

Online Computation, Ex 2.

David Ponarovsky

January 31, 2023

ex1. Find a simple description of the work-function algorithm in the case of uniform metric space.

Solution. Recall that in the work-function algorithm we weight the configurations by the price one has pay for serving all the requests and ending at those configuration combing the distance between them and the current configuration. In the paging problem the configurations are the content of the cache stack. Denote by m and k the hardisk siz and the number of the servers. Enumerate each of the valid stack states by $Q = q_0, q_1, q_2, \dots, q_M$ where $M = \binom{m}{k}$. In addition define a weight function $w : Q \rightarrow \mathbb{R}$ to be hamming distance between pair of configurations. Finally define a $M \times m$ table $W_{i,j}$ to store the optimal work that has do be done while serving $\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_j$ requests and ending in configuration q_i .

```

1   $q_0 \leftarrow$  initial state
2  for each new request  $\sigma_j$  do
3      for  $i \in [m]$  do
4          if  $\sigma_j \in q_i$  then
5              for  $i' \in [m]$  do
6                   $W_{i,j} \leftarrow$ 
6                       $\min \{W_{i,j}, W_{i',j-1} + \mathbf{Ham}(q_i, q_{i'})\}$ 
7              end
8          else
9               $W_{i,j} \leftarrow \infty$ 
10         end
11     end
12 end
13  $q_0 \leftarrow \arg \min_i \{W_{i,j-1} + \mathbf{Ham}(q_i, q_0)\}$ 
14 Set  $q_0$  as the current state, evict and serve if
    needed.
15 end
```

Algorithm 1: Work-function-Algo for paging.

ex2. Consider the following 3-point metric space, $w(a, b) = 1$ and $w(\cdot, c) = M$. The initial configuration is $\{b, c\}$ (2 servers). Show that randomized competitive ratio, for some value of M is $> H_2 = 1 + \frac{1}{2}$.

Solution. Define the following distribution:

$$\tilde{\sigma} = \begin{cases} (ab)^{\frac{M}{3}} & \text{w.p } \frac{1}{2} \\ (ab)^{\frac{M^{100}}{3}} & \text{w.p } \frac{1}{2} \end{cases}$$

Using Yao's principle, it's enough to show that any deterministic algorithm is H_2 competitive in expectation against that specific distribution. First, notice that knowing what is the exactly drawn σ , fixes an optimal strategy which is one of

the following: moving the server initialized at a between a, b points alternately or choosing first the server that is located in c into a in the second scenario. Putting down, we obtain that:

$$\mathbf{E}[c_{\text{base}}(\sigma) : \sigma \sim \tilde{\sigma}] = \frac{1}{2} \left(\frac{M}{3} + M \right) = \frac{5}{6}M$$

Meanwhile, by the fact that reading any prefix of requests series at length less than $\frac{M}{3}$ doesn't expose any information about the drawn input which wasn't known at the initialized moment, it follows by indistinguishable arguments that the best a randomized algorithm can do is to guess.

ex3. Show that randomized marking algorithm cannot be c -competitive against the adaptive online adversary, for $c = o(k)$.

Solution. Assume by contradiction that there is a constant $c > 1$, and a randomized algorithm which is an c -competitive in the adaptive online setting. According to the theorem shown at class, If there exists an α competitive alg for an online problem in the non-adaptive setting and inadtton there exists a β competitive algorithm for the same problem against adaptive online adversary, then it holds that there exists an algorithm which is $\alpha\beta$ competitive against an offline adaptive adversary. Combining the fact that randomized can't help against such an adversary, we obtain that the deterministic competitive ratio is lower than $\alpha\beta$. As we know that a k -lowerbound for the deterministic regime and also a $\log k$ solution using randomization against a non-adaptive adversary, we obtain that

$$\begin{aligned} \alpha\beta &\geq k \\ \Rightarrow \frac{k}{c} \log k &\geq k \end{aligned}$$

But for any $k \leq \log 2^c$ we obtain the oppsite direction. This means that there is a range of valid k that obtains a better ratio than the lower bound. And that is a contradiction.

ex4 - Ski Rental. At each step, the adversary decides either to continue or stop. Stop terminating the game. If it continues, the online algorithm decides either to rent or buy. Rent costs 1. Buy costs $M > 1$. Design a primal-dual randomized online ski-rental algorithm with a better than 2 competitive ratio.

Solution. . Let's start by formulating an integer LP for the Ski-Rental problem. Denote by m the days' number, and associate a variable x , indicating whether the algorithm decides

to buy. Also, let's associate a variable ξ_j for each day which indicates if the algorithm pays for rent. In each turn, the solution must satisfy the restrictions $\xi_j + x \geq 1$. The cost which we would like to minimize is $M \cdot x + \sum_j \xi_j$. So, in overall, we get that LP is:

$$\begin{aligned} & \min Mx + \sum_j \xi_j \\ \text{s.t. } & x + \xi_j \geq 1 \Leftrightarrow \\ & \begin{bmatrix} 1 & 1 & 0 & 0 & \cdot \\ 1 & 0 & 1 & 0 & \cdot \\ 1 & 0 & 0 & 1 & \cdot \\ 1 & 0 & 0 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x \\ \xi_1 \\ \xi_2 \\ \xi_3 \\ \cdot \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \cdot \end{bmatrix} \end{aligned}$$

So the dual program is

$$\begin{aligned} & \max \sum_j z_j \\ \text{subject to} & \\ & \begin{bmatrix} 1 & 1 & 1 & 1 & \cdot \\ 1 & 0 & 0 & 0 & \cdot \\ 0 & 1 & 0 & 0 & \cdot \\ 0 & 0 & 1 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x \\ z_1 \\ z_2 \\ z_3 \\ \cdot \end{bmatrix} \leq \begin{bmatrix} M \\ 1 \\ 1 \\ 1 \\ \cdot \end{bmatrix} \end{aligned}$$

So in total.

```

1 for each new day  $j$  do
2   if  $x < 1$  then
3      $\xi_j \leftarrow 1 - x$ 
4      $x \leftarrow \left(1 + \frac{1}{M}\right)x + \frac{1}{(c-1)M}$ 
5      $z_j \leftarrow 1$ 
6   end
7 end

```

Algorithm 2: Ski-Rental

ex5. Prove Yao's minimax principle.

$$\begin{aligned} & \forall \text{rand. } \tilde{\text{alg}} \exists \sigma \\ & \mathbf{E} [c_{\text{alg}}(\sigma) : \text{alg} \sim \tilde{\text{alg}}] \geq c \cdot c_{\text{base}}(\sigma) \\ \Leftrightarrow & \exists \text{rand. } \tilde{\sigma} \forall \text{alg} \\ & \mathbf{E} [c_{\text{alg}}(\sigma) : \sigma \sim \tilde{\sigma}] \geq c \mathbf{E} [c_{\text{base}}(\sigma) : \sigma \sim \tilde{\sigma}] \end{aligned}$$

Solution. First direction, assume through contradiction that there exists a deterministic algorithm such that for all distributions $\tilde{\sigma}$:

$$\mathbf{E} [c_{\text{alg}}(\sigma) : \sigma \sim \tilde{\sigma}] < c \mathbf{E} [c_{\text{base}}(\sigma) : \sigma \sim \tilde{\sigma}]$$

And that holds, in particular, for distribution $\tilde{\sigma}$ which supported by a single σ . Hence, because any deterministic algorithm is also a randomized algorithm, set it to be $\tilde{\text{alg}}$, and that immediately yields a contradiction. It is left to show the second direction. By the monotonic property of random variables,

we have that for any distribution $\tilde{\sigma}$:

$$\begin{aligned} & \mathbf{E} [\mathbf{E} [c_{\text{alg}}(\sigma) : \sigma \sim \tilde{\sigma}] : \text{alg} \sim \tilde{\text{alg}}] \\ & \geq c \cdot \mathbf{E} [\mathbf{E} [c_{\text{base}}(\sigma) : \sigma \sim \tilde{\sigma}] : \text{alg} \sim \tilde{\text{alg}}] \\ & \mathbf{E} [\mathbf{E} [c_{\text{alg}}(\sigma) : \text{alg} \sim \tilde{\text{alg}}] : \sigma \sim \tilde{\sigma}] \\ & \geq c \cdot \mathbf{E} [\mathbf{E} [c_{\text{base}}(\sigma) : \text{alg} \sim \tilde{\text{alg}}] : \sigma \sim \tilde{\sigma}] \\ & \mathbf{E} [\mathbf{E} [c_{\text{alg}}(\sigma) : \text{alg} \sim \tilde{\text{alg}}] : \sigma \sim \tilde{\sigma}] \\ & \geq c \cdot \mathbf{E} [c_{\text{base}}(\sigma) : \sigma \sim \tilde{\sigma}] \end{aligned}$$

And by the fact that inequality of expectation between random variables follows an existence of atomic event on which the inequality holds, we obtain that there must exist at least a single σ such that:

$$\mathbf{E} [\mathbf{E} [c_{\text{alg}}(\sigma) : \text{alg} \sim \tilde{\text{alg}}] : \sigma \sim \tilde{\sigma}] \geq c \cdot \mathbf{E} [c_{\text{base}}(\sigma) : \sigma \sim \tilde{\sigma}]$$

And that ends the proof.