SPRING 2010: **COT 6936** Topics in Algorithms
Notes on Online Algorithms
Giri Narasimhan

These notes are compiled from a variety of sources. See the following references: [Manasse et al., 1990]. As with the previous notes, **this is an evolving document.** So, please revisit and download this document every so often so that you see the most updated version.

1 An optimal offline algorithm

Let $C_{OPT}(\sigma, S)$ be a function whose value is the cost of a minimum-cost algorithm that handles request sequence σ and ends up in state S. Note that the state of the system is simply the location of the servers, i.e., the set of vertices in which the servers are located. Assuming that the servers are initially in set S_0, we can write a recursive description of $C_{OPT}(\sigma, S)$ as follows:

$$C_{OPT}(\epsilon, S) = \begin{cases} 0, & \text{if } S = S_0 \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

$$C_{OPT}(\sigma v, S) = \begin{cases} \min_T C_{OPT}(\sigma, T) + d(T, S), & \text{if } v \text{ is covered in } S \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Note that $d(T, S)$ is the cost of a transition from state T to state S. Also, note that the state T that causes the minimum to be reached is the state after request $i - 1$.

The above recurrence can be computed using a dynamic programming (DP) algorithm. As with all DP algorithms, you will need a table – in this case one with $|\sigma| + 1$ rows (one for each prefix of σ) and $\binom{n}{k}$ columns (one for each possible state). The minimum value entry in the last row is the cost of a minimum-cost algorithm for the request sequence σ. As with most DP algorithms, this algorithm can be modified to store the state in the previous row that caused any given entry to be determined. The time required to determine any given entry is proportional to the number of entries in the previous row, i.e., the number of columns in the matrix. I will leave it to you to figure out the time complexity of the entire algorithm.

References

Mark S. Manasse, Lyle A. McGeoch, and Daniel D. Sleator. Competitive algorithms for server problems. *J. Algorithms*, 11(2):208–230, 1990.