Recap: Online Algorithms

~ don't know anything about the future (not Poisson)
what happens in the worst-case?
→ take decisions online

- Ski Rental
  - don't know how long we are skiing for (weather/accident)
  - each day decide: continue renting OR buy

→ more formal:

\( u \): days we end up skiing | chosen by Adv.
\( Z \): the day we buy skis | chosen by us

\[
\text{cost}_{\text{Alg}} = \begin{cases} 
  u & u \leq Z \quad \text{(always rent)} \\
  Z + 1 & u > Z \quad \text{(we buy + rent 2 days)}
\end{cases}
\]

\[
\text{cost}_{\text{opt}} = u = \min (u, Z) \quad \text{best offline Alg.} \text{ (knows the future)}
\]
OA Analysis: Comp ratio

\[ \text{cost}_A \leq \Gamma \cdot \text{cost}_{opt} \]

\[ \Gamma = \frac{\text{cost}_A}{\text{cost}_{opt}} \]

Det. Ski Rental 2-comp.

Q: can we do better?

---

so what's the problem?

Adv. knows what we will do \( \Rightarrow \) can always make us pay a lot

Idea: randomize our strategy!

Adv: still knows what we will do

(i.e. flip a coin w p=0.8 & then do X)

BUT does not know random outcome (is it heads)

\( \Rightarrow \) forces Adv. to prepare for multiple outcomes
Randomized Ski Rental

Approach I

choose \( z_1 \) with probability \( p_1 \)

else choose \( z_2 \) \((p_2 = (1-p_1))\)

\[
\text{cost}_A = \begin{cases} 
  u, & u \leq z_1 \\
  p_1 \cdot (z_1 + 1) + p_2 \cdot u, & z_1 < u < z_2 \\
  p_1 \cdot (z_1 + 1) + p_2 \cdot (z_2 + 1), & z_2 \leq u
\end{cases}
\]

Obs: for Adv. it makes sense to either have

\[ u = z_1 + \varepsilon \quad \text{or} \quad z_2 + \varepsilon \]

\( \sim \) immediately stop doing after we buy

\[ z_1 = \frac{1}{2} \quad z_2 = 1 \]

\[ u = z_1 + \varepsilon \]

\[ u = z_2 \]

\[ \text{cost}_A = p_1 \cdot (z_1 + 1) + p_2 \cdot z_1 \]

\[ = p_1 + \frac{1}{2} \]

\[ \text{cost}_A = p_1 \cdot (z_1 + 1) + p_2 \cdot (z_2 + 1) \]

\[ = 2 - \frac{1}{2} p_1 \]
\[
\frac{\cos^2 A}{\cot (\frac{\pi}{2})} = 2p_1 + 1
\]

\[
\frac{\cos^2 A}{\cot (-1)} = 2 - \frac{1}{2} p_1
\]

\[
\Rightarrow 2p_1 + 1 = 2 - \frac{1}{2} p_1
\]

\[
\Rightarrow p_1 = \frac{2}{5}
\]

\[
\Rightarrow \frac{\cos A}{\cos \alpha +} = \frac{9}{5} < 2
\]

\[
\Rightarrow \text{can do better with Randomization!}
\]
Approach 2

- don’t choose 2 values
  \[ \rightarrow \text{INF many} \ldots \]
  choose a distribution

- now to get our comp. ratio
  \[
  \frac{p_1}{Z_1} \quad \frac{p_2}{Z_2} \quad p(z)
  \]

\[
\frac{\int_{z=0}^{z=1} (z+1) p(z) \, dz \, du + \int_{u=0}^{u=1} u \cdot p(z) \, dz \, du}{\int_0^1 u \, d(u) \, du}
\]

\[ \overset{\text{Adv}}{\overset{\text{tries to max.}}{\rightarrow}} \]
\[ \overset{\text{Alg}}{\overset{\text{tries to minimize}}{\rightarrow}} \]

too complex instead recall

\[
\text{cost}_A(u) \leq r \cdot \text{cost}_{OPT}(u) \quad \text{for all} \quad u
\]

\[ \rightarrow \text{if we can ensure that we do not pay much more than the OPT for any input} \quad u \quad \text{(that Adv. can choose)} \quad \text{it doesn’t matter what Adv. does} \]
\[ \rightarrow \text{we get} \quad r\text{-comp.} \]
\[ \text{cost}_{A}(u) \leq r \cdot \text{cost}_{\text{opt}}(=u) \]

\[ \int_{0}^{u} (z+1) p(z) \, dz + \int_{u}^{1} u \cdot p(z) \, dz \leq r \cdot u \]

\[ \text{if } u > z \text{ buy} \]

\[ \text{if } u < z \text{ rent} \]
\[ \int_0^{z+1} p(z) \, dz + u \int_0^1 p(z) \, dz = \gamma \cdot u \]

Differentiating, we have:

\[ \frac{\partial}{\partial u} \left[ \int_a^b f(x) \, dx \right] = \frac{\partial}{\partial u} \left[ F(b) - F(a) \right] \]

\[ F(u) - F(0) = f(b) - f(a) \]

Hence:

\[ f(a) \quad \Rightarrow \quad u \cdot \int_0^1 p(z) \, dz \]

\[ (u+1) \, p(u) + \int_0^1 p(z) \, dz + u \cdot -p(u) \]

\[ = p(u) + \int_u^1 p(z) \, dz = \gamma \]

\[ \frac{\partial}{\partial u} \frac{p(u)}{u} = 0 \quad \Rightarrow \quad p(u) = \alpha \cdot e^u \]

Putting in and \( \alpha = \frac{1}{e-1} \)

\[ \frac{e^u}{e-1} \]
\[ r = p(u) + \int_{-1}^{1} p(u(z)) dz \]

\[ = \frac{e^u}{e-1} + \frac{e^1 - e^u}{e-1} = \frac{e}{e-1} \approx 1.58 \]
Lower Bound

How good could we get with randomization?

Yao's Principle:

choose an input $d(u)$

if all det. $\text{Alg} \geq r \cdot \text{comp.}$ (on $d(u)$)

$\Rightarrow$ all randomized $\text{Alg} \geq r \cdot \text{comp.}$ (on $d(u)$)

$\Rightarrow$ allows us to devise lower bounds for real $\text{Alg}$ by derandomizing det $\text{Alg}$.
Ski Example

\[ a(0 \leq u \leq 1) = \frac{1}{2} \quad \text{you should rent} \]
\[ a(\infty) = \frac{1}{2} \quad \text{you should have bought} \]

\[ \text{OPT Alg (knows future "offline")} \]

buys if \( u = \infty \), rents otherwise.

\[ \text{cost} = \frac{1}{2} \int_0^1 u \, du + \frac{1}{2} \cdot 1 = \frac{3}{4} \]

\[ r = \frac{CA}{\text{cost}} \]

what's the best det. Alg?

\[ \text{could show best} \quad z = 0 \rightarrow \text{cost} = 1 \]

\[ \Rightarrow r = \frac{\text{best det CA}}{\text{best det CP}} = \frac{4}{3} \approx 1.33 \]

Yao's principle LB of 1.33!

round. 1.58

(∞ one is a little gap)
TCP

Setting: we receive packets ....
every new & clear we have to send ACK
(we'd like to not send an ACK every time
but said .... but also not wait too long)

online problem

metric = # packets
+ latency of each packet

Ack's time
acknowledge up to here
\( z = 4 \)  
**Alg:** as soon as a rect \( n \geq 4 \) send an ACK

**Claim:** OPT sends an ACK before every two ACKs we send

**Claim 2 comp.**

\[
\begin{align*}
\text{OPT: } & \text{keep + latency at } t \text{ } \\
\text{CAlg: } & \text{keep + latency at}
\end{align*}
\]

\[
\text{CAlg} \leq \sqrt{Z} \cdot \text{OPT}
\]
OPT sends ACK btw every two of our Alg. 

$\text{Alg} \Rightarrow \kappa_{\text{Alg}} \leq \kappa_{\text{OPT}}$

$\text{cost}_{\text{Alg}} \leq \sum_{i=1}^{n} \text{packets}_{\text{Alg}}$

$\text{cost}_{\text{Alg}} = \sum_{i=1}^{n} \left( \#\uparrow + \#\square \right)_{\text{opt}}$

$\leq \#\uparrow + \#\square_{\text{opt}}$

$\leq 2 \left( \#\uparrow + \#\square \right)$