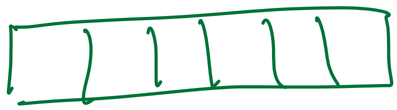
Online Algorithms — Lost Cows and Coffee

Data Structures

Manage a sequence of operations minimize total cost

compared to optimal clairvoyant algorithm

Canonical example: paging



Cache hold k items

Request memory address (x):
  if x in cache
    cost=0
  else
    move x into cache
    eject something else
    cost= 1

LRU + FIFO + FWF are k-competitive

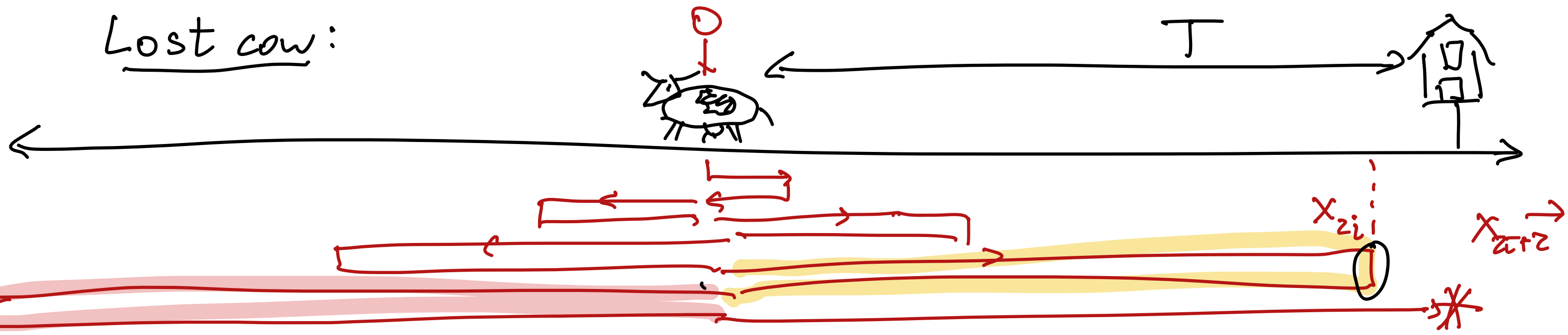
For any access seq, cost incurred is $\le k \cdot OPT$

eject item used again furthest in future

Randomized Marking

$$\frac{E[\#\text{ cache misses}]}{OPT} \le H_k$$

Lost cow:



for $i \leftarrow 0$ to ∞
  walk to x_i (alternate signs)
  walk to 0

If $T \ge 0$, $x_{2i} < T \le x_{2i+2}$

$$dist(T) = 2x_0 + 2x_1 + \cdots 2x_{2i} + 2x_{2i+1} + T$$

Goal: $\min_T \boxed{\max_{x_0 \dots} \frac{dist(x,T)}{T}}$ competitive ratio

Choose $x_0 \dots$ so that $dist(x,T) \le c \cdot T$ for <u>all</u> T

<u>Optimal:</u> $x_i = (-2)^i$

if $2^{2i} < T \le 2^{2i+2}$

then $dist(T) = \underbrace{2+4+8+\dots+2\cdot 2^{2i}+2\cdot 2^{2i+1}}_{2^{2i+3}-2} + T$

$2^{2i+3} - 2 + T < \boxed{9T-2}$

<u>Randomize!</u> <u>Oblivious adversary</u>

with prob 1/2, start +1

1/2 start −1

expand by 2 at all later steps

$dist_R(T) = 2^{2i+3} - 2 + T$ if $2^{2i} < T \le 2^{2i+2}$

$dist_L(T) = 2^{2i+2} - 2 + T$ if $2^{2i-1} < T \le 2^{2i+1}$

if $2^{2i} < T \le 2^{2i+1}$

$E[dist(T)] = \frac{1}{2} dist_R + \frac{1}{2} dist_L$

$= \underline{\frac{1}{2} 2^{2i+3}} + \underline{\frac{1}{2} 2^{2i+2}} + T - 2$

$= 6 \cdot 2^{2i} + T - 2 \le \boxed{7T - 2}$

better:

$b = 1+\sqrt{2}$

$c = 4+2\sqrt{2} \approx 6.28$

[Kao Reif '93]

Fix b

choose δ uniformly between 0 and 1

for $i \leftarrow 0$ to ∞

walk to $(-b)^{i+\delta}$

walk to 0

$E[\text{comp ratio}] \approx 4.61\dots$

optimal!

$1 + \frac{b+1}{\ln b}$

<u>Coffee shop problem:</u>

Every day
either rent for \$1
or buy for \$B all future rents free
but world ends after T days

$OPT = \min(T, B)$

$Cost(i,T)$ = your total cost if you rent i times before buying

$= (i+B)[T>i] + T[T\le i]$

best: $i=B-1$ $\quad \underbrace{Cost(B-1,T)}_{OPT} = (\underbrace{2B-1}_{B})[T\ge B] + \underbrace{T}_{T}[T<B]$

ratio $\boxed{2-1/B}$

<u>Randomize!</u> "rent i times" with probability p_i

$$E[cost(T)] = \sum_{i=0}^{\infty} p_i \cdot cost(i,T)$$

min c

$$\boxed{= \underbrace{\sum_{i=0}^{T-1} p_i(i+B) + \sum_{i=T}^{B-1\ \cancel{\infty}} p_i \cdot T}_{\text{increases with } T} \le c \cdot \underline{}_{OPT}}$$

we want ... $\le c$ ... for all T

Observation 1: Adversary wants $T<B$ or $T=\infty$

Observation 2: Algorithm better if $p_i=0$ for all $i\ge B$

LP with B inequalities +1 = $(\Sigma p_i = 1)$ and B+1 variables $(p_0 \cdots p_{B-1}, c)$

$\downarrow$

Optimal basis has all $p_i>0$ for $i<B$
solve $(B+1)\times(B+1)$ linear system

$$c = \frac{1}{1-(1-\frac{1}{B})^B} < \frac{e}{e-1} \approx 1.58\text{ish}$$