This version `Tue Sep 16 12:56:34 PDT 2003`, and still under construction.

Using Potential to Prove Competitiveness

A common method used to prove competitiveness of an online algorithm (either deterministic or randomized) is through the use of a *potential*.

Informal Overview. A potential is a real number which represents the amount of "unused credit" possessed by the algorithm. Each time the algorithm incurs a cost, it must expend credit, subtracting from the potential, and each time the adversary incurs a cost, the potential increases by C times that cost. In the deterministic case, the potential must be non-negative at the end of the request sequence, while in the randomized case, the expected value of the potential must be non-negative at the end of the request sequence. In most situations, the actual requirement is that the potential is non-negative at all times, since we never know when the request sequence will end.

Potentials

Initially, to simplify the discussion, we will ignore randomization, and presume that our online algorithm $\mathcal{A}$ is deterministic.

Let $t \geq 0$ be an integer which measures "time." Most commonly, t measures the number of steps, but in some situations, t measures the number of phases, or perhaps something else. Initially, $t = 0$. We refer to all action that takes place between time $t-1$ and time t as "time interval t," or, if appropriate, "step t."

Definition of Competitiveness. For a given request sequence, let $cost^t_{\mathcal{A}}$ be the cost incurred by $\mathcal{A}$ during time interval t, and let $cost^t_{adv}$ be the cost incurred by the adversary during time interval t. (In most situations, we assume that the adversary is an optimal offline algorithm.) If the number of time intervals is n, we then define

$$\begin{aligned} cost_{\mathcal{A}} &= \sum_{t=1}^{n} cost^t_{\mathcal{A}} \\ cost_{adv} &= \sum_{t=1}^{n} cost^t_{adv} \end{aligned}$$

We then say that $\mathcal{A}$ is C-competititive against that particular adversary if there exists a constant K, independent of the choice of request sequence, such that

$$cost_{\mathcal{A}} \leq C \cdot cost_{adv} + K$$

for all request sequences.

Definition of Potential. A *C-potential* (we just say "potential" if C is understood) for $\mathcal{A}$, against a given adversary, is some real-valued function Φ, whose value at time t we express as Φ^t. For some fixed K which does not depend on the request sequence, it must satisfy the following conditions for each time interval t:

$$\begin{aligned} \Phi^0 &= K \\ \Phi^t &\geq 0 \\ \Phi^{t-1} + C \cdot cost^t_{adv} &\geq cost^t_{\mathcal{A}} + \Phi^t \end{aligned}$$

Theorem 1 *If there is a C-potential for $\mathcal{A}$, then $\mathcal{A}$ is C-competitive.*

Exercise 1 Prove Theorem 1.

The Randomized Case. If $\mathcal{A}$ is randomized, we use expected values.

Exercise 2 Give a formal definition of C-potential for the randomized case.

The randomized analog of Theorem 1 is then:

Theorem 2 *If there is a C-potential for a randomized online algorithm $\mathcal{A}$, then $\mathcal{A}$ is C-competitive.*

The Parameters of a Potential. The potential can have any parameters that make sense. It is normally a function of some combination of past actions, which could include the past request sequence and the current state of the online algorithm. In some cases, the potential is defined in terms of the future request sequence, as in our proof that RANDOM is k-competitiveness. Naturally, in this case, the online algorithm does not "know" the potential. Thus, the primary purpose of a potential is to help prove competitiveness, not to help the algorithm make decisions.

The Competitiveness of RANDOM for the k-Cache Problem

In the proof given below, will define a k-potential that appears to be quite "arbitrary." One of the things that baffles students is that proofs frequently contain mysterious choices that just "happen" to work out. During the lecture, I will explain that the choice of Φ is far from arbitrary; indeed, its value follows from a rather simple assumption (minimality) and is not difficult to calculate.

Theorem 3 *RANDOM is k-competitive for the k-cache problem, for any $k \geq 2$.*

Proof: For any $m > 0$, define $\Phi_m = kH_m$, where $H_m = 1 + \ldots \frac{1}{m}$ is the harmonic number.

We will assume that, at $t = 0$, the cache is full. (This simplifies the calculuation.) Suppose that $\varrho = r^1 r^2 \ldots r^n$ is a request sequence, and assume that *opt* is an optimal service of ϱ. A step consists of a request, followed by a service of that request by RANDOM. Let m^t be the difference between the optimal cache and RANDOM's cache after t steps, *i.e.*, the number of pages in RANDOM's cache that are not in the adversary's cache.

We write $\Phi^t = \Phi_{m^t}$. (This notation may seem confusing: but just remember, the superscript denotes time, while the subscript denotes the size of the difference between $\mathcal{A}$'s cache and the optimal cache.)

We now need to verify the master inequality you wrote for Exercise 2. By Theorem 2, we will then be done.

Let ϱ be any request sequence. For each $t > 0$, we will prove the inequality

$$E(\Phi^{t-1}) + E(kcost^t_{adv}) \geq E(cost^t_{\mathcal{A}}) + E(\Phi^t) \tag{1}$$

Our method is to define the following numerical random variable:[1]

$$X^t = \Phi^{t-1} + kcost^t_{adv} - cost^t_{\mathcal{A}} - \Phi^t$$

and then prove that $E(X^t) \geq 0$.

For each $t > 0$, we also define a non-numeric random variable Y^t. To prove that $E(X^t) \geq 0$, we will prove that the random variable $E(X^t|Y^t)$ never takes on a negative value, then apply Exercise 11 of the handout Random Variables, which you proved using Theorem 5 of that handout.

Let S^t be the set of pages in the optimal cache after t steps. Let T^t be the set of pages in RANDOM's cache after t steps. Let $C^t = S^t \cap T^t$, $A^t = S^t - C^t$, and $B^t = T^t - C^t$. Thus, $|A^t| = |B^t| = m^t$, while $|C^t| = k - m^t$. $S^0 = T^0$, hence A^0 and B^0 are both empty. Note that, for each t, S^t, T^t, C^t, A^t, and B^t. are random variables. Unlike X^t, Φ^t, and m^t, they are not numeric.

For each $t > 0$, let $Y^t = (()S^{t-1}, T^{t-1}, r^t, S^t)$, a random variable whose type is, of course, not numeric. We will prove that, for any given value of Y^t, say Y_j, $E(X^t|Y^t = Y_j) \geq 0$. Then, by Theorem 4 of Random Variables, we are done.

□

[1] For each $t > 0$, X^t is a random variable. Do not become confused and think that the superscript "t" represents a possible outcome; it does not.