Beyond Competitive Analysis: Loose Competitiveness

Arash Pourdamghani
Networking Theory Seminar
Summer 2021
University of Vienna

Competitive Analysis

- An online Algorithm ALG against offline algorithm OPT on an input sequence σ.

$$\inf_{ALG}\left(sup_{\sigma} \frac{ALG(\sigma)}{OPT(\sigma)} \right)$$

- Disadvantages of basic approach:
  - On all possible inputs, even unrealistic ones
  - With an OPT having full-knowledge about ALG and future
  - No ranking between algorithms with same ratio

Beyond Completive Analysis

- Limiting the input:

$$\inf_{ALG}\left(sup_{\sigma^*} \frac{ALG(\sigma^*)}{OPT(\sigma^*)} \right)$$

Examples:

  - **Locality of Reference:** A recent item in σ^* appears again soon
  - **Access Graph Model:** Items in σ^* describe a walk in a graph
  - **Stochastic Model:** σ^* comes from a priorly known distribution

Beyond Completive Analysis

- Empowering the algorithm:

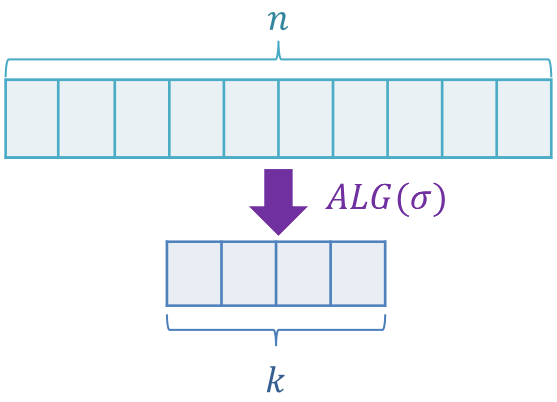
$$\inf_{ALG}\left(sup_{\sigma} \frac{ALG^*(\sigma)}{OPT(\sigma)} \right)$$

Examples:

- **Randomness:** ALG^* has a random coin (that OPT doesn't know about)
- **Advice:** An expert gives ALG^* information about future
- **Augmentation:** ALG^* has additional resources in comparison to OPT

Paging

- Large slow memory with size n, small cache with size k
- Items with uniform size, uniform cost of moving
- Sequence $\sigma = (\sigma_1, \dots, \sigma_m)$ of page requests
- Cost of an algorithm: number of *page faults*
- Algorithms:
  - Furthest in Future(FIF)
  - Marking algorithms: Least Recently Used(LRU)

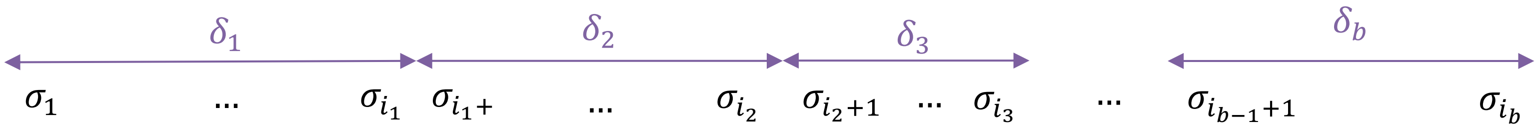


Lower Bound

- **Theorem 1:** Competitive ratio of any deterministic online paging algorithm is at least k.
    - Consider a sequence of $k + 1$ elements, at each point in time request page that is not in cache!
- Increasing cache size also increases the competitive ratio!
- 100% cache fault rate is unavoidable!

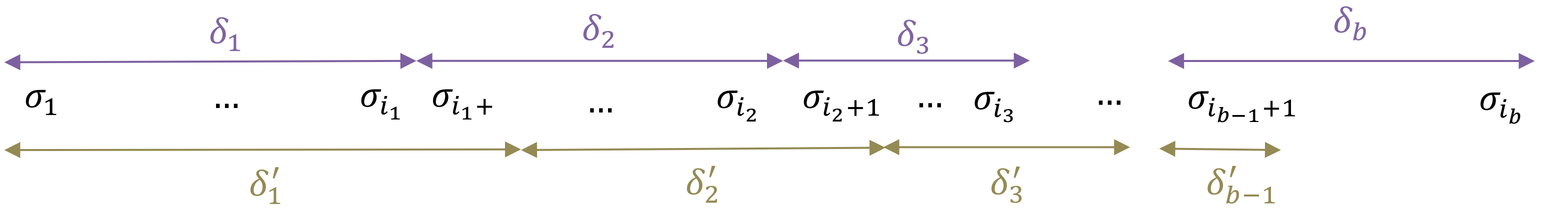
Upper Bound

- **Theorem 2:** Any marking algorithm ALG has a competitive ratio at most k with an additive error.
  - Partition input sequence σ into phases $(\delta_1, \dots \delta_b)$, each with access to k distinct page.
  - ALG has k page fault in each phase



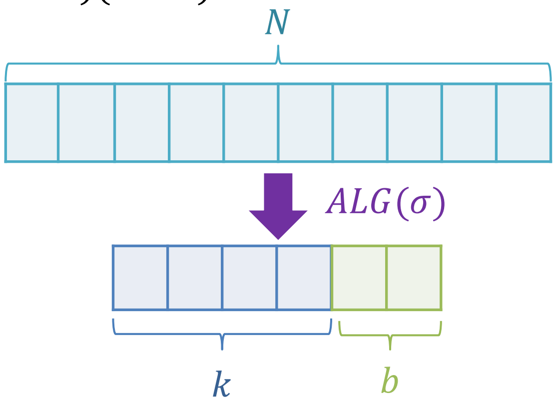
Upper Bound

- **Theorem 2:** Any marking algorithm ALG has competitive ratio at most k with an additive error that goes to zero.
  - Shift phases by one to $(\delta_1', \ldots \delta_b')$,
  - OPT has at least 1 page fault in each shifted phase, $OPT(\sigma) = (b-1)$+k
  - $ALG(\sigma) \leq k \cdot OPT(\sigma) + \frac{k}{b-1+k}$



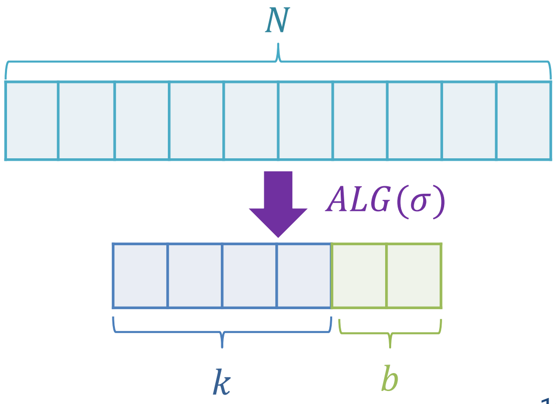
Resource Augmentation

- ALG^a has additional cache with size a
- **Theorem 3:** Any marking algorithm ALG^a has competitive ratio $\frac{k+a}{a+1}$ with an additive error that goes to zero.
  - In each shifted phase, OPT has at least $a+1$ page faults, $OPT(\sigma) = (b-1)(a+1)$+k
  - $ALG(\sigma) \leq \frac{(k+a)}{(a+1)} \cdot OPT(\sigma) + \frac{k}{(b-1)(a+1)+k}$



Resource Augmentation

- Two step approach:
  - Find a cache size that optimal algorithm preforms well
  - Competitive ratio drops to 2 with doubling cache size!



Loosely Comeptitve Ratio

- For a given input sequence, there could not be many ”bad” cache sizes ☺
- **Theorem 4:** For every request sequnce σ, each cache size k in $\{1,2,\ldots,n\}$, the LRU algorithm either has.
    - A Competitve ratio $O(\frac{1}{\delta}\log\frac{1}{\epsilon})$, or
    - At most $\epsilon \cdot |\sigma|$ page fault, or
    - No better gurantee, but for at most δ fraction of

Loosely Comeptitve Ratio

- Proof. Fix an additional augmentation a, First assume cache size k such that
$$LRU(k) > 2LRU(k+a)$$
- Assume that we have δn *bad* cache sizes, then consider the following cache sizes that are at least a apart:
$$1 < k_1 \leq k_2 + a \leq \cdots \leq k_{\frac{\delta n}{a}} + (\ell - 1)a \leq t$$
- Then for each i we have
$$LRU(\mathrm{k_i}) < \frac{1}{2}\mathrm{LRU}(\mathrm{k_{i-1}})$$
- Therefore we have:
$$LRU(t) < \frac{1}{2^{\frac{\delta n}{a}}}\mathrm{LRU}(1)$$

Loosely Comeptitve Ratio

- Therefore we have:

$$LRU(t) < \frac{1}{2^{\frac{\delta n}{a}}} \mathrm{LRU}(1)$$

We want to have $\epsilon \leq \frac{1}{2^{\frac{\delta n}{a}}}$, therefore $a \leq \frac{\delta n}{\log \frac{1}{e}}$ and for every $k \geq t$

$$LRU(k) < \epsilon \cdot |\sigma|$$

Loosely Comeptitve Ratio

- Proof. Now assume cache sizes that have the following property:

$$LRU(k) \leq 2LRU(k+a,\sigma)$$

Then using Theorem 3, we have

$$LRU(k) \leq 2\,\frac{k+a}{a+1}OPT(k) = 2(1+\frac{k-1}{a+1})OPT(k)$$

Having $a \leq \frac{\delta n}{\log\frac{1}{e}}$, then $LRU(k)$ is $O(\frac{1}{\delta}\log\frac{1}{e})$-competitive

Loosely Comeptitve Ratio

- Any $\tau(k, a)$-competitve algorithm, for some function τ that is increasing in k and decreasing in a, for any $\delta, \epsilon, t, \ell$ with $\ell < \delta n + 1$, algorithm is c-loosely competitve for

$$c = \tau(n, \ell)\epsilon^{\frac{-b+1}{\delta n - b - 1}}$$

Thank you