An Introduction to Competitive Analysis for Online Optimization

Maurice Queyranne
University of British Columbia,
and IMA Visitor (Fall 2002)

IMA, December 11, 2002

Overview

Online Optimization

- sequences $\sigma = (\sigma_1, \sigma_2, \dots)$ of *requests*
- every request must be *served* when it arises
  this involves a *service decision*
- service decisions must be made *online*, i.e.,
  without knowledge of the future requests
- objective is to minimize some measure of
  the *total cost* of all these decisions

Example: Ski Rental

Tamon starts to enjoy skiing

He must decide whether to buy skis,
or to keep renting them.

- renting costs *$20* per day
- buying costs *$300*

Tamon's best decision depends on
how many days he will be skiing

... and he does not know that when he makes these decisions

$\boxed{\textit{What should Tamon do?}}$

(Ski Rental: renting: \$20/day; buying: \$300)

Off-line Solution:

If Tamon knew today that
he will be skiing d days (*instance* I_d),
his problem would be easy:

- if $20\,d \leq 300$ then rent
- else buy

$\Rightarrow$ offline optimum cost

$$OPT(I_d) = \min\{20\,d, 300\}$$

... but Tamon does not know d

General Online Ski Rental Algorithm A_x:

- rent for up to x days
- then buy (if still skiing)

How to evaluate the cost of an online algorithm?

(General Ski Rental Algorithm A_x

- *rent for up to x days, then buy)*

If Tamon ends up skiing d days,
his actual cost is

$$C(A_x, I_d) = \begin{cases} 20\,d & \text{if } d < x \\ 20\,x + 300 & \text{otherwise} \end{cases}$$

whereas he could have paid only

$$\text{OPT}(I_d) = \min\{20\,d, 300\}$$

... but we don't know which case will apply

(General Ski Rental Algorithm A_x

- *rent for up to x days, then buy*

$$
\begin{aligned}
C(A_x, I_d) &= \begin{cases} 20\,d & \textit{if } d < x \\ 20\,x + 300 & \textit{otherwise} \end{cases} \\
\mathrm{OPT}(I_d) &= \min\{20\,d, 300\}
\end{aligned}
$$

)

Example:

If he decides to buy after $x = 5$ days
and he skis $d \geq 5$ days
then he spends $20 \times 5 + 300 = \$400$

If he has to quit just after buying, i.e., $d = 5$
then he could have spent only $20 \times 5 = \$100$

He may end up paying **4 times** as much
as $\mathrm{OPT}(I_5)$ (... had he known $d = 5$)

Competitive Ratio

Let $\mathcal{I}$ denote a set of instances (possible request sequences)

Algorithm A is *c-competitive* (w.r.t. $\mathcal{I}$) if $$C(A, I) \leq c\, \mathrm{OPT}(I) \quad \text{for all } I \in \mathcal{I}$$

c is called the *competitive ratio* of algorithm A

Example:
What is the competitive ratio of Tamon's A_5 algorithm?

(What is the competitive ratio of Tamon's A_5 *algorithm?)*

The Adversary Problem:

Given algorithm A $$\text{find } I \in \mathcal{I} \text{ to maximize } \frac{C(A, I)}{\text{OPT}(I)}\ ?$$

Here, given ski rental algorithm A_x ($x \geq 0$)

$$\text{find } d \geq 0 \text{ to maximize } \frac{C(A_x, I_d)}{\text{OPT}(I_d)}\ ?$$

(Competitive ratio:
$c(A_x) = \max\{C(A_x, I_d)/OPT(I_d) : d \geq 0\}$ *)*

Tamon Ski Rental Adversary Problem:

Maximum is attained when $d = x$ and

$$c(A_x) = \frac{C(A_x, I_x)}{\mathrm{OPT}(I_x)} = \frac{20\,x + 300}{\min\{20\,x,\ 300\}}$$

Example:
Algorithm A_5 (rent for 5 days, then buy) is 4-competitive (since 400/100 = 4)

Can Tamon be more competitive?

The Algorithm Design Problem:

$$\boxed{\text{find an algorithm } A \text{ to minimize } c(A)?}$$

Ski rental: find $x \geq 0$ to minimize $c(A_x)$?

Ski Rental Algorithm Design:

find $x \geq 0$ to minimize $c(A_x)$?

Proposition: For the ski rental problem with
- buying cost b dollars and
- rental cost r dollars/day,

it is optimum to rent for b/r days, then buy
The resulting algorithm is 2-competitive

Proof:
(i) For any $x \geq 0$,

$$c(A_x) = \frac{r\,x + b}{\min\{r\,x,\ b\}} \geq 2$$

(ii) for $x = b/r$, $c(A_{b/r}) = 2$ QED

Example: for $b = \$300$ and $r = \$20$/day,
Tamon should rent for 15 days, and then buy
This algorithm is 2-competitive,
and this is best possible

- *Is it really best possible?*

(Ski rental algorithm $A_{b/r}$ is 2-competitive
- *Is this really best possible?)*

How about using a Randomized Algorithm?

- use *coin flips* (random bits) to make decisions

Example: randomized ski rental algorithm $\tilde{A}$:
buy after renting for
- 10 days, with probability 0.5, or
- 15 days, with probability 0.5

His expected cost $E([C(\tilde{A}, I_d)]$

$$
\begin{aligned}
&= 0.5 \cdot C(A_{10}, I_d) + 0.5 \cdot C(A_{15}, I_d) \\
&= \begin{cases} 20\,d & \text{if } 0 \leq d \leq 10 \\ 0.5(500) + 0.5(20\,d) & \text{if } 10 < d \leq 15 \\ 0.5(500) + 0.5(600) & \text{if } d > 15 \end{cases}
\end{aligned}
$$

and

$$
\begin{aligned}
c(\tilde{A}) &= \max_{d \geq 0} E[C(\tilde{A}, I_d)]/OPT(I_d) \\
&= E[C(\tilde{A}, I_{16})]/OPT(I_{16}) \\
&= 550/300 = 1.8\bar{6} < 2
\end{aligned}
$$

Competitiveness of a Randomized Algorithm:

Randomized Algorithm $\tilde{A}$ is *c-competitive* (with respect to instance class $\mathcal{I}$) if

$$E[C(\tilde{A}, I)] \leq c\,\mathrm{OPT}(I) \quad \text{for all } I \in \mathcal{I}$$

Randomized Algorithm Design Problem:

find a randomized algorithm $\tilde{A}$
to minimize $c(\tilde{A})$?

Example: Tamon Ski Rental:

- *what is the best competitive ratio of a randomized ski rental algorithm?*

(What is the best competitive ratio of a randomized ski rental algorithm?)

Which Adversary to use? Recall that $\tilde{A}$ is c-competitive (w.r.t. $\mathcal{I}$) if

$$E[C(\tilde{A}, I)] \le c \, \mathrm{OPT}(I) \quad \text{for all } I \in \mathcal{I}$$

Thus the instance I must be fixed *before* the expectation is taken

An **Oblivious** Adversary must choose her (worst) instance for randomized alg. $\tilde{A}$ *without* knowledge of the realizations of the random variables used by $\tilde{A}$

(Obliv.) Adversary Lower Bound Problem

Find the largest constant c^* such that: for every randomized algorithm $\tilde{A}$ there exists an instance $I_{\tilde{A}} \in \mathcal{I}$ such that $E[C(\tilde{A}, I_{\tilde{A}})] \ge c^* OPT(I_{\tilde{A}})$?

Lower bound problem: maximize c^* subject to

$$\forall\, \tilde{A} \quad \exists\, I_{\tilde{A}} \quad E[C(\tilde{A}, I_{\tilde{A}})] \geq c^*\, OPT(I_{\tilde{A}})$$

Key Insight (Yao, 1977):

A randomized algorithm may be viewed as a random choice between deterministic algorithms

The competitive ratio of randomized alg. A_P (specified by probability distribution P) is:

$$c(A_P) = \sup_{I \in \mathcal{I}} \frac{E_P[c(A_P, I)]}{\mathrm{OPT}(I)}$$

The *best* competitive ratio for Designer is:

$$c^* = \inf_P \sup_{I \in \mathcal{I}} \frac{E_P[c(A_P, I)]}{\mathrm{OPT}(I)}$$

(The Lower Bound problem
for randomized algorithms)

The whole situation may be viewed as a *2-Person Zero-Sum Game* where:

- the *players* are the Algorithm Designer and the Adversary

- Adversary's *strategies* are all instances $I \in \mathcal{I}$
- Designer strategies are all *deterministic* algorithms $A \in \mathcal{A}$

- the *payoff* to Designer is $c(A, I)/\text{OPT}(I)$

The players may use *randomized strategies*:

- Designer chooses algorithms according to probability distribution P on $\mathcal{A}$
- Adversary chooses instances according to probability distribution Q on $\mathcal{I}$

The Lower Bound problem as

a 2-person zero-sum game

Best competitive ratio for Designer:

$$c^* = \inf_{P} \sup_{I \in \mathcal{I}} E_P \left[\frac{c(A_P, I)}{\mathsf{OPT}(I)} \right]$$

von Neumann Minimax Theorem:

$$\boxed{c^* = \sup_{Q} \inf_{A \in \mathcal{A}} E_Q \left[\frac{c(A, I_Q)}{\mathsf{OPT}(I_Q)} \right]}$$

$\Rightarrow$ any particular Q gives a lower bound:

$$c^* \leq \inf_{A \in \mathcal{A}} E_Q \left[\frac{c(A, I_Q)}{\mathsf{OPT}(I_Q)} \right]$$

(The Lower Bound problem as
a 2-person zero-sum game, continued)

$$c^* \leq \inf_{A \in \mathcal{A}} E_Q\left[\frac{c(A, I_Q)}{\mathrm{OPT}(I_Q)}\right]$$

Example: For Ski Rental, Adversary may choose

- instance I_5 with probability 0.5, and
- instance I_{20} with probability 0.5

If $I_Q = I_5$ then $\mathrm{OPT}(I_5) = 100$ and

$$\frac{c(A_x, I_5)}{\mathrm{OPT}(I_5)} = \begin{cases} \frac{20\,x+300}{100} & \text{if } 0 \leq x \leq 5 \\ \\ \frac{100}{100} & \text{if } x > 5 \end{cases}$$

else $I_Q = I_{20}$ then $\mathrm{OPT}(I_{20}) = 300$ and

$$\frac{c(A_x, I_{20})}{\mathrm{OPT}(I_{20})} = \begin{cases} \frac{20\,x+300}{300} & \text{if } 0 \leq x \leq 20 \\ \\ \frac{400}{300} & \text{if } x > 20 \end{cases}$$

(Ski rental Adversary strategy
Prob$\{I_Q = I_5\} =$ *Prob*$\{I_Q = I_{20}\} = 0.5$ *)*

$$E_Q\left[\frac{c(A_x, I_Q)}{\mathrm{OPT}(I_Q)}\right] = \begin{cases} 0.5(0.2\,x + 3) + 0.5(x/15 + 1) \\ \qquad\qquad \text{if } 0 \le x \le 5 \\ 0.5 \times 1 + 0.5(x/15 + 1) \\ \qquad\qquad \text{if } 5 < x \le 20 \\ 0.5 \times 1 + 0.5(4/3) \\ \qquad\qquad \text{if } x > 20 \end{cases}$$

and

$$\inf_{x \ge 0} \; E_Q\left[\frac{c(A_x, I_Q)}{\mathrm{OPT}(I_Q)}\right] = \frac{7}{6} = 1.1\bar{6} \quad (\text{for } x > 20)$$

- *Can we reduce the gap between the lower bound 7/6 and upper bound 11/6 ?*

(Ski rental: can we reduce the gap between the Lower and Upper bounds?)

Adversary's Lower Bound problem:

$$c^* = \sup_{Q} \inf_{A \in \mathcal{A}} E_Q \left[\frac{c(A, I_Q)}{\mathrm{OPT}(I_Q)} \right]$$

Normalize by rescaling monetary units and time so $r = 1$ (per fortnight) and $b = 1$ (ski set)

$$\frac{c(A_z, I_u)}{\mathrm{OPT}(I_u)} = \begin{cases} \frac{z+1}{u} & \text{if } 0 \le z \le u \le 1 \\ z+1 & \text{if } 0 \le z \le u \text{ and } u > 1 \\ u & \text{if } z > u > 1 \\ 1 & \text{otherwise} \end{cases}$$

- for any distribution Q of u, optimum $z \le 1$
- then ratio decreases in u when $u > 1$

$\Rightarrow$ hence assume $0 \le u \le 1$ and $0 \le z \le 1$

Adversary's Lower Bound problem (cont'd)

Find $q(u)$ for all $0 \le u \le 1$ to

maximize c

$$\text{s.t.} \quad \int_0^z q(u)\,du + \int_z^1 \frac{1+z}{u} q(u)\,du \ge c \qquad \text{for all } z \in [0,1]$$

$$\int_0^1 q(u)\,du = 1$$

$$q(u) \ge 0 \qquad \text{for all } u \in [0,1]$$

Optimum solution:

$$c^* = \frac{e}{e-1} \approx 1.58$$

$$q^*(u) = \begin{cases} u\,\frac{\exp(1-u)}{e-1} & \text{if } 0 \le u < 1 \\ 0 & \text{if } u > 1 \end{cases}$$

and a probability mass:

$$\text{Prob}\{u = 1\} = \frac{1}{e-1}$$

Designer problem:

Find $p(z)$ for all $0 \le z \le 1$ to

minimize c

$$\text{s.t.} \quad \int_0^u \frac{1+z}{u} p(z)\,dz + \int_u^1 p(z)\,dz \le c \qquad \text{for all } u \ge 0$$

$$\int_0^1 p(z)\,dz = 1$$

$$p(z) \ge 0 \qquad \text{for all } z \in [0,1]$$

(a dual to the Adversary's problem)

Optimum solution:

$$p^*(z) = \begin{cases} \frac{\exp(z)}{e-1} & \text{if } 0 \le z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$c^* = \frac{e}{e-1}$$