#### §6 Competitive Analysis of Online Algorithms

- Motivation: Ski Rental
  - Break-Even Algorithm
  - is 2-competitive; optimality
- Online Paging
  - *Least-Recently Used* is *k*-competitive
  - Least-Frequently Used is not competitive
  - LRU is optimal among deterministic online
- Randomization and expected competitiveness
  - 1.84-competitive randomized Ski Rental









**Competitive ratio** = online cost / offline cost

#### Ski Rental Math Quiz Design & Analysis of Algorithms Martin Ziegler

 $L / \min(L,D)$  :  $L \le D$ 2D-1 /  $\min(L,D)$  :  $L \ge D$ 

**a)** Prove:  $\leq (2-1/D)$  either case

**c)** Is 
$$\frac{X+D}{\min(X+1,D)} \geq 2$$

for every  $X \in \mathbb{N}$  ?

 $\frac{X+D}{\min(X+1,D)} \ge 2-1/D$ for every  $X \in \mathbb{N}$ 

### **Online Paging**

*k* pages of *fast* memory, caching  $K \gg k$  *slow* pages. For any sequence  $\underline{a} = a_1, ..., a_N \in \{1, ..., K\}$  of accesses, minimize the number of cache misses/load/evictions.



Input revealed gradually; *online* algorithm must makes decisions with <u>partial</u> knowledge. Analyze *online* algorithm's <u>output</u> in comparison to optimal *offline* algorithm: **competitive ratio**.

# Online Paging: <u>LRU</u>

**Theorem:** LRU has competitive ratio k = #pages **Proof:** Compare LRU to optimal offline algorithm  $\mathcal{A}$ , started with *same* initial cache contents.

 $\begin{array}{c} & & & & \\ a_1 & & & \\ a_2 & & \\ \end{array}$ Divide 1,...N into rounds  $1 < t_0 < t_1 < ... < t_M = N$  s.t. LRU incurs precisely k faults in  $(t_{m-1}...t_m]$  and [1...k] faults in  $[1...t_1]$ . In each round,  $\geq k+1$  pages get accessed; fault hence  $\mathcal{A}$  incurs at least 1 page fault! Whenever a new page is accessed, evict the one Least Recently Used.

Analyze *online* algorithm <u>output</u> in comparison to the *offline* optimum: **competitive ratio**.

## Online Paging: <u>LFU</u>

**Theorem:** LFU has no (finite) competitive ratio!

**Proof:** Compare LFU to the optimal online algorithm on the following access sequence for K=k+1:



Analyze *online* algorithm <u>output</u> in comparison to the *offline* optimum: **competitive ratio**.

### **Optimality in Online Paging**

**Theorem:** Every deterministic online algorithm  $\mathcal{A}$  has competitive ratio  $\geq k = \#$  pages

**Proof:** Let K > k. Simulate  $\mathcal{A}$  on <u>initial</u> access sequence  $\underline{a} = (1, 2, ..., k)$ . Pigeonhole: choose  $\widehat{a_{n+1}} \in \{1, ..., K\}$  <u>not</u>

in  $\mathcal{A}$ 's cache after serving  $(a_1, a_2, \dots, a_n)$ .

 $\mathcal{B}$  faults only every *k*-th request!

"adversary"

 $\mathcal{A}$  faults on

every request!

Pigeonhole: Omnicient *offline* algorithm  $\mathcal{B}$  (can) choose to evict a page <u>not</u> to be accessed in the <u>next</u> k-1 steps.

Analyze *online* algorithm <u>output</u> in comparison to the *offline* optimum: **competitive ratio**.

#### **Randomized** Online Algorithm

Design & Analysis of Algorithms Martin Ziegler

Each morning: (i) **Rent** at \$1 for <u>another</u> day or (ii) **buy** <u>once</u> for \$*D*>1

*Breakeven* is (2–1/*D*)-competitive, and best possible:

Fix <u>any</u> algorithm  $\mathcal{A}$ , run on  $\infty$  season, let X=# days it rents before buying. Restart, abort season on day #X+1.

**Randomized** Ski Rental: Flip a fair coin. **Head:** Breakeven (  $\approx$  rent for D days, then buy) **Tail:** Rent for  $\frac{2}{3}D$  days, then buy. 1.833

 $L \ge D: \mathbb{E}[\cos t] \approx \frac{1}{2} \cdot (2D) + \frac{1}{2} \cdot (\frac{2}{3} + 1) \cdot D = \frac{11}{6} \cdot D$  $\frac{2}{3}D \le L < D: \mathbb{E}[\cos t] \approx \frac{1}{2} \cdot (L) + \frac{1}{2} \cdot (\frac{2}{3} + 1) \cdot D \le \frac{(\frac{1}{2} + \frac{1}{2} \cdot (\frac{2}{3} + 1)/\frac{2}{3}) \cdot L}{L < \frac{2}{3}D: \mathbb{E}[\cos t] \approx \frac{1}{2} \cdot (L) + \frac{1}{2} \cdot (L) = L$ 1.75

"Randomization can beat an adversary!"

### §6 Summary

Design & Analysis of Algorithms Martin Ziegler

- Motivation: Ski Rental
  - Break-Even Algorithm
  - is 2-competitive; optimality
- Online Paging
  - *Least-Recently Used* is *k*-competitive
  - Least-Frequently Used is not competitive
  - LRU is optimal among deterministic online
- Randomization and expected competitiveness
  - 1.84-competitive randomized Ski Rental