SAARLAND UNIVERSITY

COMPUTER SCIENCE

On the Smoothness of Paging Algorithms

Jan Reineke and Alejandro Salinger

Technische Universität Dortmund

November 17, 2016

Motivation: Real-time Systems

Side airbag in car
Reaction in < 10 ms

Crankshaft-synchronous tasks
Reaction in < 45 microsec

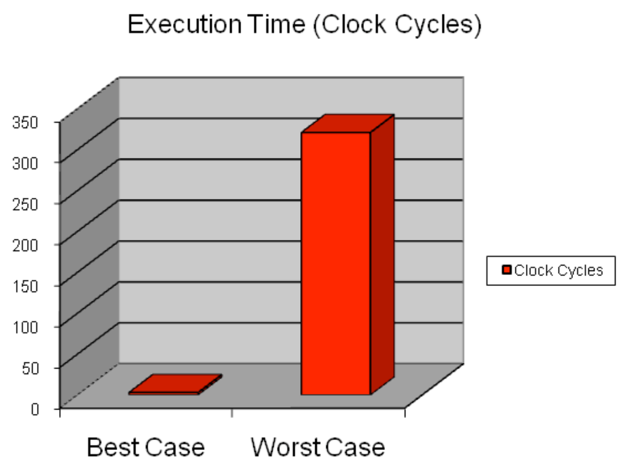
Controllers must finish their tasks within given time bounds

→ Need to determine Worst-Case Response Times (WCRT)

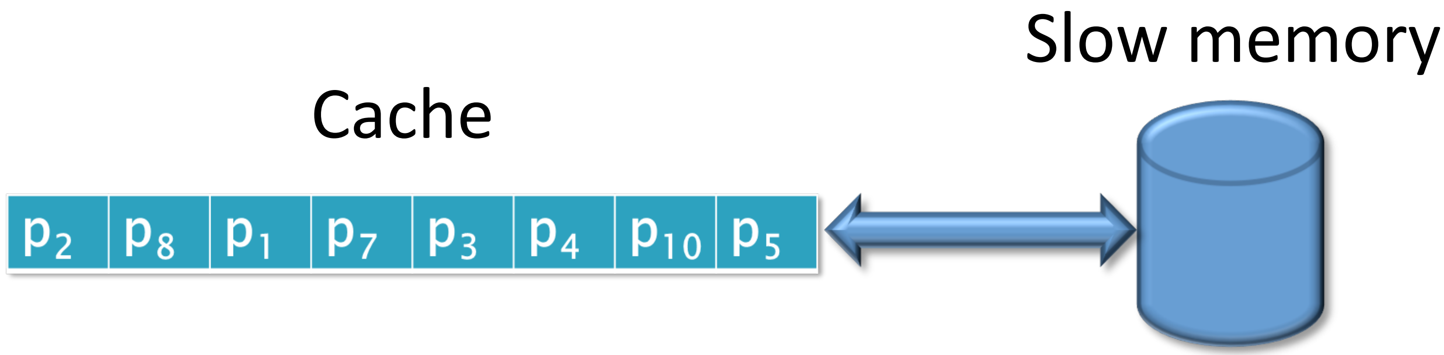
Influence of Caches on Execution Times

```
x=a+b;  →  LOAD   r2, _a
           LOAD   r1, _b
           ADD    r3,r2,r1
```

PowerPC 755



Basics: Caching/Paging



$\sigma = \ldots p_6\, p_3\, p_2\, p_4\, p_4\, p_2\, p_{10}\, p_{11}\, p_5\, p_4 \ldots$

Is p_i in the cache?

-Yes Hit

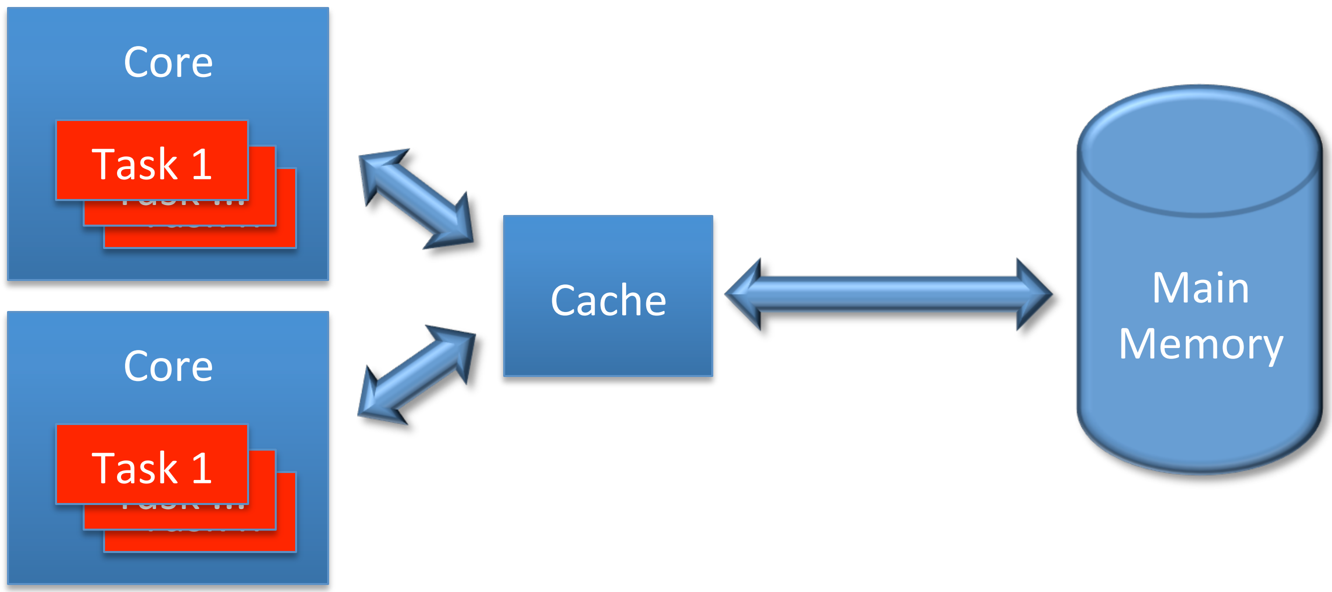
-No Fault (miss)

Fetch p_i from slow memory, evict one page from cache

→ Replacement policy determines page to evict

→ Access sequence + policy determine cache state

Why are Caches a Challenge?



1. Input-dependent memory accesses
2. Interference due to preempted tasks
3. Interference due to co-running tasks

Two Approaches to Timing Analysis

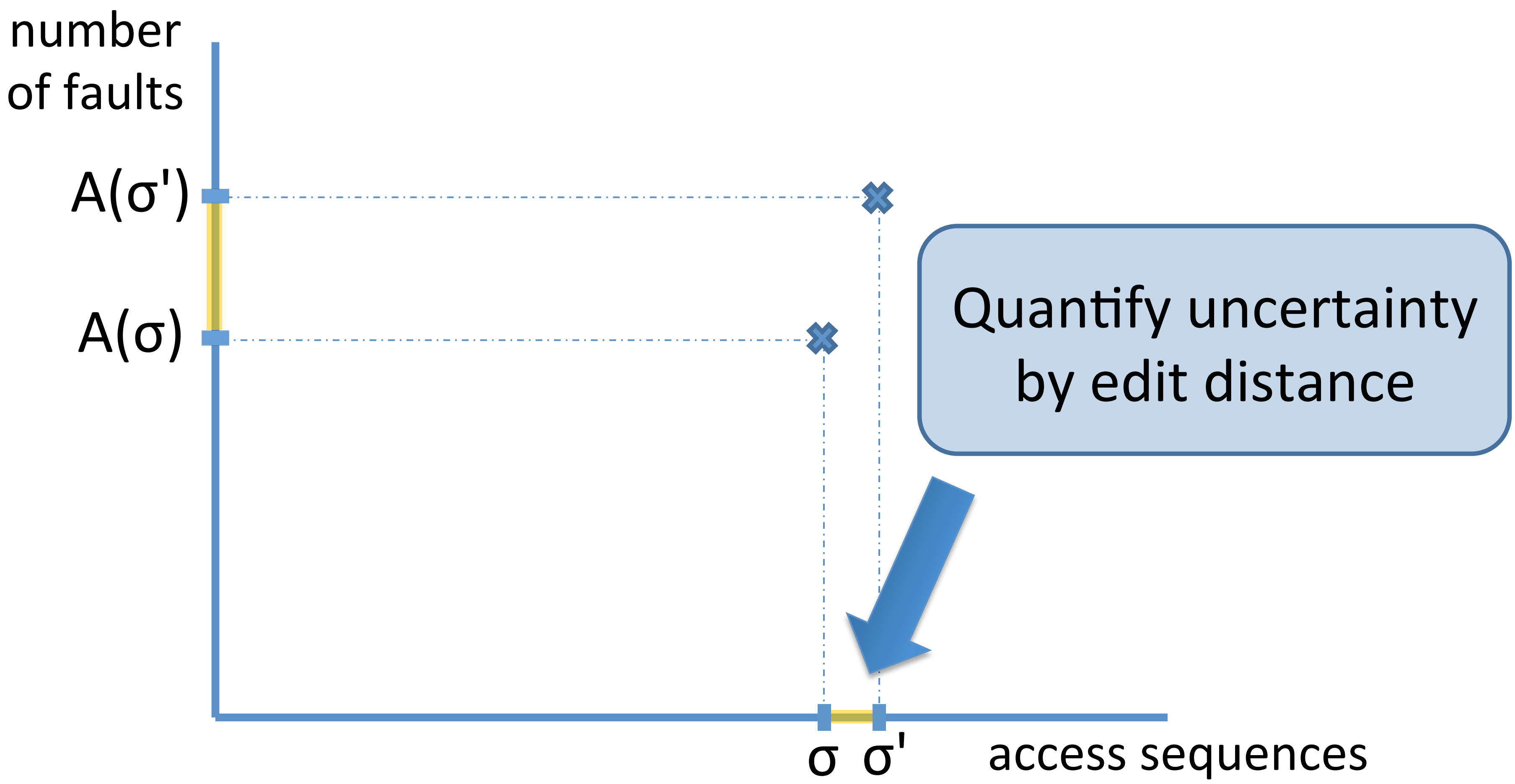
1. Static Analysis:

How does uncertainty about memory accesses affect uncertainty about number of faults?

2. Measurement-based Analysis:

How representative are measurements on a subset of the possible cases?

How does uncertainty about memory accesses affect uncertainty about execution time?



Definition of Smoothness

$dist(\sigma, \sigma')$ = edit distance between σ and σ'

An online algorithm A is (α, β, δ)-smooth if for all σ, σ' with $\mathrm{dist}(\sigma, \sigma') \leq \delta$

$$A(\sigma') \leq \alpha \cdot A(\sigma) + \beta$$

For generic δ:

An online algorithm A is (α, β)-smooth if for all σ, σ'

$$A(\sigma') \leq \alpha(\delta) \cdot A(\sigma) + \beta(\delta)$$

where α and β are functions and $\mathrm{dist}(\sigma, \sigma') \leq \delta$

We call A **smooth** if it is (1,β,1)-smooth for some β.

Key Questions

- How smooth are known paging algorithms?
- Are there fundamental bounds on the smoothness of paging algorithms?
- Are smoothness and high performance contradictory goals?
- Can randomization help?

Deterministic Replacement Policies / Paging Algorithms

Online Algorithms:

- LRU: Least-Recently-Used
- FIFO: First-In-First-Out
- FWF: Flush-When-Full
- …

Offline Algorithm:

- FITF: Furthest-In-The-Future (OPT, LFD, Belady's)

Smoothness of LRU

$\sigma = \ldots a\ b\ c\ d\ a\ b\ c\ d\ \ldots$

$\sigma' = \ldots a\ b\ c\ d\ p\ a\ b\ c\ d\ \ldots$

$dist(\sigma, \sigma') = 1$

$LRU(\sigma)$ [d c b a] → [d c b a] → [a d c b] → [b a d c]

$LRU(\sigma')$ [d c b a] $\xrightarrow{p}$ [p d c b] $\xrightarrow{a}$ [a p d c] $\xrightarrow{b}$ [b a p d]

miss miss miss

→ [c b a d] → [d c b a]

$\xrightarrow{c}$ [c b a p] $\xrightarrow{d}$ [d c b a]

miss miss

Same state

$$LRU(\sigma') = LRU(\sigma) + 5$$

Smoothness of LRU

$\forall\ \sigma, \sigma'$ with $dist(\sigma, \sigma') = 1$

$$LRU(\sigma') \leq LRU(\sigma) + (k+1)$$

Proof sketch

- Age of page: number of distinct requests since last request
- A request is a miss if age is $\geq\ k$
- At all times at most k pages have age $< k$
- New page p can increase age of at most k requests from $k-1$ to k

$\Rightarrow$ At most $(k+1)$ extra misses

What about $\delta > 1$?

For any replacement policy A:

If A is (1,c,1)-smooth, then A is (1,δc)-smooth

$LRU(\sigma') \leq LRU(\sigma) + (k+1)$ for σ, σ' with $dist(\sigma, \sigma') = 1$

$\Downarrow$

$LRU(\sigma') \leq LRU(\sigma) + \delta(k+1)$ for σ, σ' with $dist(\sigma, \sigma') \leq \delta$

LRU is $(1, \delta(k+1))$-smooth

FIFO is not smooth

There exist σ and σ' with $dist(\sigma, \sigma') = 1$ such that caches of $FIFO(\sigma)$ and $FIFO(\sigma')$ are reversed

miss

$FIFO(\sigma)$ [a b c d] → [e a b c] → [e a b c] → [e a b c]

e a b

$FIFO(\sigma')$ [d c b a] → [e d c b] → [a e d c] → [b a e d]

miss miss miss

miss

→ [e a b c] → [d e a b] → [d e a b] → [d e a b]

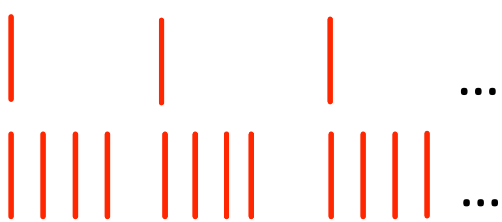
c d e a …

→ [c b a e] → [d c b a] → [e d c b] → [a e d c]

miss miss miss miss

$FIFO(\sigma) =$

$FIFO(\sigma') =$



$$FIFO(\sigma') \geq k \cdot FIFO(\sigma)$$

Lower Bounds (1 of 2)

- An algorithm is *demand paging* if it only evicts pages when needed
  - e.g., LRU, FIFO

No deterministic, demand-paging algorithm is better than $\big(1, \delta(k+1)\big)$-smooth

- But not all algorithms are demand paging:

How about competitive algorithms?

Competitive Analysis

- $A(\sigma)$: number of misses of A on sequence σ
- An online algorithm A is r-competitive if for all σ

$$A(\sigma) \leq r \cdot OPT(\sigma) + \beta$$

- Minimum such r is A's competitive ratio $CR(\text{A})$
- $CR(LRU) = CR(FIFO) = CR(FWF) = k$
- Randomized algorithms can achieve CR in $O(\log k)$

Lower Bounds (2 of 2)

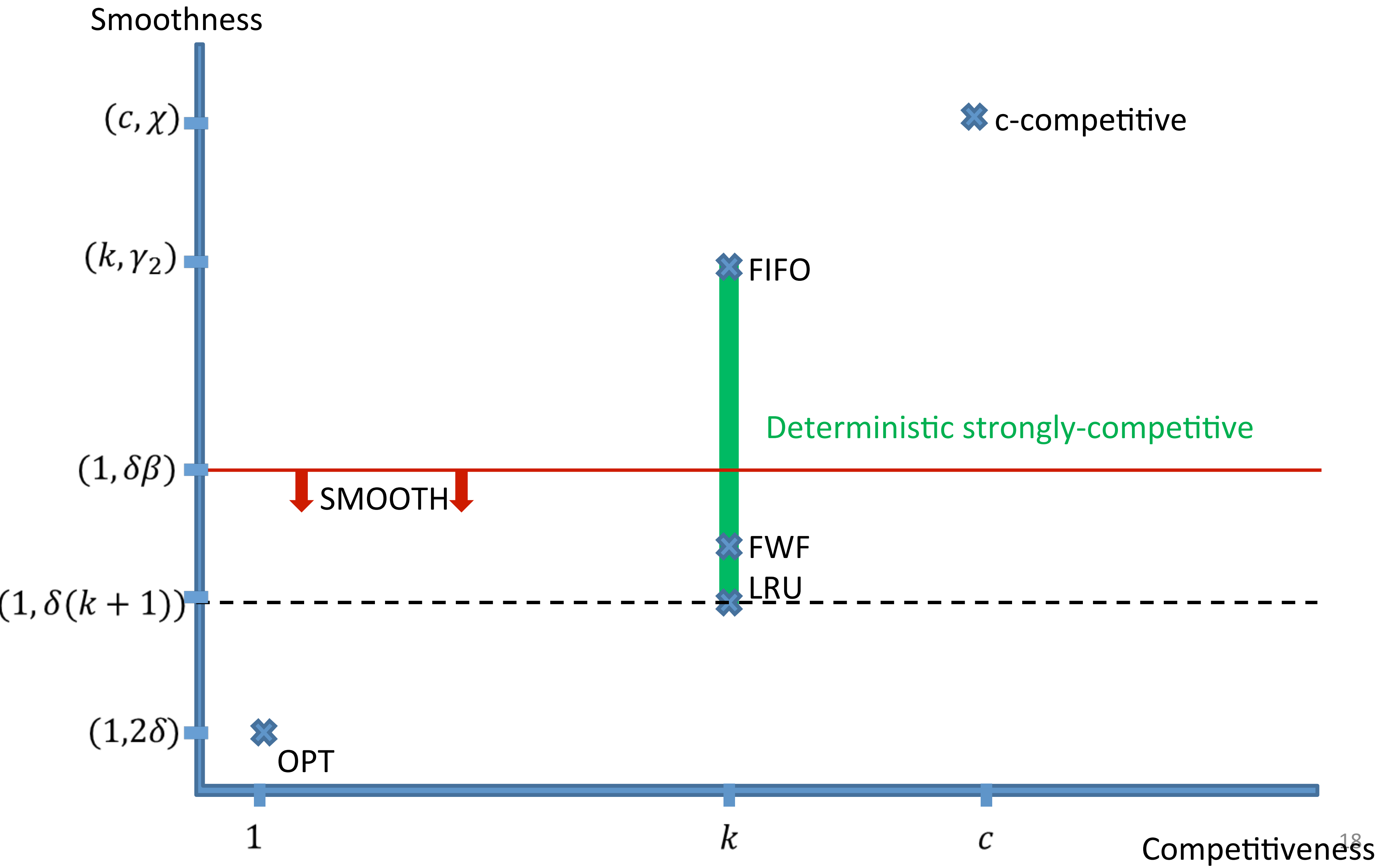
- An algorithm is *demand paging* if it only evicts pages when needed
  - e.g., LRU, FIFO

No deterministic, demand-paging algorithm is better than $\big(1, \delta(k+1)\big)$-smooth

- But not all algorithms are demand paging
  - e.g., FWF

No deterministic, competitive algorithm is better than $\big(1, \delta(k+1)\big)$-smooth

Deterministic Algorithms



Can randomization help?

Randomized Replacement Policies

- RAND: Evict page chosen uniformly at random

- MARK: Evict only "unmarked" pages
- PARTITION, EQUITABLE: define state probability distribution based on OPT's cache contents

- Evict-On-Access: like RAND, but evict on hits too!

Randomized Replacement Policies

Algorithm	Competitive ratio	Smoothness	
RANDOM	k	$(1, \delta(k+1))$	
MARK	$2H_k - 1$	$(\Theta(H_k), \beta)$	
PARTITION	H_k	$(1+\epsilon, \beta, 1)$	$(H_k, 2H_k)$
EQUITABLE	H_k	$(1+\epsilon, \beta, 1)$	$(H_k, 2H_k)$
EOA	∞	$\left(1, \left(1 + \frac{k}{2k-1}\right)\delta\right)$	

$$H_k = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}$$

No randomized, demand-paging algorithm is better than $\left(1, H_k + \frac{1}{k}, 1\right)$-smooth

No randomized, **strongly**-competitive algorithm is better than $(1, \delta H_k)$-smooth

RANDOM is $(1, \delta(k+1))$-smooth

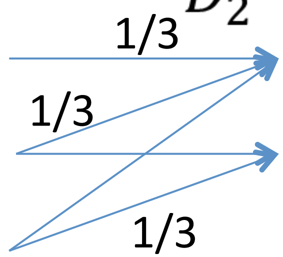
- Bound extra misses by distance between distributions

D_1 D_2

[a b c] → 1/3 [a b c] → 2/3

[a b d] → 1/3 [a b e] → 1/3

[a c d] → 1/3



$$\Delta(D_1, D_2) = \min_{\alpha} \sum_{s_1 \in D_1, s_2 \in D_2} \alpha(s_1, s_2) \cdot cost(s_1, s_2)$$

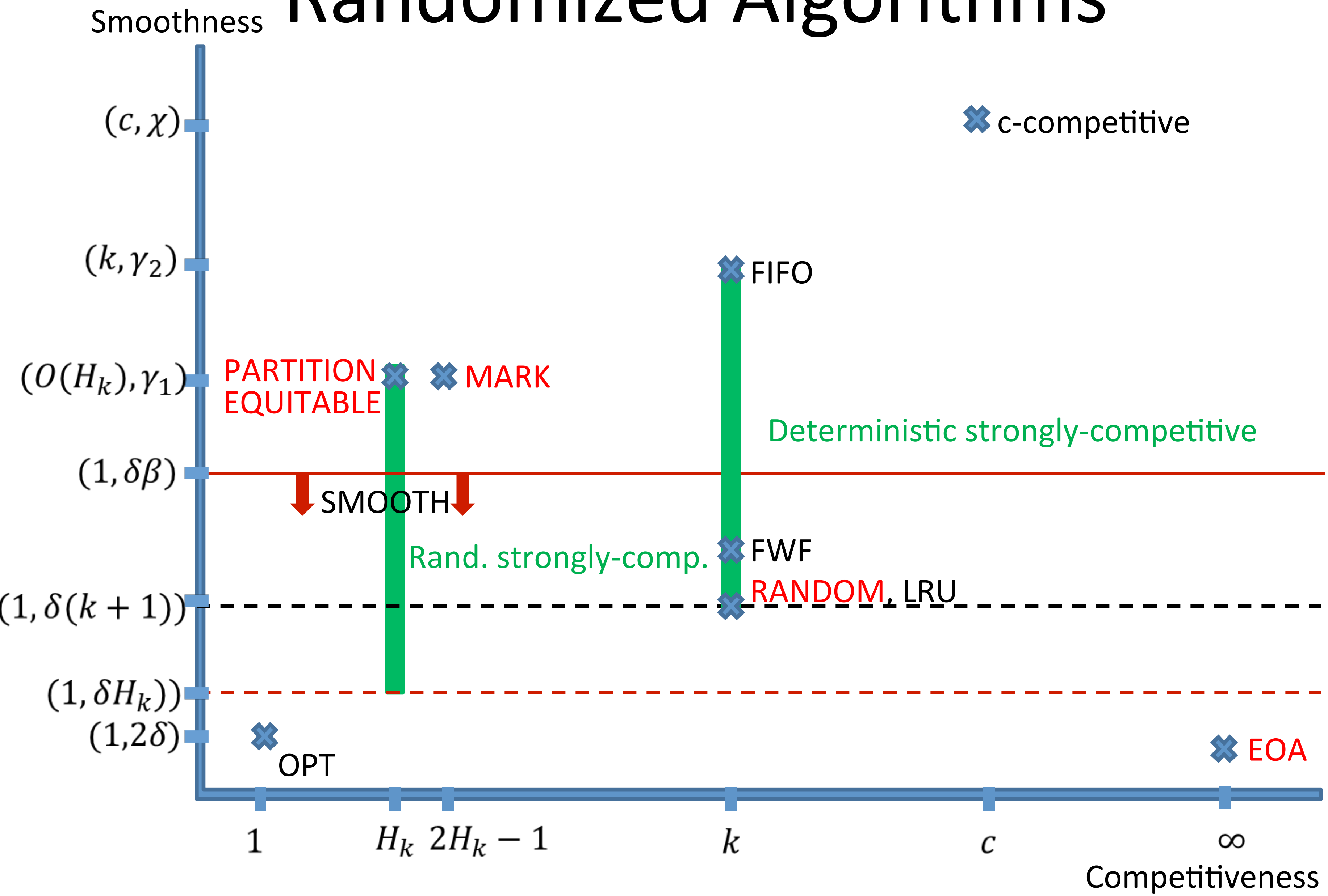
$cost(s_1, s_2) = kH_c$ where $c = |s_1 \backslash s_2|$

2 Claims:

D and D' are distributions resulting from serving ρ and ρ' with $\text{dist}(\rho, \rho') = 1$. Then, $\Delta(D, D') \leq k$

$$\forall \sigma, RAND(D_1, \sigma) - RAND(D_2, \sigma) \leq \Delta(D_1, D_2)$$

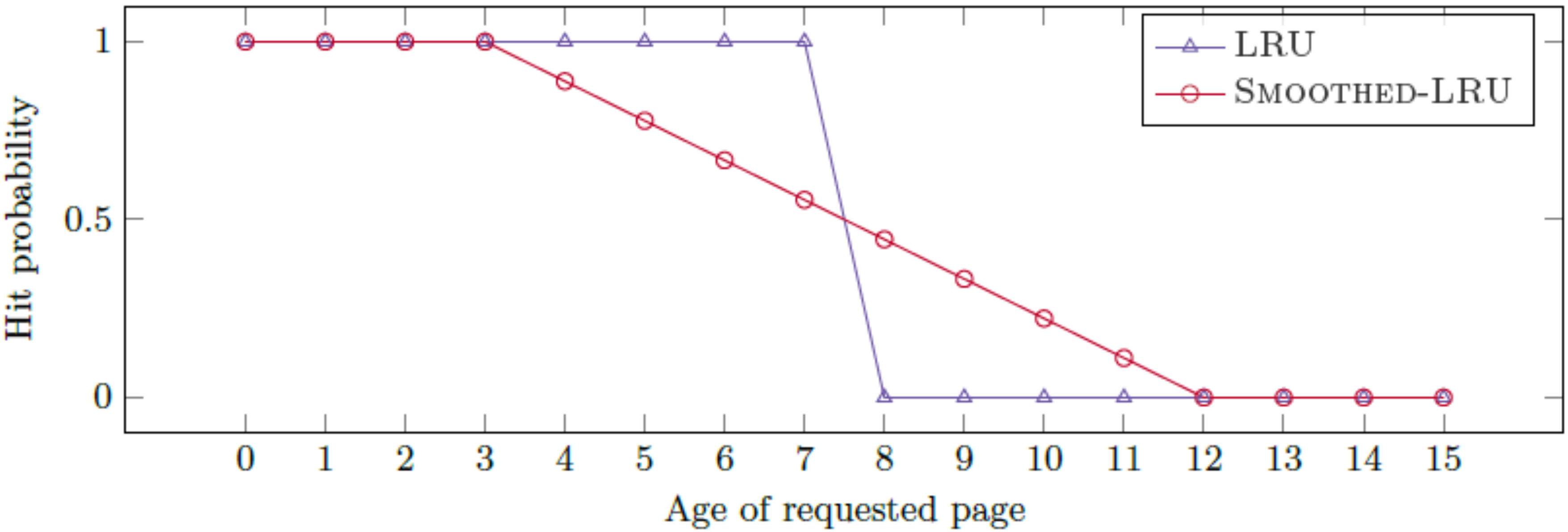
Randomized Algorithms



Can we design smoother algorithms?

Smoothed-LRU

- Recall the age of a page in LRU's cache: hit if age $< k$
- Idea: smooth this transition



$k = 8$
$i = 4$

Smoothed-LRU

Smoothed-LRU is $(1, \delta \cdot \min(\frac{2k-1}{2i+1}+1, \frac{k+i-1}{2i+1}+2))$-smooth

- i = 0 → as smooth as LRU
- i = k-1 → as smooth as OPT

Is it competitive?

For any sequence σ and $l \leq k-i$:

$$\text{Smoothed-LRU}_{k,i}(\sigma) \leq \frac{k-i}{k-i-l+1} \cdot \mathrm{OPT}_l(\sigma) + l$$

As competitive as LRU for size k-i.

LRU-Random

- Smoothed-LRU and EOA are very smooth, but not competitive

Is there a "reasonable" policy that beats the additive $\delta(k+1)$ lower bound?

- LRU-Random: evict the i^{th} oldest page with probability $\frac{1}{iH_k}$

LRU-Random is k-competitive (against adaptive adversary)

LRU-Random is $\left(1, \left(1 + \frac{11}{6}\right)\delta\right)$-smooth for $k = 2$

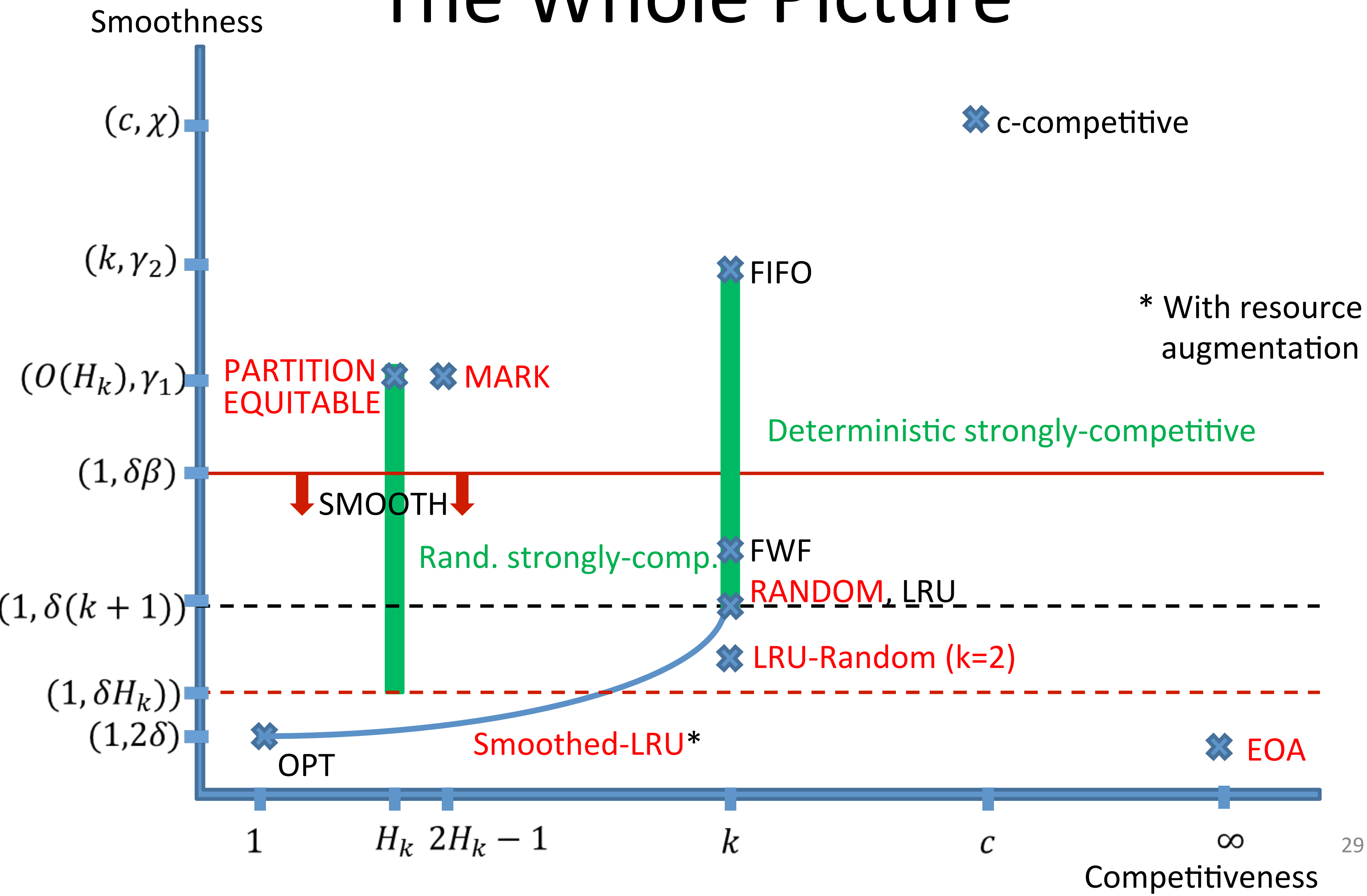
$$1 + 11/6 = 2.8\overline{3} < 3 = k + 1$$

LRU-Random: Conjectures

LRU-Random is $(1, \Theta(H_k^2)\ \delta)$-smooth

LRU-Random is $\Theta(H_k^2)$-competitive against an oblivious adversary

The Whole Picture



Open Problems

- (Generalize smoothness proof for LRU-Random)
- Is there a randomized „LRU-sibling“?

- Are there randomized algorithms that are smooth „with high probability“?

- Are there „less pessimistic“ notions of smoothness?

Thank you for your attention!