# DES Lecture Joël

16. 11. 23

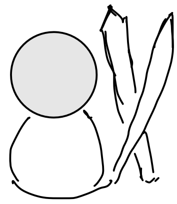
## Recap: Online Algorithms

~ 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

$$\text{cost}_{Alg} \begin{cases} u & u \leq z \quad \text{(always rent)} \\ z+1 & u > z \quad \text{(we buy + rent z days)} \end{cases}$$

$$\text{cost}_{OPT} \begin{cases} u \\ 1 \end{cases} = \min(u, 1)$$

best offline Alg (knows the future)

OA Analysis : Comp. ratio

$$\text{cost}_A \leq r \cdot \text{cost}_{OPT}$$

$$r = \frac{\text{cost}_A}{\text{cost}_{OPT}}$$

Det. Ski Rental 2-comp.

Q: can we do better?

---

so what's the problem?

Adv. knows what we will do → 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)
⇝ " forces Adv. to prepare for multiple outcomes "

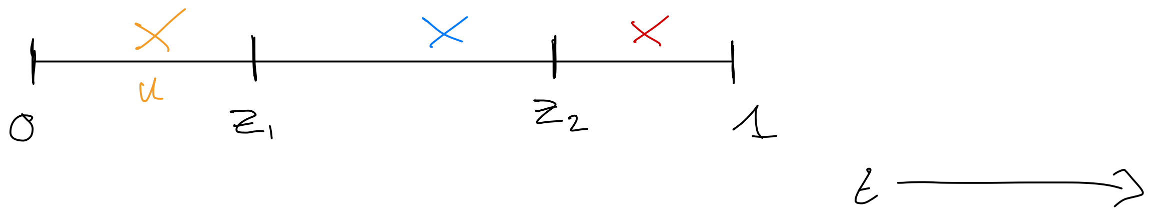
# Randomized Ski Rental

## Approach I

choose $z_1$ with probability $p_1$

else choose

$z_2$ $\quad (p_2 = (1-p_1))$




$$\text{cost}_A \begin{cases} u & u \le z_1 \quad \text{always rent} \\ p_1(z_1+1) + p_2 \cdot u & z_1 < u < z_2 \\ p_1 \cdot (z_1+1) + p_2 \cdot (z_2+1) & z_2 < u \end{cases}$$

Obs: for Adv. it makes sense to either have

$u = z_1 + \varepsilon$ or $z_2 + \varepsilon$

$\sim$ immediately stop skiing after we buy

$z_1 = \frac{1}{2}$ $\quad z_2 = 1$

$u = z_1 + \varepsilon$ $\quad \text{cost}_A = p_1(z_1+1) + p_2 z_1$ $\quad p_1 + \frac{1}{2}$

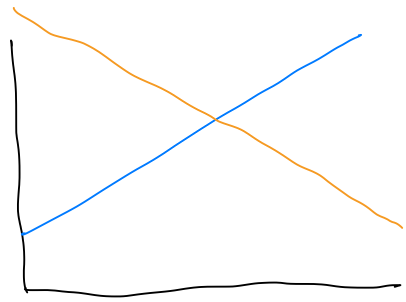
$= p_1 + \frac{1}{2}$

$u = z_2$ $\quad \text{cost}_A = p_1(z_1+1) + p_2(z_2+1)$

$= 2 - \frac{1}{2} p_1$

$z_1 = \frac{1}{2}$ $\quad \dfrac{\text{cost } A}{\text{OPT } (=\frac{1}{2})} = 2p_1 + 1$

$z_1 = 1$ $\quad \dfrac{\text{cost } A}{\text{OPT } (=1)} = 2 - \frac{1}{2} p_1$



$$\Longrightarrow \quad 2p_1 + 1 = 2 - \frac{1}{2} p_1$$

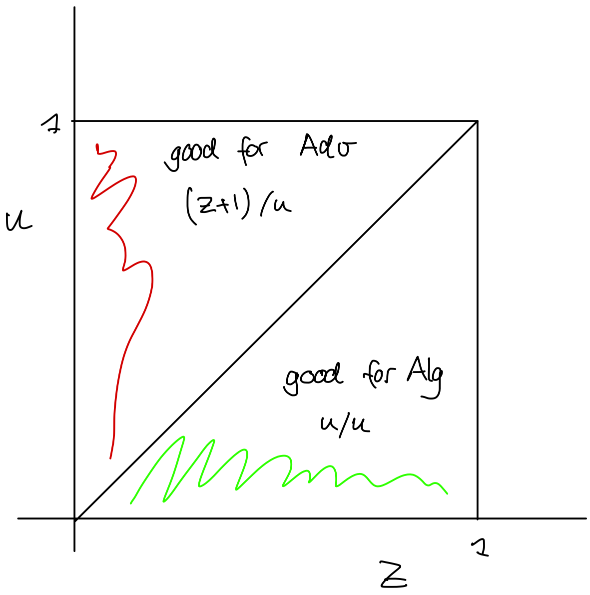
$$\Longrightarrow \quad p_1 = \frac{2}{5}$$

$$\Longrightarrow \quad \frac{\text{cost } A}{\text{cost OPT}} = \frac{9}{5} < 2$$

$\Longrightarrow$ can do better with Randomization!

# Approach 2

- don't choose 2 values
  $\longrightarrow$ INF many ....
  choose a distribution




- now to get our comp. ratio

$p_1$ $p_2$
$z_1$ $z_2$ $p(z)$ $[0,1] \longrightarrow$

$$\frac{\int_0^1\int_0^u (z+1)\, p(z)\, d(u)\, dz\, du + \int_0^1\int_u^1 u \cdot p(z)\, d(u)\, dz\, du}{\int_0^1 u\, d(u)\, du}$$

$\uparrow$ Adv $d(u)$ tries to max.

$\downarrow$ Alg $p(z)$ tries to minimize

- too complex instead recall

$$cost_A(u) \leq r \cdot cost_{OPT}(u) \quad \text{for all } u$$

~ if we can ensure that we do not pay much more than the OPT for any input u (that Adv. can choose) it doesn't matter what Adv. does

$\longrightarrow$ we get r-comp.

so $\quad \text{cost}_z = \begin{cases} u & u \leq z \quad \text{rent} \\ z+1 & u > z \quad \text{buy} \end{cases}$

$$\text{cost}_A(u) \leq r \cdot \text{cost}_{OPT} \; (= u)$$

$$\underbrace{\int_0^u (z+1)\, p(z)\, dz}_{u > z \ \text{buy}} + \underbrace{\int_u^1 u \cdot p(z)\, dz}_{u < z \ \text{rent}} \leq r \cdot u$$

$$\underbrace{\int_0^u (z+1)\, p(z)\, dz}_{u > z \ \text{buy}} + \underbrace{\int_u^1 u \cdot p(z)\, dz}_{u < z \ \text{rent}} \leq r \cdot u$$

math heavy part

$$\underbrace{\int_0^u (z+1)\,p(z)\,dz}_{f} + u\cdot\int_u^1 p(z)\,dz = r\cdot u$$

diff wrt $u$

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

$$\frac{\partial}{\partial u}F(b) - \frac{\partial}{\partial u}(F(a)) =$$

$$F(u) - F(0) \qquad = f(b) - f(a)$$

$$f(u)$$

$$u\cdot\int_a^1 p(z)\,dz$$

$\Rightarrow$

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

$$= p(u) + \int_u^1 p(z)\,dz = r$$

$$\frac{\partial\, p(u)}{\partial u} - p(u) = 0 \quad \Rightarrow p(u) = a\cdot e^u$$

plug in and $a = \frac{1}{e-1}$

$$\frac{e^u}{e-1}$$

$$\Longrightarrow \quad r = p(u) + \int_u^1 p(z)\,dz$$

$$= \frac{e^u}{e-1} + \frac{e^1 - e^u}{e-1} = \frac{e}{e-1}$$

$$\approx 1.58$$

# Lower Bound

How good could we get with randomization?

Input ≠ Algorithm

## Yao's Principle:

choose an input distr $d(u)$

if all det. Alg $\geq$ r-comp. (on $d(u)$)

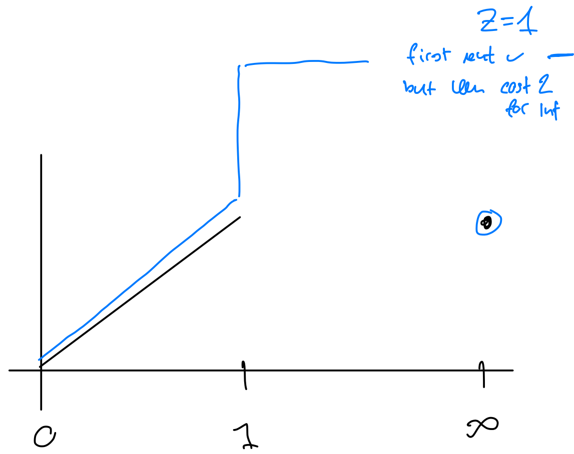
$\Longrightarrow$ all randomized Alg $\geq$ r-comp (on $d(u)$)

$\longrightarrow$ allows us to derive lower bounds for rand. Alg by studying deter.

# Ski Example

$d(0 \le u \le 1) = \frac{1}{2}$ you should rent

$d(\infty) = \frac{1}{2}$ you should have bought




OPT Alg (knows future „offline")

buys if $u=\infty$, rents otw.

$$C_{OPT} = \underbrace{\frac{1}{2}\int_0^1 u\,du}_{\text{rent}} + \underbrace{\frac{1}{2}\cdot 1}_{\text{buy}} = \frac{3}{4}$$

$$r = \frac{C_A}{C_{OPT}}$$

for Yao:

I. $d(u)$ ✓

II. $\forall$ det Alg $\ge r$

what's the best det. Alg?

could show best $z=0 \to$ cost $= 1$

$$\Rightarrow r = \frac{\overset{\text{best det OA}}{1}}{\underset{\text{best OPT}}{\frac{3}{4}}} = \frac{4}{3} \approx 1.33$$

Yao's principle LB of 1.33!

rand. 1.58

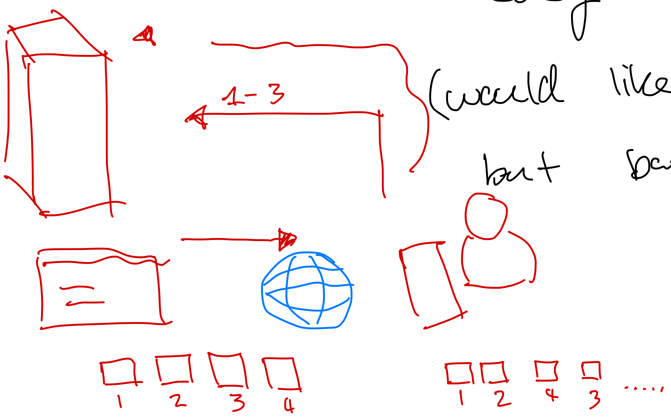
(so there is a little gap)

# TCP

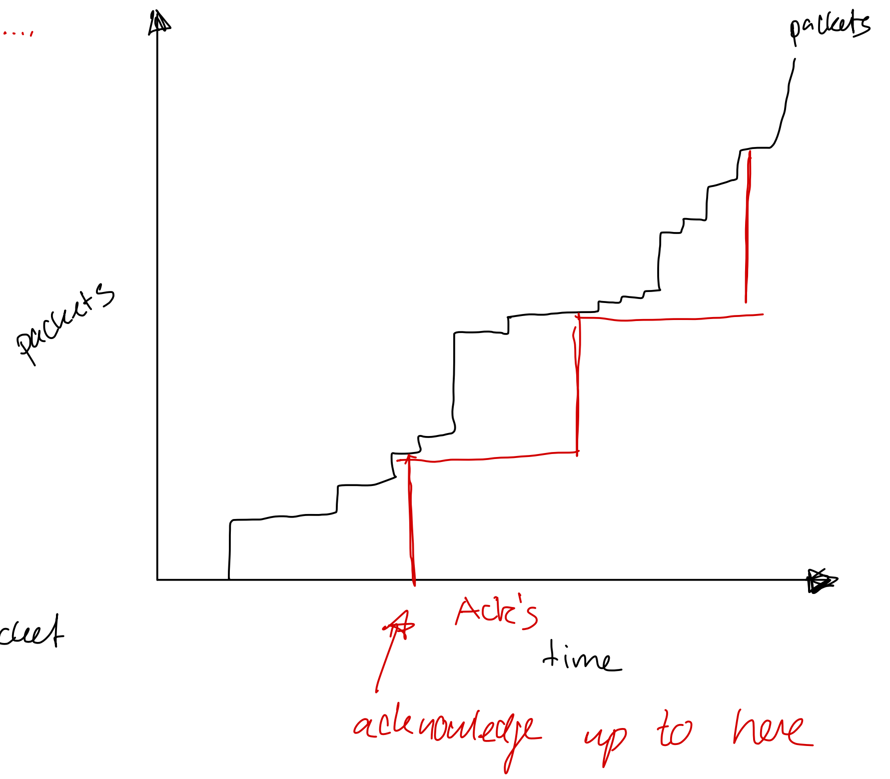
Setting: we receive packets .....

every now & then we have to send ACK

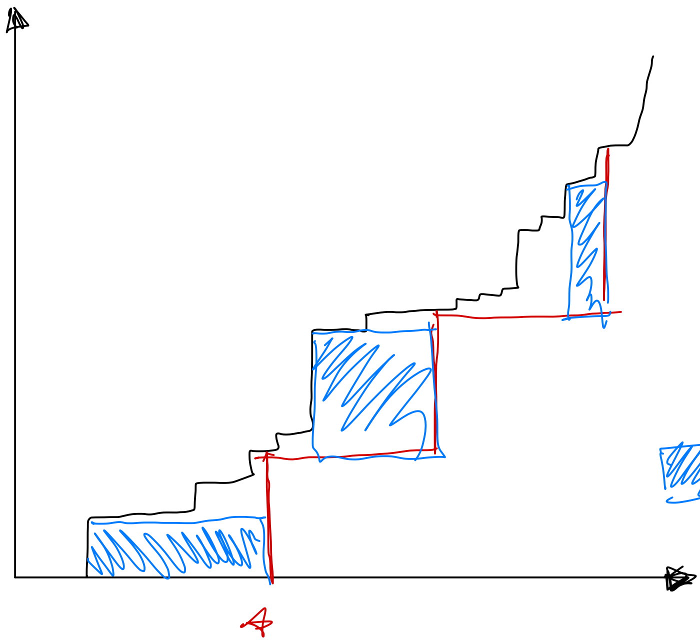
(would like to not send an ACK every time but batch .... but also not wait too long)

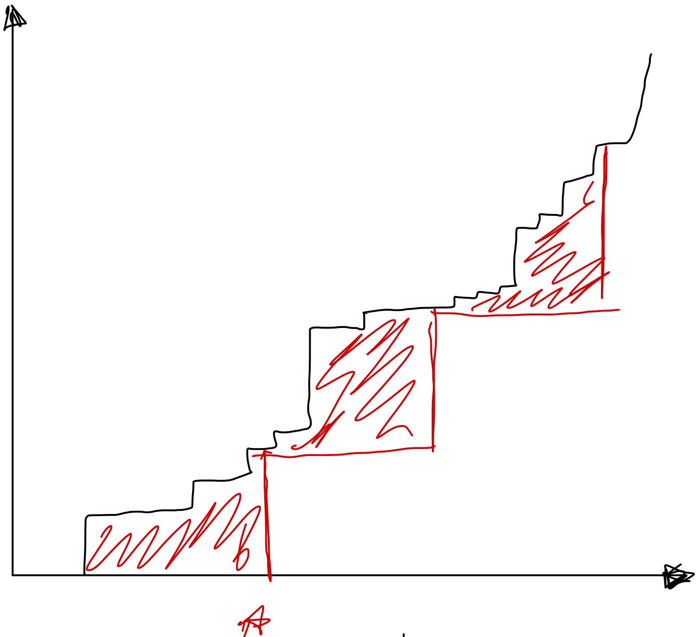



online problem




metric = # packets + latency of each packet





time

acknowledge up to here

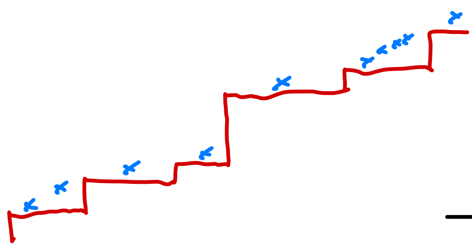
= 1

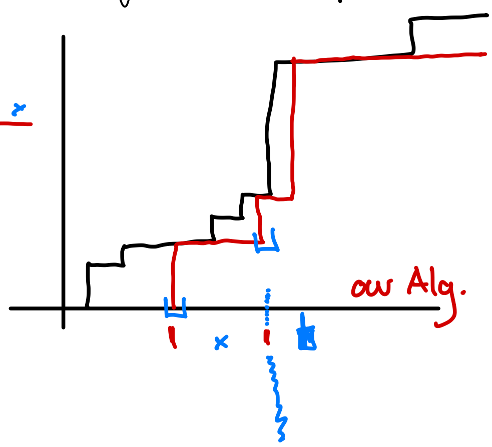
=> send ACK

MIN + #

$z = 1$ Alg : as soon as a rect $\geq 1$ send an ACK

claim : OPT sends an ACK b/w every two ACKs we send
packets






claim 2 comp.

$C_{OPT}$ : $k_{OPT}$ + latency OPT

$C_{ALG}$ : $k_{ALG}$ + latency ALG

$C_{ALG} \leq \boxed{\ } \cdot C_{OPT}$ 2

OPT sends ACK btw every two of our Alg $\Rightarrow$ $k_{ALG} \leq k_{OPT}$