

Ski Rental

Yanze Song

The LOG Reading Group @ University of Alberta

November 5, 2023

Outline

1. Problem Setup
2. Performance Metric
3. Algorithms
4. OPD Framework
5. Algorithms with OPD
6. Summary
7. References

Outline

1. Problem Setup

2. Performance Metric

3. Algorithms

4. OPD Framework

5. Algorithms with OPD

6. Summary

7. References

Problem Setup

Definition (*Ski Rental Problem*)

- Rent skis: \$1 per day
- Buy skis: $B > 1$
- Total T days of skiing (unknown)
- On each day t , a player will be told if she'll ski or not, and then she'll decide to buy or to rent.

Goal: Develop an algorithm that costs as little money as possible in all cases

Problem Setup

Define the optimization variables as follows:

$$x = \begin{cases} 1, & \text{if buy skis} \\ 0, & \text{else} \end{cases} \quad z_t = \begin{cases} 1, & \text{if rent on day } t \\ 0, & \text{else} \end{cases}$$

Then the offline problem can be formulated as follows:

$$\begin{aligned} \min_{x, z_i} \quad & B \cdot x + \sum_{i=1}^T z_i \\ \text{s.t.:} \quad & x + z_i \geq 1, \forall i \\ & x \in \{0, 1\} \\ & z_i \in \{0, 1\}, \forall i \end{aligned}$$

Outline

1. Problem Setup
- 2. Performance Metric**
3. Algorithms
4. OPD Framework
5. Algorithms with OPD
6. Summary
7. References

Performance Metric

Definition (*Competitive Ratio*)

- Compare with the offline optimal under all inputs
- A deterministic online algorithm is α -competitive if

$$\alpha = \max_{\sigma} \frac{c(\text{Alg}, \sigma)}{c(\text{Opt}, \sigma)}$$

- A randomized online algorithm is α -competitive if

$$\alpha = \max_{\sigma} \frac{\mathbb{E}[c(\text{ALG}, \sigma)]}{c(\text{Opt}, \sigma)}$$

Outline

1. Problem Setup
2. Performance Metric
- 3. Algorithms**
4. OPD Framework
5. Algorithms with OPD
6. Summary
7. References

Algorithms: Deterministic

Algorithm (Alg_B)

Rent up to day $B - 1$, and then buy skis on day B

- $(2 - \frac{1}{B})$ -competitive

Two cases to consider:

- $T < B$: $\frac{c(Alg_B)}{c(Opt)} = \frac{T}{T} = 1$
- $T \geq B$: $\frac{c(Alg_B)}{c(Opt)} = \frac{2B-1}{B} = 2 - \frac{1}{B}$

$$\alpha = \max_{\sigma} \frac{c(Alg, \sigma)}{c(Opt, \sigma)} = \max\{1, 2 - \frac{1}{B}\} = 2 - \frac{1}{B}$$

Algorithms: Deterministic

Theorem (*Optimality of Alg_B*)

$2 - \frac{1}{B}$ is the optimal C.R. among all deterministic algorithms.

All deterministic algorithms have the form:

Alg_t : Rent up to day $t - 1$, and buy on day t

Then two cases to consider:

- $t > B$: Let $T > t > B$, then $\frac{c(Alg_t)}{c(Opt)} = \frac{t-1+B}{B} \geq \frac{B+B}{B} > 2 - \frac{1}{B}$
- $t < B$: Let $t \leq T \leq \frac{t-1+B}{2-\frac{1}{B}} < B$, then $\frac{c(Alg_t)}{c(Opt)} = \frac{t-1+B}{T} \geq 2 - \frac{1}{B}$

Algorithms: Randomized

A toy example ($B = 4$):

C.R.	T=1	T=2	T=3	T=4	T=5	T=6	...
Alg_1	4/1	4/2	4/3	4/4	4/4	4/4	
Alg_2	1/1	5/2	5/3	5/4	5/4	5/4	
Alg_3	1/1	2/2	6/3	6/4	6/4	6/4	
Alg_4	1/1	2/2	3/3	7/4	7/4	7/4	
Alg_5	1/1	2/2	3/3	4/4	8/4	8/4	
Alg_6	1/1	2/2	3/3	4/4	5/4	9/4	
...							

Each algorithm has its own expertise, i.e., is optimal w.r.t. a certain input T . In fact, the expertise of $\{Alg_t\}_{t=1}^4$ spans all possibilities of T , in other words, $\{Alg_t\}_{t \geq 5}$ are somehow useless (informally).

Algorithms: Randomized

Idea

We give each algorithm a chance to be used - we construct a probability distribution $\{p_t\}_{t=1}^{\infty}$ on $\{Alg_t\}_{t=1}^{\infty}$.

Recall that, a randomized algorithm is α -competitive if

$$\alpha := \max_{\sigma} \frac{\mathbb{E}[c(ALG, \sigma)]}{c(Opt, \sigma)}$$
$$\Rightarrow \sum_{t=1}^{\infty} c(Alg_t, \sigma) \cdot p_t \leq \alpha \cdot Opt(\sigma), \forall \sigma$$

Algorithms: Randomized

W.r.t. the toy example, if we want to achieve α -competitiveness, we need

$$4p_1 + 1p_2 + 1p_3 + 1p_4 + 1p_5 + 1p_6 + \dots \leq 1\alpha$$

$$4p_1 + 5p_2 + 2p_3 + 2p_4 + 2p_5 + 2p_6 + \dots \leq 2\alpha$$

$$4p_1 + 5p_2 + 6p_3 + 3p_4 + 3p_5 + 3p_6 + \dots \leq 3\alpha$$

$$4p_1 + 5p_2 + 6p_3 + 7p_4 + 4p_5 + 4p_6 + \dots \leq 4\alpha$$

$$4p_1 + 5p_2 + 6p_3 + 7p_4 + 8p_5 + 5p_6 + \dots \leq 4\alpha$$

\vdots

Two observations:

- $p_t = 0, \forall t > B$ - subtract the t^{th} from the $t + 1^{th}$ inequality
- After the B^{th} inequality, they're useless - same as the B^{th}

Algorithms: Randomized

After two simplifications:

- Keep the 1st inequality
- Subtract the t^{th} from the $t + 1^{th}$ inequality

we then have

$$Bp_1 + p_2 + p_3 + \cdots + p_B \leq \alpha$$

$$Bp_2 + p_3 + \cdots + p_B \leq \alpha$$

$$Bp_3 + \cdots + p_B \leq \alpha$$

$$\vdots$$

$$Bp_B \leq \alpha$$

Algorithms: Randomized

We're to minimize α s.t. the above constraints, that is, we're to solve the following problem:

$$\begin{aligned} \min_{p_t} \quad & \alpha \\ \text{s.t.:} \quad & Bp_1 + p_2 + p_3 + \cdots + p_B \leq \alpha \\ & Bp_2 + p_3 + \cdots + p_B \leq \alpha \\ & Bp_3 + \cdots + p_B \leq \alpha \\ & \vdots \\ & Bp_B \leq \alpha \\ & p_t \in [0, 1], \forall t \\ & \sum_{t=1}^{\infty} p_t = 1 \end{aligned}$$

Algorithms: Randomized

We may notice that

α is minimized \Leftrightarrow all inequalities become equalities.

Proof for “ \Rightarrow ”:

- Assume α is minimized, and there exist a strict inequality
- We then can adjust p_t s.t. all become strict inequalities, and then α can be further pushed down - contradicts with the minimality of α

Proof for “ \Leftarrow ”:

- Assume all are equalities
- It becomes solving a linear system with full rank. There's only one solution, and hence the minimality is achieved

Algorithms: Randomized

Algorithm

Choose Alg_t with $p_t = \frac{\alpha}{B}(\frac{B-1}{B})^{B-t}$, where $\alpha = \frac{1}{1-(1-\frac{1}{B})^B}$

- $\frac{1}{1-(1-\frac{1}{B})^B}$ -competitive
- $\frac{1}{1-(1-\frac{1}{B})^B} < \frac{e}{e-1}, \forall B; \lim_{B \rightarrow \infty} \frac{1}{1-(1-\frac{1}{B})^B} = \frac{e}{e-1} \approx 1.58$

Algorithms: Randomized

Theorem

$\frac{1}{1-(1-\frac{1}{B})^B}$ is the optimal C.R. among all randomized algorithms.

Do we need to prove the optimality of α if we achieve it by solving the above optimization problem?

- No, the optimality of α is guaranteed based on the procedures above
- The set of all deterministic algorithms is clear for this problem
- What if the problem is more complicated - some hidden deterministic algorithms?
- Unrealistic to traverse all randomized algorithms

Algorithms: Randomized

Theorem (*Yao's Principle*)

The expected cost of the best randomized algorithm on the worst-case input is no lower than the expected cost of the deterministic algorithm that performs the best on the worst-case input distribution, i.e., $\forall ALG, \forall X$:

$$\max_{x \in \mathcal{X}} \mathbb{E}[c(ALG, x)] \geq \min_{d \in D} \mathbb{E}[c(Alg_d, X)],$$

where

- ALG : Randomized algorithm, a random variable
- X : Input distribution, a random variable
- \mathcal{X} : Input set
- D : The set of all deterministic algorithms (indices)

Algorithms: Randomized

Theorem (*Yao's Principle*)

Under the competitive ratio context, we have $\forall \text{ALG}, \forall X$:

$$\max_{x \in \mathcal{X}} \frac{\mathbb{E}[c(\text{ALG}, x)]}{c(\text{Opt}, x)} \geq \min_{d \in D} \mathbb{E}\left[\frac{c(\text{Alg}_d, X)}{c(\text{Opt}, X)}\right],$$

or equivalently,

$$\underbrace{\min_{\text{ALG}} \max_{x \in \mathcal{X}} \frac{\mathbb{E}[c(\text{ALG}, x)]}{c(\text{Opt}, x)}}_{\text{optimal C.R.}} \geq \underbrace{\max_X \min_{d \in D} \mathbb{E}\left[\frac{c(\text{Alg}_d, X)}{c(\text{Opt}, X)}\right]}_{\text{lower bound (tight?)}}$$

We're then to construct the worst-case input distribution X .

Outline

1. Problem Setup
2. Performance Metric
3. Algorithms
- 4. OPD Framework**
5. Algorithms with OPD
6. Summary
7. References

OPD Framework

- Primal:

$$\begin{aligned} \min_{x, z_i} \quad & B \cdot x + \sum_{i=1}^T z_i \\ \text{s.t.:} \quad & x + z_i \geq 1, \forall i \\ & x \in \{0, 1\} \\ & z_i \in \{0, 1\}, \forall i \end{aligned}$$

p^* is the opt value

- Relaxed Primal:

$$\begin{aligned} \min_{x, z_i} \quad & B \cdot x + \sum_{i=1}^T z_i \\ \text{s.t.:} \quad & x + z_i \geq 1, \forall i \\ & x \geq 0 \\ & z_i \geq 0, \forall i \end{aligned}$$

p_r^* is the opt value

- Dual:

$$\begin{aligned} \max_{y_i} \quad & \sum_{i=1}^T y_i \\ \text{s.t.:} \quad & \sum_{i=1}^T y_i \leq B \\ & 0 \leq y_i \leq 1, \forall i \end{aligned}$$

d^* is the opt value

$$p^* \geq p_r^*, \text{ by relaxation}$$

$$p_r^* \geq d^*, \text{ by weak duality}$$

$$d^* \leq p_r^* \leq p^*$$

OPD Framework

Several clarifications:

- p_t : The primal objective value up to time t incurred by our algorithm
- d_t : The dual objective value up to time t incurred by our algorithm
- $\alpha := \min_{\sigma} \frac{c(\text{Opt}, \sigma)}{c(\text{Alg}, \sigma)} \leq 1$, so we are to maximize α

So far,

- What we have: $d^* \leq p_r^* \leq p^*$
- Further more, we have $d_T \leq d^* \leq p_r^* \leq p^*$
- What we want: $\alpha p_T \leq p^*$
- The above holds if $\alpha p_T \leq d_T$

That is,

$$\alpha p_T \leq d_T + \text{primal and dual feasibility} \Rightarrow \alpha\text{-competitiveness}$$

OPD Framework

In the online manner, $\alpha p_T \leq d_T$ is hard to achieve, so we want to achieve some inequality at each step, and then the cumulative effect yields $\alpha p_T \leq d_T$, so

$$\begin{aligned} & \alpha p_T \leq d_T \\ \Leftrightarrow & \alpha \left(\sum_{t=1}^T (p_t - p_{t-1}) + p_0 \right) \leq \left(\sum_{t=1}^T (d_t - d_{t-1}) + d_0 \right) \\ \Leftrightarrow & \alpha (p_t - p_{t-1}) \leq d_t - d_{t-1}, \forall t, \text{ if } p_0 = d_0 = 0 \end{aligned}$$

That is, if $p_0 = d_0 = 0$, to achieve $\alpha p_T \leq d_T$, we need

$$\alpha \Delta p_t \leq \Delta d_t, \forall t$$

which is called the incremental inequality.

OPD Framework

Previously,

$$\alpha p_T \leq d_T + \text{primal and dual feasibility} \Rightarrow \alpha\text{-competitiveness}$$

Now, if $p_0 = d_0 = 0$,

$$\alpha \Delta p_t \leq \Delta d_t + \text{primal and dual feasibility} \Rightarrow \alpha\text{-competitiveness}$$

Generally, to design an α -competitive algorithm, we need:

- Primal and dual feasibility
- Initial conditions (p_0, d_0 , not always 0) and terminal conditions (e.g. a necessary condition for maximizing α)
- Update methods for the primal and dual variables s.t. the incremental inequality holds (depends on the initial and terminal conditions)

Outline

1. Problem Setup
2. Performance Metric
3. Algorithms
4. OPD Framework
- 5. Algorithms with OPD**
6. Summary
7. References

Algorithms with OPD: Deterministic

W.r.t. the relaxed problem, need to design x_0, y_0 , and the update methods for x_t, z_t, y_t .

Idea

Assume we want to use the following (very simple) idea:

- For feasibility and simplicity, choose $x_0 = y_0 = 0$
- Since deterministic, we buy on day G , i.e., $x_G = 1$, and keep $x_{t>G} = 1$
- For feasibility and simplicity, keep $x_t \in [0, 1]$, $z_t = 1 - x_t$, so keep $z_{t>G} = 0$
- $\Delta p_{t>G} = 0$, no need for $\Delta d_{t>B} = y_{t>B}$ to leave the room for the primal variables to change, so $y_{t>G} = 0$. Then we fully use, and evenly split B to $y_{t \leq G}$, i.e. $y_{t \leq G} = \frac{B}{G}$
- Due to the previous, $G \geq B$, or otherwise y_t is infeasible

Algorithms with OPD: Deterministic

For the **fractional** ski rental problem (relaxed primal), here's our design:

- Initial condition:

$$x_0 = y_0 = 0$$

- **Terminal condition** (motivated by the optimality of Alg_B):

Buy on day B , i.e. $x_B = 1$, and $x_{t>B} = 1$

- Update method of the primal variable - incomplete:

$$x_t \in [0, 1], z_t = 1 - x_t, \text{ so } z_{t>B} = 0$$

- Incremental inequality:

$$\alpha \Delta p_t \leq \Delta d_t \Rightarrow \alpha(B(x_t - x_{t-1}) + z_t) \leq y_t$$

- Update method of the dual variable:

$$y_{t \leq B} = 1, \text{ and } y_{t > B} = 0$$

Algorithms with OPD: Deterministic

Algorithm (Draft)

$x_0 = 0$

for each day t :

if $t < B$:

update x_t s.t.
$$\begin{cases} \alpha(B(x_t - x_{t-1}) + z_t) \leq 1 \text{ with } \alpha \text{ and } z_t \text{ to be defined} \\ x_B = 1 \\ x_t \in [0, 1] \end{cases}$$

$z_t = 1 - x_t$

else:

$x_t = 1, z_t = 0$

Algorithms with OPD: Deterministic

Now we solve the (in)equalities:

- $\alpha(B(x_t - x_{t-1}) + (1 - x_t)) \leq 1 \Leftrightarrow x_t \leq \underbrace{\left(1 + \frac{1}{B-1}\right)x_{t-1} + \frac{1-\alpha}{\alpha(B-1)}}_{\text{update method for } x_t \text{ is here}}$
- $x_t = \left(1 + \frac{1}{B-1}\right)x_{t-1} + \frac{1-\alpha}{\alpha(B-1)} = \frac{\alpha-1}{\alpha} \left(1 - \left(1 + \frac{1}{B-1}\right)^t\right)$ is a good candidate, increasing
- $x_B = 1 \Rightarrow \alpha = 1 - \frac{1}{\left(1 + \frac{1}{B-1}\right)^B}$

We then check if α -competitiveness is guaranteed (obvious):

- Primal and dual feasibility: $x_0 = 0, x_B = 1, x_t$ is increasing
- $\alpha(B(x_t - x_{t-1}) + z_t) \leq 1$: Directly holds

Algorithms with OPD: Deterministic

Algorithm

$x_0 = 0$

for each day t :

if $t < B$:

$$x_t = \left(1 + \frac{1}{B-1}\right)x_{t-1} + \frac{1-\alpha}{\alpha(B-1)} \text{ with } \alpha = 1 - \frac{1}{\left(1 + \frac{1}{B-1}\right)^B}$$

$$z_t = 1 - x_t$$

else:

$$x_t = 1, z_t = 0$$

- $1 - \frac{1}{\left(1 + \frac{1}{B-1}\right)^B}$ -competitive for the **fractional** problem ($\alpha p_{r_T} \leq d_T \leq p_r^*$)
- $1 - \frac{1}{\left(1 + \frac{1}{B-1}\right)^B} > \frac{e-1}{e}, \forall B; \lim_{B \rightarrow \infty} 1 - \frac{1}{\left(1 + \frac{1}{B-1}\right)^B} = \frac{e-1}{e}$

Algorithms with OPD: Deterministic

We can derive the algorithm even **without** the intuition of the terminal condition $x_B = 1$:

- Initial condition:

$$x_0 = y_0 = 0$$

- Terminal condition:

Buy on day $G \geq B$, i.e. $x_G = 1$, and $x_{t>G} = 1$

- Update method of the primal variable - incomplete:

$$x_t \in [0, 1], z_t = 1 - x_t, \text{ so } z_{t>G} = 0$$

- Incremental inequality:

$$\alpha \Delta p_t \leq \Delta d_t \Rightarrow \alpha(B(x_t - x_{t-1}) + z_t) \leq y_t$$

- Update method of the dual variable:

$$y_{t \leq G} = \frac{B}{G}, \text{ and } y_{t > G} = 0$$

Algorithms with OPD: Deterministic

By the incremental inequality,

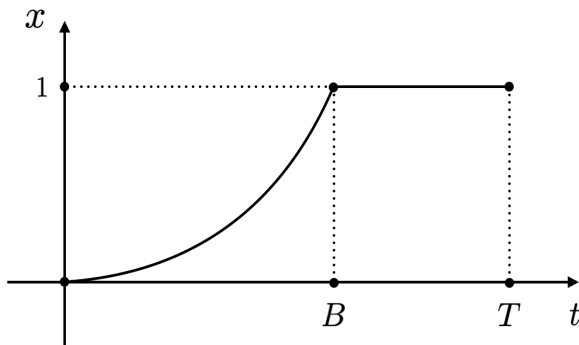
$$\alpha(B(x_t - x_{t-1}) + (1 - x_t)) \leq \frac{B}{G} \Leftrightarrow x_t \leq \frac{B - \alpha G}{\alpha(B - 1)G} + (1 + \frac{1}{B - 1})x_{t-1}$$

Pick $x_t = \frac{B - \alpha G}{\alpha(B - 1)G} + (1 + \frac{1}{B - 1})x_{t-1} = \frac{\alpha G - B}{\alpha G}(1 - (1 + \frac{1}{B - 1})^t)$, and by $x_G = 1$ we have

$$\alpha = \frac{B(1 - \frac{1}{(1 + \frac{1}{B - 1})^G})}{G}$$

which is essentially a function α of G , where $G \geq B$. Note that it's decreasing, so if we want to maximize α , we need $G = B$, and this yields the algorithm above.

Algorithms with OPD: Deterministic



The “potential” of buying skis is increasing

Algorithms with OPD: Randomized

W.r.t. the primal problem (un-relaxed),

- $\alpha p_T \leq d_T$ not satisfied (but $\alpha p_{r_T} \leq d_T$ satisfied)
- Dual feasibility satisfied
- Primal feasibility not satisfied

Can we extend this result to the integral problem?

- Buy $x_t \in [0, 1]$ amount of skis, rent $z_t \in [0, 1]$ amount of skis, $x_t + z_t = 1$
- **Fractions** \Leftrightarrow **Expectations** \Leftrightarrow Probabilities
- $x_t = \mathbb{E}[X_t] = \mathbb{P}_t^B$, $z_t = \mathbb{E}[Z_t] = \mathbb{P}_t^R$, where X_t, Z_t are standard Bernoulli RVs
- Update $\mathbb{P}_t^B, \mathbb{P}_t^R$ instead

Algorithms with OPD: Randomized

Algorithm

$$\mathbb{P}_0^B = 0$$

for each day t :

if $t < B$:

$$\mathbb{P}_t^B = \left(1 + \frac{1}{B-1}\right) \mathbb{P}_{t-1}^B + \frac{1-\alpha}{\alpha(B-1)} \text{ with } \alpha = 1 - \frac{1}{\left(1 + \frac{1}{B-1}\right)^B}$$

$$\mathbb{P}_t^R = 1 - \mathbb{P}_t^B$$

Decide to buy or to rent based on the distribution $\{\mathbb{P}_t^B, \mathbb{P}_t^R\}$

else:

$$\mathbb{P}_t^B = 1, \mathbb{P}_t^R = 0, \text{ i.e., buy}$$

Algorithms with OPD: Randomized

Now the primal feasibility is satisfied, what about $\alpha \mathbb{E}[p_T] \leq d_T$? Actually, we have

$$\mathbb{E}[p_T] = p_{r_T}, \text{ and hence } \alpha \mathbb{E}[p_T] \leq d_T,$$

where the equality can be proved as below:

$$\mathbb{E}[p_t] = \mathbb{E}[X_t + \sum_{i=1}^t Z_i] = \mathbb{E}[X_t] + \sum_{i=1}^t \mathbb{E}[Z_i] = \mathbb{P}_t^B + \sum_{i=1}^t \mathbb{P}_t^R = x_t + \sum_{i=1}^t z_i = p_{r_T}.$$

Therefore, $1 - \frac{1}{(1 + \frac{1}{B-1})^B}$ can be extended to the integral problem.

Outline

1. Problem Setup
2. Performance Metric
3. Algorithms
4. OPD Framework
5. Algorithms with OPD
- 6. Summary**
7. References

Summary

For the ski rental problem,

- Optimal C.R. for the deterministic algorithm: $2 - \frac{1}{B}$
- Optimal C.R. for the randomized algorithm: $\frac{1}{1 - (1 - \frac{1}{B})^B}$
- We use the OPD framework to derive a $\frac{1}{1 - (1 - \frac{1}{B})^B}$ -competitive deterministic algorithm for the fractional ski rental problem
- Then we extend the $\frac{1}{1 - (1 - \frac{1}{B})^B}$ -competitiveness to the integral problem using randomization

Outline

1. Problem Setup
2. Performance Metric
3. Algorithms
4. OPD Framework
5. Algorithms with OPD
6. Summary
- 7. References**

References

Thank You!