Online Algorithms — Lost Cows and Coffee

Data Structures

Manage a sequence of operations minimize total cost compared to optimal clairvoyant algorithm

Canonical example: paging

```
  1 1 1 1 1 1 1
```

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 + FW are \( k \)-competitive

For any access seq, cost incurred is \( \leq k \cdot OPT \)

Randomized Marking

\[
E \left[ \frac{\# \text{cache misses}}{OPT} \right] \leq H_k
\]

Lost cow:

For \( t = 0 \) to \( \infty \)

- walk to \( x_i \) (alternate signs)
- walk to 0

If \( T \geq 0 \)

\[
\text{dist}(T) = x_0 + x_1 + \ldots + 2x_{z_t} + 2x_{z_t+1} + T
\]
Goal: \( \min_T \max_{x_0} \frac{\text{dist}(x,T)}{T} \)

Choose \( x_0 \) so that \( \text{dist}(x,T) \leq c \cdot T \) for all \( T \)

Optimal: \( x_i = (-2)^i \)

\[
\begin{align*}
\text{dist}(T) &= 2 + 4 + 8 + \cdots + 2 \cdot 2^i + 2 \cdot 2^2 + 2^3 + \cdots + 2^{i+1} + T \leq 2^{i+3} - 2 + T \\
\end{align*}
\]

Randomize!

with prob \( \frac{1}{2} \) start +1

\( \frac{1}{2} \) start -1

expand by 2 at all later steps

\[
\begin{align*}
\text{dist}_R(T) &= 2^{i+3} - 2 + T \\
\text{dist}_L(T) &= 2^i + 2 - 2 + T \leq T \\
\end{align*}
\]

if \( 2^i < T \leq 2^{i+2} \)

if \( 2^{i+1} < T \leq 2^{i+3} \)

Oblivious adversary

Better:

\( b = 1 + \sqrt{2} \)

\( c = 4 + 2\sqrt{2} \approx 6.28 \)

Ecomp ratio \( \approx 4.61 \) -- optimal!

Fix \( b \) uniformly between 0 and 1

for \( i \leq 0 \) to go

walk to \((-b)^i + 8\)

walk to 0

\[
E[\text{comp ratio}] \approx 1 + \frac{b+1}{\ln b}
\]

[Kao, Reif '95]
Coffee shop problem:
Every day
either rent for $1
or buy for $B all future rents free
but world ends after $T$ days

$$\text{OPT} = \min(T, B)$$

Cost($i, T$) = your total cost if you rent $i$ times before buying
$$= (i + B) [T > i] + T [T \leq i]$$

best: $i = B-1$
$$\text{Cost}(B-1, T) = (2B-1) [T > B] + T [T \leq B]$$

$$\text{OPT}$$

ratio $\frac{2 - \frac{1}{B}}{}$

Randomize! “rent i times” with probability $P_i$

$$E[\text{cost}(T)] = \sum_{c=0}^{B} P_i \cdot \text{Cost}(i, T)$$

we want
$$\min C$$

min $C$

for all $T$

increases with $T$

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

Observation 2: Algorithm better if $P_i = 0$ for all $i \geq B$

LP with $B$ inequalities and $B^{2} + 1$ variables

Optimal basis has all $p_i > 0$ for $i < B$

solve $(B+1) \times (B+1)$ linear system

$$C = \frac{1}{1 - (1 - \frac{1}{B})^B} < \frac{e}{e-1} \approx 1.583$$