Randomized Online Algorithms

Online Problems
1) The Paging Problem
2) Server Problem
3) Cat/Mouse Games

Paging Prob

\[ N \text{ pages in slow memory } \quad k < N \quad FM \]

Request seq = \[ V = v_1, v_2, \ldots, v_m \]

Cost model

<table>
<thead>
<tr>
<th>Request</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_i \in \text{FM} )</td>
<td>0</td>
</tr>
<tr>
<td>( v_i \notin \text{FM} )</td>
<td>1 swap in ( v_i ) and evict a page.</td>
</tr>
</tbody>
</table>
On-line Strategy (Deterministic)

LRU = Least Recently Used
(Evict page not used in longest time)

Off-line Strategy (Det)

LFD = Longest Forward Distance
(Evict page not needed for longest time)

Note LRU & LFD are lazy alg.
Eager alg move before needed.
Know Results

Thm. Lazy legs suffice

Thm. LFD is off-line opt

no pts
Recall Metric Space =
1) Set S
2) Distance measure \( d( , ) \)

\[
\forall u \in S \quad d(u, u) = 0
\]

\[
\forall u, v \in S \quad d(u, v) \geq 0
\]

\[
\forall u, v, w \in S \quad d(u, v) + d(v, w) \geq d(u, w)
\] (triangle inequality)


K-server Prob

1) Metric space \( |S| \geq K+1 \)
2) K-servers \( \{h_1, \ldots, h_K\} = H \subseteq S \)
3) Request sequence \( T = T_1, \ldots, T_m \) \( T_i \in S \)

Cost Model

<table>
<thead>
<tr>
<th>request</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_i \in H )</td>
<td>0</td>
</tr>
<tr>
<td>( T_i \notin H )</td>
<td>Move some server ( h_j ) to ( T_i )</td>
</tr>
<tr>
<td>| |</td>
<td></td>
</tr>
<tr>
<td>Cost = ( d(h_j, T_i) )</td>
<td></td>
</tr>
</tbody>
</table>

A 2-server example = 2-headed disk Prob.

\[ \uparrow \quad \uparrow \]

head1    head2

\( d(\text{head}_1, \text{head}_2) = \text{distance} \)
Paging Prob as a $k$-server Prob

$S =$ pages of slow memory

Fast memory as $\subseteq S$

$$d(u,v) = \begin{cases} 0 & \text{if } u = v \\ 1 & \text{otherwise} \end{cases}$$

---

**Known Thms**

**Thm** $\forall k$-server prob the competitive factor $\geq k$.

**Thm** $\forall k$-server probs $\exists 2k$-comp. alg

**Conj** $\forall k$-server probs $\exists k$-comp. alg.
Back to Paging Prob

The competitive factor for LRU versus LFD.

Consider case \( k+1 = N \)

Request \( 1, 2, 3, \ldots, N, 1, 2, 3, \ldots, N, 1, 2, \ldots \)

Note: After request \( 1, \ldots, k \)

LRU has a page fault per request.

While

LFD has a page fault every \((k-1)\)th request.

<table>
<thead>
<tr>
<th>FM</th>
<th>requests</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [1, \ldots, N-1] )</td>
<td>1, 2, \ldots, N-1</td>
<td>0</td>
</tr>
<tr>
<td>( [1, \ldots, N-2, N] )</td>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>( [1, \ldots, N-2, N] )</td>
<td>1, \ldots, N-2</td>
<td>0</td>
</tr>
<tr>
<td>( [1, \ldots, N-3, N-1, N] )</td>
<td>N-1</td>
<td>0</td>
</tr>
<tr>
<td>( [1, \ldots, N-3, N-1, N] )</td>
<td>N, 1, \ldots, N-3</td>
<td>7</td>
</tr>
<tr>
<td>( [1, \ldots, N-4, N-2, N-2, N] )</td>
<td>N-2</td>
<td>1</td>
</tr>
</tbody>
</table>
Thus LRU is at most \((k-1)\)-competitive.

Goal: Get better cont-factor using randomization.

Def: A randomized alg \(A\) is \(c\)-competitive if there exists a constant \(c\) such that for all instances \(I\) and all offline alg \(B\),

\[
\text{Expect}[C_A(I)] \leq c \cdot C_B(I) + \alpha
\]

We will show

\(\exists\) paging alg (randomized) that is \(O(\log N)\)-Competitive.
Yet Another online Prob.

The Cat/Mouse Game

1) 1 Cat & 1 Mouse
2) N-hiding places

Cat = seq of probes looking for mouse

Cost = \(\begin{cases} 1 & \text{if mouse found} \\ 0 & \text{o.w.} \end{cases}\)

Note Cat/Mouse just Paging with \(K+1 = N\)

Question: Find a good randomized strategy for mouse

First Try

RAND: If found move to random new home.
Claim: RAND not good!

Suppose

Cat visits: 1, 2, ..., N-1, 2, 3, ..., N-1, 1

She does not probe N.

Opt off-line: Mouse moves to N

Total cost is 1.

RAND:

1) If at N, cost is 0
2) If not at N

What is expected # of moves to land at N?

This is the same as:

Expect # of rolls of an N-sided die
to get, say, N.
Let $E$ be the expected number of rolls.

$E$ satisfies recurrence:

$$E = \frac{1}{N} (1) + \left( \frac{N-1}{N} \right) (1 + E)$$

$$= 1 + \left( \frac{N-1}{N} \right) E$$

$$\frac{1}{N} E = 1 \Rightarrow E = N$$

Back to RAND

$\text{Expect}[\text{RAND}] = N$

Thus RAND is $\mathcal{O}(N)$-competitive!
Claim: All alg are $\mathcal{O}(\log N)$-Competitive.

Pf

Cas Alg: Probe randomly for $t$ times

where $t = N \log N$.

On-line Alg: Expected cost = $t/N$

thus $= N \log N / N = \log N$

Off-line = looking into future for a place to hide!

Question: # of probes for cat to inspect every square?

Let $X$ be a random variable = # probes.
Let $P_i = \text{Prob of seeing a new sq after seeing i squares.}$

Thus $P_i = \frac{N-i}{N}$

Let $X_i = \text{Random variable \# of probes to see a new sq after seeing i sq.}$

Note $X = \sum_{i=0}^{N-1} X_i$ \& $E(X_i) = \frac{N}{N-i}$

Thus

$$E(X) = \sum_{i=0}^{N-1} E(X_i) = \sum_{i=0}^{N-1} \frac{N}{N-i} = N \sum_{i=1}^{N} \frac{1}{i}$$

$$= \Theta(N \log N)$$

Expect cost of off-line = $O(1)$

$\therefore \Omega(\log N)$ - Competitive
MARKING: 1) Start at random place.
2) Mark each probed place.
3) When found move to random unmarked place.
4) When all places marked unmark and restart.

Claim: Marking is $O(\log N)$-competitive.

Def: Phase = time from a restart to a restart.
2 Types of probes
1) probing marked spot (no cost)
2) probing unmarked spot

Since Cat knows your strategy no type 1 probes.

Let \( M_i \) = \# of moves per phase.
(random variable)

\( M_i = \begin{cases} 1 & \text{if found at probe } i, \\ 0 & \text{o.w.} \end{cases} \)

\[ M = \sum M_i \& E(M_i) = \frac{1}{N-i+1} \]

Thus \( E(M) = \sum E(M_i) = \Theta(\log N) \)

Since every place probed per phase
Opt \( \geq 2 \)

Thus Marking is \( O(\log N) \) - Competitive
Marking for Pages

Init = no marks

1) Mark each requested page.
2) Eject a random unmarked page.
3) When all page in fast memory are marked restart.

Known: This alg is 2Hk - Competitive