Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Susanne Albers
Moritz Fuchs

Spring Semester 2014
Problem set 9
June 5, 2014

Online and approximation algorithms

Due June 18, 2014 before class!

Exercise 1 (EXPO - 10 points)

Recall the algorithm $EXPO$ from the lecture. It can be modified to work even without knowledge about ϕ. Let $\mu = \{q(i)\}_{i=0}^{\infty}$ be a probability distribution over the natural numbers (i.e. i is chosen with probability $q(i)$).
$EXPO_\mu$ chooses the reservation price $p_1 2^i$ with probability $q(i)$ for $i = 0, 1, ...$ where p_1 is the first price that is revealed.
Prove that $EXPO_\mu$ is $\frac{2}{q(\lfloor \log \phi \rfloor)}$-competitive against an oblivious adversary, where ϕ is the posteriori global fluctuation ratio.

Exercise 2 (EXPO II - 10 points)

Extend algorithm $EXPO$ and its analysis to the case in which ϕ is not a power of 2.

Exercise 3 (k-Server - 10 points)

Recall the k-server problem from the lecture. Algorithm *Greedy* always finds the nearest server to each request and moves it to the request location.

(a) Show that *Greedy* is not competitive against an oblivious adversary.

(b) Show that metrical task systems generalize the k-server problem in finite spaces.

Exercise 4 (Lower envelope - 10 points)

Recall the Lower Envelope Algorithm (LEA) from the lecture. In the lecture we showed, that the algorithm is $3 - 2\sqrt{2}$ competitive for general state systems.
We consider the special case where the costs for powering down are additive, i.e. for $i < j < k$, powering down from state s_i to s_j and from s_j to s_k is equally expensive as powering down from s_i to s_k.
Prove that in this setting LEA is 2-competitive.

Hint: Fold the cost for powering down into the cost for powering up. This yields a system where powering down is free and only transitions from low-power states to higher-power states create costs.

Exercise 5 (Energy efficiency - 10 points)

Recall the Energy-efficiency problem from the lecture. In the lecture we showed, that given S,R and D, an online algorithm with the competitive ratio $c^* + \epsilon$ can be constructed in exponential time, where c^* is the optimal competitive ratio possible for this system.
The proof in the lecture introduced the notion of eagerness, now we add the notion of earliness. We define $E(s, p)$ as the earliest transition time at which any online algorithm can transition to state s while still being p-eager (e.g. $E(s_0, p) = 0$). A transition to state s that happens at time $E(s, p)$ is called p-early. A p-early strategy consists exclusively of p-early transitions.
Use dynamic programming to define a decision procedure $EXISTS$ that takes S, R, D as well as a constant p and determines whether a p-competitive strategy for this system exists or not. Show that your procedure runs in polynomial time.

Hint: Without proof use the following lemma: If there is a p-competitive strategy A, then there exists a p-eager and p-early p-competitive strategy.