CS264: Homework #2

Due by midnight on Wednesday, October 8, 2014

Instructions:

(1) Form a group of 1-3 students. You should turn in only one write-up for your entire group.

(2) Turn in your solutions at `http://rishig.scripts.mit.edu/cs264-bwca/submit-paper.html`. You can contact the TA (Rishi) at `bwca-staff@lists.stanford.edu`. Please type your solutions if possible and feel free to use the LaTeX template provided on the course home page.

(3) Students taking the course for a letter grade should complete all exercises and problems. Students taking the course pass-fail should complete 5 of the exercises.

(4) Write convincingly but not excessively. Exercise solutions rarely need to be more than 1-2 paragraphs. Problem solutions rarely need to be more than a half-page (per part), and can often be shorter.

(5) You may refer to your course notes, and to the textbooks and research papers listed on the course Web page *only*. You cannot refer to textbooks, handouts, or research papers that are not listed on the course home page. Cite any sources that you use, and make sure that all your words are your own.

(6) If you discuss solution approaches with anyone outside of your team, you must list their names on the front page of your write-up.

(7) Exercises are worth 5 points each. Problem parts are labeled with point values.

(8) No late assignments will be accepted.

Lecture 3 Exercises

Exercise 7

Prove that for every cache size $k \geq 1$ and every page sequence σ, $\text{cost}(LRU, k+1, \sigma) \leq \text{cost}(LRU, k, \sigma)$.

[As usual, $\text{cost}(A, k, \sigma)$ denotes the number of page faults incurred by the paging algorithm A on the page sequence σ when the cache size is k.]

Exercise 8

Recall the final two results from this lecture: a resource augmentation guarantee for LRU (competitive ratio $k/(k-h+1)$ against the optimal offline paging algorithm with cache size h); and Young's corollary that LRU performs well (in either a relative sense or an absolute sense) for most cache sizes. Which subset of these two guarantees applies also to the FIFO algorithm? What about to the Flush-When-Full algorithm? Explain your answer.

Exercise 9

Suppose we try to interpolate between online and offline algorithms using the notion of *lookahead*. Precisely, an *ℓ-lookahead* paging algorithm makes each eviction decision using only knowledge of past requests and the ℓ next requests. (Online algorithms are the $\ell = 0$ case, offline algorithms the $\ell = +\infty$ case.) Prove that for every finite $\ell \geq 0$ and cache size $k \geq 1$, every deterministic ℓ-lookahead algorithm has competitive ratio at least k.

[Interpretation: this result gives yet another sense in which worst-case competitive analysis gives misleading information about online paging algorithms — it suggests that any finite amount of lookahead is completely useless.]

Lecture 4 Exercises

Exercise 10

Prove that the model of locality introduced in this lecture has no implications for the competitive ratio of an algorithm. Precisely, prove that for every concave function f with $f(2) = 2$ and $\lim_{\ell \to \infty} f(\ell) = N$ (where N is the total number of pages), every cache size k, and every deterministic online algorithm A, the competitive ratio of A on the subset of sequences legal for f is at least k.

Exercise 11

Recall the proof of the first lower bound (of $\alpha_f(k)$) given in this lecture. Given a concave function f with $f(2) = 2$, the proof considered a page sequence σ that consists of an arbitrarily large number of phases. Each phase consists of the same sequence of page requests: m_2 requests in a row for page p_1, m_3 requests in a row for page p_2, ..., and finally m_k requests in a row for page p_{k-1}. (Recall that m_j denotes the number of (consecutive) values of i for which $f(i) = j$, and that m_j is nondecreasing in j by concavity of f.) Prove that the sequence σ conforms to f.

Exercise 12

Here's an alternative simple model of locality in page sequences. Suppose we change the cost model so that every cache hit costs 1 and every cache miss costs $m \geq 1$. Recall from Lecture #3 how we break a sequence σ into blocks $\sigma_1, \sigma_2, \ldots, \sigma_b$, where each σ_i is a maximal sequence in which only k distinct pages are requested. Define the locality $L(\sigma)$ of sequence σ as its average block length (the number $|\sigma|$ of requests divided by b). Intuitively, the larger this value, the more locality exhibited by the sequence.

Prove that for every sequence σ with locality at least αm, the cost (in this new cost model) of LRU is at most $1 + \frac{k-1}{\alpha+1}$ times that of the optimal (furthest-in-future) algorithm.

Problems

Problem 3

(10 points) Recall from Lecture #4 that there exists a concave function f and a cache size k such that the worst-case page fault rate of FIFO on sequences legal for f is strictly larger than that of LRU. The point of this problem is that it is never too much larger. Specifically, prove that for every $k \geq 2$ and concave f with $f(2) = 2$, the worst-case page fault rate of FIFO on request sequences σ that conform to f is at most

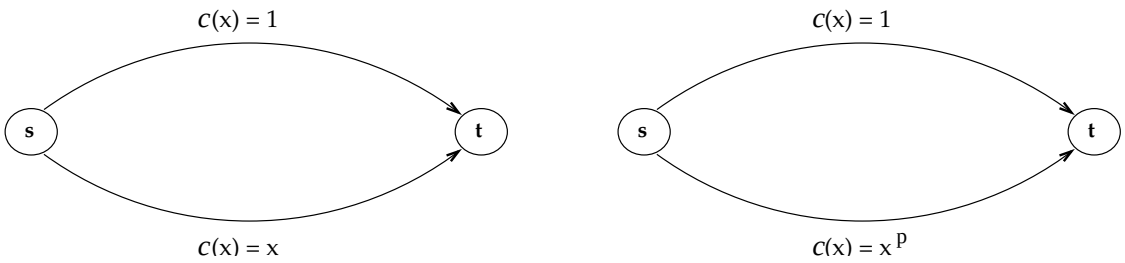
$$\frac{k}{f^{-1}(k+1) - 1}. \tag{1}$$

[Hint: make some minor modifications to the upper bound proof for LRU. Note that the expression in (1) suggests defining phases such that (i) FIFO makes at most k faults per phase; and (ii) if one looks at a phase plus one additional request, these must comprise requests for at least $k + 1$ distinct pages.]

[Interpretation: note that for most k and f the upper bound in (1) is not much bigger than our upper and lower bound $\alpha_f(k)$ for LRU.]

Problem 4

Exercise 9 shows that competitive analysis is not useful for measuring the benefit of lookahead. This problem takes uses a different analysis method to tackle the problem.



(a) Pigou's example (b) A nonlinear variant

Figure 1: The cost function $c(x)$ describes the cost incurred by users of an edge, as a function of the amount of traffic routed on the edge.

Fix a cache size k and a lookahead value ℓ. To quantify the power of lookahead, we examine the following ratio:

$$\max_{A \in \mathcal{C}_\ell} \min_{B \in \mathcal{C}_0} \max_{\sigma} \frac{\text{cost}(B, k, \sigma)}{\text{cost}(A, k, \sigma)}, \tag{2}$$

where $\mathcal{C}_0$ denotes the set of deterministic online algorithms and $\mathcal{C}_\ell$ the set of deterministic ℓ-lookahead algorithms (defined in Exercise 9). In other words, imagine a game being played by two players: the first player picks a good ℓ-lookahead algorithm A with the goal of showing the power of lookahead; the second player picks a standard online algorithm B with the goal of showing that it is not much worse than A; finally, the first player picks an input z that showcases how much better A is than B. The bigger the expression in (2), the more powerful the class $\mathcal{C}_\ell$ relative to $\mathcal{C}_0$.

(a) (5 points) Assume that $\ell < k$. Prove that (2) is at least $\ell + 1$.

[Hint: modify the first lower bound argument from Lecture #3, using a good ℓ-lookahead algorithm in place of the optimal offline algorithm.]

(b) (10 points) Prove that (2) is at most $\ell + 1$.

[Hint: given a choice of A, respond with an algorithm B that, for a given page sequence σ, approximately simulates (the cache of) A. One approach is to define B so that the following statement holds (by induction and a case analysis): after every subsequence of requests that causes $c \in \{0, 1, \ldots, \ell\}$ page faults to B and zero page faults to A, there are at most $\ell - c$ pages that are in the cache of A but not in the cache of B.]

Problem 5

The point of this problem is to illustrate another application of resource augmentation. We need to describe some preliminaries about *selfish routing networks*. We consider a directed flow network $G = (V, E)$, with r units of traffic traveling from a source vertex s to a sink vertex t. Each edge e of the network has a flow-dependent cost function $c_e(x)$. For example, in the network in Figure 1(a), the top edge has a constant cost function $c(x) = 1$, while the cost to traffic on the bottom edge equals the amount of flow x on the edge.

The optimal solution is the fractional s-t flow that routes the r units of traffic to minimize the cost $\sum_e c_e(f_e) f_e$, where f_e denotes the amount of flow on edge e. For example, in Figure 1(a), the optimal flow splits traffic evenly between the two paths, for a cost of $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$. In the *equilibrium flow*, all traffic is routed on shortest paths, where the length of a path P is the flow-dependent quantity $\sum_{e \in P} c_e(f_e)$. The optimal flow in Figure 1(a) is not an equilibrium flow: traffic on the top path is not routed on a shortest path, and hence would want to revise its route. In the only equilibrium flow in Figure 1(a), all traffic uses the bottom path for an overall cost of $0 \cdot 1 + 1 \cdot 1 = 1$.

With general (nonlinear) cost functions, the ratio in cost between the equilibrium and optimal flows (known as *the price of anarchy*) can be arbitrarily large. To see this, replace the cost function on the bottom

edge by $c(x) = x^p$ for p large (Figure 1(b)). The equilibrium flow remains the same, with all selfish traffic using the bottom edge for an overall cost of 1. The optimal flow improves with p, however: splitting the traffic $1 - \epsilon$ on the bottom edge and ϵ on the top yields a flow with cost $\epsilon + (1 - \epsilon)^{p+1}$. As $p \uparrow \infty$ and $\epsilon \downarrow 0$ appropriately, this cost tends to zero, and hence the price of anarchy (the ratio of costs) tends to infinity.

Despite the negative example above, a very general resource augmentation guarantee holds in selfish routing networks.

Theorem 1 *For every network G with continuous and nondecreasing cost functions, for every traffic rate r and $\delta > 0$,*

$$\textit{equilibrium flow cost at traffic rate } r \leq \frac{1}{\delta} \cdot \textit{optimal flow cost at traffic rate } (1+\delta)r.$$

This upper bound is the best possible for every $\delta > 0$, as shown by considering the network in Figure 1(b) and letting $p \to \infty$.

We do not prove Theorem 1 here (though it's not too hard, see e.g. Lecture #12 of the instructor's CS364A course), and instead focus on proving an analog of Young's result from Lecture #3, which roughly states that, in every network, the price of anarchy is much better for "most" traffic rates than for a worst-case traffic rate.

Formally, let $\pi(G, r)$ denote the ratio of the costs of equilibrium flows at rate r and rate $r/2$. By Theorem 1, the price of anarchy in the network G at rate r is at most $\pi(G, r)$.

(a) (8 points) Use Theorem 1 to prove that, for every G and $r > 0$, and for at least a constant fraction of the traffic rates $\hat{r}$ in $[r/2, r]$, the price of anarchy in G at traffic rate $\hat{r}$ is at most $c \log \pi(G, r)$ (where $c > 0$ is a constant, independent of G and r).

(b) (7 points) Prove that for every constant K, there exists a network G with continuous, nondecreasing edge cost functions and a traffic rate r such that the price of anarchy in G is at least K for every traffic rate $\hat{r} \in [r/2, r]$.

[Hint: use a network with many parallel links.]