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

December 13, 2021
(Joint work with many...)

NeurIPS OPT Workshop 2021

Offline and Online Linear Programming

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

Offline and Online Linear Programming

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

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

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

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

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

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

Offline and Online Linear Programming

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

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

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

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

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

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

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

Offline and Online Linear Programming

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

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

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

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

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

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

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

Offline and Online Linear Programming

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

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

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

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

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

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

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

Offline and Online Linear Programming

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

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

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

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

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

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

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

Primal and Dual Offline LPs

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

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

Primal and Dual Offline LPs

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

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

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

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

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

Primal and Dual Offline LPs

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

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

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

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

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

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

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

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

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

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

Random Permutation (RP) Model:

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

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

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

Random Permutation (RP) Model:

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

Both assume boundedness:

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

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

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

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

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

Random Permutation (RP) Model:

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

Both assume boundedness:

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

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

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

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

Data/Model Assumptions for Analyses

Stochastic Input (i.i.d) Model:

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

Random Permutation (RP) Model:

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

Both assume boundedness:

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

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

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

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

Competitive Ratio Summary of One-Sited Market

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

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

Competitive Ratio Summary of One-Sited Market

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

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

Remarks

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

Remarks

- The optimal online algorithms have been designed for the competitive ratio analyses 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 considers a stationary benchmark where the the decision variables are required to be the same at each time period.

Remarks

- The optimal online algorithms have been designed for the competitive ratio analyses 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 considers a stationary benchmark where the the decision variables are required to be the same at each time period.
- Recent focuses are on dealing with two-sited markets/platforms, regret analyses, simple and fast algorithms, interior-point online algorithm, extension to bandit models, ...

Today's Talk: Recent Developments

- Part (I): Fast algorithms for online linear programming
  - Setup: First observe $(r_t, \mathbf{a}_t)$ then decide x_t
- Part (II): A Fairer online interior-point LP algorithm
  - Setup: A "fair" online decision-making mechanism
- Part (III): Bandits with knapsacks
  - Setup: First choose "x_t" (the arm/customer), then observe $(r_t, \mathbf{a}_t)$

Other recent works on OLP: papers by Balseiro, Lu, and Mirrokni (2020,21), etc.

Regret Analysis and Model

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

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

Regret Analysis and Model

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

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

Then define

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

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

Regret Analysis and Model

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

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

Then define

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

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

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

Part (I): Equivalent Form of the Dual Problem

Recall the dual problem

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

can be rewritten as

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

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

Part (I): Equivalent Form of the Dual Problem

Recall the dual problem

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

can be rewritten as

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

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

After normalizing the objective, it becomes

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

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

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

Theorem (Li & Y (2019))

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

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

holds for all $n > m$.

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

Theorem (Li & Y (2019))

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

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

holds for all $n > m$.

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

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

Fast Online Algorithm for Online and Binary LP

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

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

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

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

Compute

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

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

Fast Online Algorithm for Online and Binary LP

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

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

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

4:

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

5: Compute

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

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

Line 5 performs (projected) stochastic gradient descent in the dual.

Performance Analysis

Theorem (Li, Sun & Y (2020))

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

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

under both the stochastic input and the random permutation models.

- $\tilde{O}$ omits the logarithm terms and the constants related to $(\bar{a}, \bar{r})$, but the algorithm does not require any prior knowledge on the constants.
- The optimal offline value is in the range $O(mn)$.
- The algorithms runs in *nm* times - the time to read in the data.

Fast Online LP Algorithm for Solving Offline LPs?

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

Fast Online LP Algorithm for Solving Offline LPs?

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

Consider a "Replicated" LP from the original LP

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

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.

Fast Online LP Algorithm for Solving Offline LPs?

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

Consider a "Replicated" LP from the original LP

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

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

The algorithm runs in $O(kmn)$ times.

Performance of the Variable-Replicating Algorithm

Proposition (Gao, Sun, Ye & Y (2020))

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

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

Performance of the Variable-Replicating Algorithm

Proposition (Gao, Sun, Ye & Y (2020))

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

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

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

Multi-knapsack Problem Instances - Binary LP

Benchmark dataset of Chu & Beasley implementation

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

Fast Online Algorithm as Pre-Classifier for LP

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

Fast Online Algorithm as Pre-Classifier for LP

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

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

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

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

Fast Online Algorithm as Pre-Classifier for LP

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

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

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

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

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

Implementation in LP Solvers

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

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

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

Part (II): A "Fairer" Online LP Algorithm

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

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

Part (II): A "Fairer" Online LP Algorithm

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

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

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

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

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

Part (II): A "Fairer" Online LP Algorithm

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

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

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

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

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

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

Behavior of the Simplex and Interior-Point

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

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

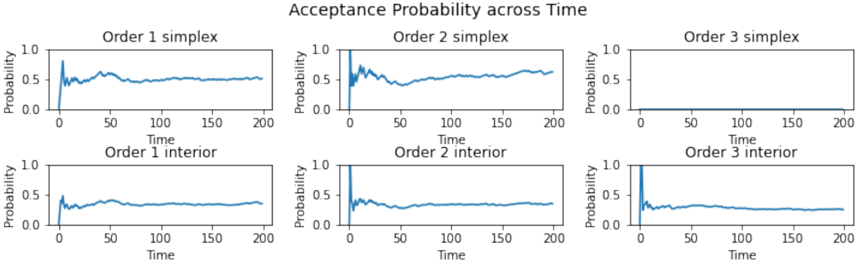
since the sample and offline LP may be degenerate or with multiple optimal solutions - a common property for real-life LP problems.

Behavior of the Simplex and Interior-Point

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

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

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



Fairness Desiderata: Time and Individual

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

Fairness Desiderata: Time and Individual

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

Individual Fairness: For certain customer types there exist **multiple** optimal allocation rules. Unfortunately, the optimal object value depends on the total resources spent, not on the resources spent on which groups - some individual or group may be ignored by the online algorithm/allocation-rule.

Fairness Desiderata: Time and Individual

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

Individual Fairness: For certain customer types there exist **multiple** optimal allocation rules. Unfortunately, the optimal object value depends on the total resources spent, not on the resources spent on which groups - some individual or group may be ignored by the online algorithm/allocation-rule.

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

Fairness Desiderata: Time and Individual

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

Individual Fairness: For certain customer types there exist **multiple** optimal allocation rules. Unfortunately, the optimal object value depends on the total resources spent, not on the resources spent on which groups - some individual or group may be ignored by the online algorithm/allocation-rule.

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

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

Fairer Solution for the Offline Problem

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

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

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

Fairer Solution for the Offline Problem

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

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

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

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

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

Fairer Solution for the Offline Problem

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

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

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

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

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

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

Our Result

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

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

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

Our Result

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

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

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

Key ideas in algorithm design:

- At each time t, we use interior-point method to obtain the sample analytic-center solution $\mathbf{y}_t$, and it is necessary to achieve the performance under weak non-degeneracy assumption and maintain fairness.
- We also adjust the right-hand-side 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

Part (III): Bandits with Knapsacks

Reverse the order of decisions and observations in online LP: decide x_t then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Part (III): Bandits with Knapsacks

Reverse the order of decisions and observations in online LP: decide x_t then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

Part (III): Bandits with Knapsacks

Reverse the order of decisions and observations in online LP: decide x_t then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

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

Part (III): Bandits with Knapsacks

Reverse the order of decisions and observations in online LP: decide x_t then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

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

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

Part (III): Bandits with Knapsacks

Reverse the order of decisions and observations in online LP: decide x_t then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

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

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

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

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

Part (III): Bandits with Knapsacks

Reverse the order of decisions and observations in online LP: decide x_t then observe $(\hat{r}_t, \hat{\mathbf{c}}_t)$.

Horizon: T time periods (T known a priori)

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

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

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

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

Goal: Select a subset of winning/optimal arms to maximize the total reward subject to the resource capacity constraints!

Offline Linear Program (LP) and Regret

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

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

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

Offline Linear Program (LP) and Regret

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

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

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

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

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

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

Literature and Our Result

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

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

Dual LP and Reduced Cost

$$\textit{Primal}: \max \ \boldsymbol{\mu}^\top \mathbf{x} \quad \text{s.t.} \ C\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0} \qquad \textit{Dual}: \min \ \mathbf{b}^\top \mathbf{y} \quad \text{s.t.} \ C^\top \mathbf{y} \ge \boldsymbol{\mu}, \mathbf{y} \ge \mathbf{0}$$

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

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

Proposition (Li, Sun & Y (2021))

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

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

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

Implications of the Regret Upper Bound

Two tasks to accomplish to reduce the regret:

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

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

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

Implications of the Regret Upper Bound

Two tasks to accomplish to reduce the regret:

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

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

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

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

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

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

Implications of the Regret Upper Bound

Two tasks to accomplish to reduce the regret:

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

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

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

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

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

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

Task II is often overlooked in the existing BwK literature.

Our Approach: A Two-Phase Algorithm

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

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

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

Our Approach: A Two-Phase Algorithm

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

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

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

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

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

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 $\tilde{\mathbf{x}}$

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

2. Normalize $\tilde{\mathbf{x}}$ into a probability and play an arm accordingly
3. Update the knapsack process $\mathbf{b}^{(t)}$ (remaining resource)

Combining the Two Phases

Proposition (Li, Sun & Ye 2021)

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 & Ye 2021)

The regret of our two-phase algorithm is bounded by

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

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

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

Combining the Two Phases

Proposition (Li, Sun & Ye 2021)

The regret of our two-phase algorithm is bounded by

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

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

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

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

Combining the Two Phases

Proposition (Li, Sun & Ye 2021)

The regret of our two-phase algorithm is bounded by

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

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

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

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

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

Summary

LP continues to play an important and significant role in today's online learning and decision-making!

Thank You