Advanced Algorithms: Lecture 10 Notes

Dana Randall Spring 2024

Sarthak Mohanty
Aadit Trivedi

The k-server problem

We continue our discussion of online algorithms with perhaps the most influential online problem. It has been a major driving force for the development of the area online algorithms.

The problem is as follows: Consider a (potentially symmetric) metric space $\mathcal{M}$ equipped with a distance function d. There are k servers in this space. The input is a series of points in $\mathcal{M}$, $\sigma = \sigma_1 \dots \sigma_n$. The goal is to move a server j to point σ_i to serve the request while minimizing the total distance moved by all servers.

The first thing to note is that this problem is objectively more difficult than the paging problem discussed last class. To see this, let $\mathcal{M}$ be the *uniform* metric space over n points. This is the metric space where the distance between each pair of points is 1. Note that this problem is equivalent to the paging problem (with n pages in slow memory and k pages in fast memory)

Lower Bounds

We can assume that M has at least $n = k + 1$ points (else the problem is trivial). We fix our attention to any n of the points in M and call them $S = \{1, 2, \dots, k+1\}$. Let the k servers be at the points $1, 2, \dots, k$ and other point is "free".

Now we need to show an adversary ADV and a request sequence σ such that for any (potentially randomized) online adversary A,

$$A(\sigma) \geq k \cdot ADV(\sigma).$$

Instead, we will give a set of k adversaries $ADV_1, \dots, ADV_k$ such that $A(\sigma) \geq \sum_{i=1}^{k} ADV_i(\sigma)$. This implies that at least one of the ADV_i's would satisfy the above condition, which in turn would imply the theorem.

The request sequence σ (for both the deterministic and randomized cases) simply causes the next request at the point in S not covered by A's servers. Thus A has to move one of its servers at every request.

We make one further assumption: the adversary's servers can have any initial position. This is purely for notational convenience, as this would give us at most an additional constant additive cost (to move the adversary's servers at the beginning). This can be swallowed up in the definition of the competitive ratio. Thus the adversary ADV_i starts with its servers on the points $S - \{i\}$. This placement of the servers gives us two properties, which we shall try to maintain invariant over time.

(a) All the adversaries ADV_i have a server at the point of the next request (i.e., at the point not covered by A).

(b) For every request covered by the servers of A, there is a unique ADV_j which does not have a server at that point.

At the first request σ_1, let A move server from j to $k+1$ to serve it. In reply, ADV_i moves its server from $k+1$ to j. The cost incurred by both is the same (as we consider the metric space to be symmetric), and is the properties listed above are maintained. The new state of the algorithm, and the adversaries is isomorphic to the initial state and hence these properties can be restored. So for any σ, we have $A(\sigma) = \sum_{i=1}^{k} ADV_i(\sigma)$.
important paper