Advanced Algorithms 11/24, 2017

Exercise 10

Lecturer: Mohsen Ghaffari *Teaching Assistant: Przemysław Uznański*

1 Offline Optimal Algorithm for Paging

Devise a polynomial-time algorithm for computing the optimal offline solution for the paging problem. Prove its correctness and analyze its time complexity.

2 Online Edge Coloring

Given a fixed set of vertices V, a set of edges $E \subseteq V \times V$ arrive over time and upon arrival of each edge, we should color it with one of colors $\{1, 2, \ldots, q\}$. This is the permanent color of that edge and cannot be changed later. The coloring should be a proper coloring at all times, i.e., no two edges that share an endpoint should receive the same color. Suppose that we are given the guarantee that at all times, the maximum degree of any node is at most Δ. Notice that by Vizing's theorem, the offline algorithm can color the edges using just $\Delta + 1$ colors.

(A) Devise an online algorithm that computes a $(2\Delta - 1)$-edge-coloring.

(B) More interestingly, prove that any deterministic online edge-coloring algorithm requires at least $2\Delta - 1$ colors, i.e., no deterministic online algorithm can get a competitive-ratio better than 2.

3 Lost Cow

Consider the lost cow problem described in the class—a cow stands on the x-axis at the origin, and it should get to the exit, which is placed somewhere on the x-axis at some integer distance $d \geq 1$ either to the left or to the right of the origin. Neither the distance nor the side are known to the cow. Devise an algorithm for the cow (!) that helps it minimize the distance it needs to travel to get to the gate, in the worst case.

(A) Prove that the exponential zigzag strategy that was mentioned in the Lecture 9 provides a 9-competitive algorithm.

(B) Prove that this ratio is the best possible for deterministic algorithms.

4 Yao's Min-Max Principle

We need to formalize our approach to proving lower bounds on competitiveness of randomized online algorithms. Consider scenarios (not necessarily in the online setting, we just need a cost function)

1. a randomized algorithm vs. a deterministic input,
2. a deterministic algorithm vs. a randomized input.

Show that, that the expected cost of randomized algorithm on the worst-case input is no better than the cost of deterministic algorithm against worst possible distribution of inputs. In other words, for minimization problems, if p is distribution over inputs $\mathcal{X}$ and q is distribution over deterministic algorithms $\mathcal{A}$, then

$$\max_{x \in \mathcal{X}} \mathbb{E}_{A \sim q}[\text{cost}(A, x)] \geq \min_{a \in \mathcal{A}} \mathbb{E}_{X \sim p}[\text{cost}(a, X)].$$

Show how to interpret this result (Yao's min-max principle) in the language of online algorithms.

5 Lower Bound for the k-Server Problem

Argue that every online deterministic algorithm for the k-server problem has competitiveness ration at least k.

6 Extending Double Coverage to Trees

Prove that the natural extension of the double coverage algorithm to trees gives a k-competitive online algorithm for the k-server problem.

7 Lower Bounds for Online Bipartite Matching

(A) Argue that no deterministic online bipartite matching algorithm can achieve a competitiveness better than $1/2$.

(B) Prove that no (randomized) online bipartite matching algorithm can achieve a competitiveness better than $1 - 1/e$.