Algorithms Under Uncertainty : Quiz 2
Full Marks : 25

29/9/2023

Write solutions in the space provided. NO extra pages will be provided. Write brief and precise solutions. Meaningless rambles fetch negative credits.

Problem 1. (10 points) Consider the following algorithm for k-server: for each server, maintain the total distance travelled by it so far. Let $D(i,t)$ be the total distance travelled by server i till time t. Now suppose a new request arrives at point p_t at time t. Let $v_{i,t}$ be the location of server i at time t (before serving the request at p_t). Now move the server to p_t for which the total distance travelled so far plus the distance to p_t is minimum. In other words, let i be the server which minimizes $D(i,t) + d(p_t, v_{i,t})$. We move this server to p_t. Prove that the competitive ratio of this algorithm is unbounded, i.e., given any number C, which could depend on k and n (recall that n is the number of points in the metric space), the competitive ratio of this algorithm is more than C. (**Hint:** An instance with $k = 2$ and $n = 4$ on the plane suffices).

Solution. Consider a rectangle with vertices labelled A, B, C, D. The length of the edge (A, B) and (C, D) is a large number L, whereas the other two edges have length 1. Now the request sequence is as follows: $A, D, B, C, A, D, B, C, ...$ Also assume that there are two servers which are initially located at B and C – call these servers s_1 and s_2 respectively. Now if s_1 serves the requests at B and C, and s_2 serves the requests at A, D, then after time T, the total cost of this algorithm is about $T/2$.

But we claim that the mentioned algorithm has cost close to $TL/4$. Let us see why. Check that the server s_1 will handle the requests at A, B and S_2 will handle the requests at C, D respectively.

Problem 2. (5 points) Consider the online fractional *weighted* set cover problem - given an universe $U = \{e_1, e_2, \cdots e_n\}$ where the elements of the U are arriving online. Also given is a family of subsets $S_1, S_2, \cdots S_m$ and a non-negative weight function w on the subsets (assume $w_S \geq 1$ for any S). The goal as usual is to maintain a fractional cover of elements that have arrived so far while minimizing $\sum_S w(S) \cdot f_S$.

a. (5 points) State the algorithm from the lecture and show that the algorithm might have arbitrarily bad competitive ratio.

   Solution. This is very easy to see even in the following trivial case. Consider just one element being covered by two sets S_1 and S_2, where S_1 has weight say 1 and S_2 has weight L where L is an arbitrarily large number. Then, just after initialization, cost of the online fractional solution is $(1 + L)/2$, whereas clearly the fractional opimal solution is just 1 (taking S_1 to a fraction of 1).

b. (5 points) Show an exponential update rule that will fix the above issue and give a competitive ratio of $\log(m \cdot w_{\max})$ where $w_{\max}$ is the maximum weight of any subset. Just write the correct update rule. No proof required.

   Solution. Everything else remain the same except that in the initialization phase, set $x_S = 1/(w_S \cdot m)$ for each set S. The rest remains exactly the same.

c. (5 points) Show an exponential update rule that will give a competitive ratio of $\log(m)$. Just write the correct update rule. No proof required.

Solution. This one is a little tricky.

Initialize $x_S \leftarrow 0$
for each arriving element e_t at round $t \geq 1$ **do**
    while $\sum_{e_t \in S} x_S < 1$ **do**
        for each set S such that $e_t \in S$ **do**
            $x_S \leftarrow x_S \left(1 + \frac{1}{w(S)}\right) + \frac{1}{m \cdot w_S}$
        end for
    end while
end for

In fact, the m in the update state can be replaced by f_{e_t} = number of sets that contain e_t.