CS 224 Advanced Algorithms — Spring 2017

Problem Set 4

Due: 11:59pm, Wednesday, March 8th

Submit solutions to Canvas, one PDF per problem:
https://canvas.harvard.edu/courses/21996

Solution max page limits: One page each for problems 1 and 2, and two pages each for problems 3 and 4

See homework policy at http://people.seas.harvard.edu/~cs224/spring17/hmwk.html

Problem 1: (5 points) Prove Theorem 6 from Lec. 10 ("approx. complementary slackness").

Problem 2: (5 points) Consider here the purely multiplicative form of competitive ratios: we say that an algorithm $\mathcal{A}$ has competitive ratio α if for every input sequence σ, $C_{\mathcal{A}}(\sigma) \leq \alpha \cdot \mathsf{OPT}$, where $C_{\mathcal{A}}(\sigma)$ denotes the cost of $\mathcal{A}$ on σ. Recall in weighted paging we have a cache of size k, and there are n pages where each page p has some weight $w(p) > 0$. For the purposes of this problem, when we compare an online weighted paging algorithm with OPT, we assume that both of their caches are empty at beginning of processing σ. Recall in class we said that weighted paging reduces to k-server, by having one metric space point per page with the distance $d(p, q)$ between pages being $(w(p) + w(q))/2$. This almost works, but not quite: if some server visits a sequence of pages $p_1, \ldots, p_r$ over the course of processing σ, then this server has added an amount to our total cost equal to $(1/2)(w(p_1) + w(p_r)) + \sum_{1 < i < r} w(p_i)$. Thus, the first and last page are only *half* counted, which is not quite right.

In this problem, you are to fix the above reduction. In particular, show that if $\mathcal{A}$ achieves a purely multiplicative competitive ratio of α for k-server, then for every $\epsilon > 0$, $\mathcal{A}$ can be used in some subroutine $\mathcal{A}_\epsilon$ which solves weighted paging with a purely multiplicative competitive ratio of at most $\alpha + \epsilon$.

Problem 3: (Problem due to Nikhil Bansal). In Lecture 9 we showed that 1-bit LRU is k-competitive. Let us try to give a different proof using the online primal-dual framework. First, let's write the primal LP. Let k denote the size of cache and n denote the total number of pages in the universe. There are variables x_i^t for each page i. This is intended to be 1 if i is absent from the cache immediately after servicing the request time t, and 0 otherwise. Let $r(t) \in \{1, \ldots, n\}$ denote the page requested at time t. The variable $z_{i,t}$ is intended to be 1 if we evict page i at time t and is 0 otherwise. This leads to the following LP relaxation:

$$\min \sum_{t=2}^{T} \sum_{i=1}^{n} z_{i,t}$$

$$
\begin{aligned}
\text{s.t.} \quad \sum_{i=1}^{n} x_i^t &\geq n-k && \forall 1 \leq t \leq T \\
x_{r(t)}^t &\leq 0 && \forall 1 \leq t \leq T \\
x_i^t &\leq 1 && \forall 1 \leq i \leq n, 1 \leq t \leq T \\
z_i^t &\geq x_i^t - x_i^{t-1} && \forall 1 \leq i \leq n, 1 < t \leq T \\
x_i^t, z_i^t &\geq 0 && \forall 1 \leq i \leq n, 1 \leq t \leq T
\end{aligned}
$$

(a) (3 points) Write the dual of the above LP. (Note any LP can be written in the form stated in class.) **Hint:** Other than the final nonnegativity constraints, there are four types of constraints above. These should give rise to four types of variables in the dual.

(b) (7 points) Give a primal/dual analysis of the online 1-bit LRU algorithm, showing that it suffers at most $k(\mathsf{OPT} + 1)$ page faults. **Hint:** One of the types of variables in the dual, let's call them the d_i^t variables, should keep track of whether page i is marked at time t. Define a potential function $\Phi(t) = \sum_i d_i^t$ and show that if $P(t)$ is the increase primal cost at time t (compared with time $t-1$) and $D(t)$ is the increase in dual profit, then there is a way to maintain primal and dual feasible solutions online so that for all t, $P(t) \leq k \cdot D(t) + (\Phi(t) - \Phi(t-1))$.

Problem 4: Recall single source shortest paths problem in directed graphs with nonnegative edge weights. There is a directed graph $G = (V, E)$, $|V| = n$, $|E| = m$. We will identify V with $\{1, \ldots, n\}$. There is also a length function $L : E \to \mathbb{Z}_{\geq 0}$ (i.e. every edge has some nonnegative integer length). We are given a "source" vertex s. For this problem we will assume every vertex in V is reachable from s. We would like to recover the "shortest path" tree T from s. T is directed, rooted at s and with all edges pointing away from s, such that the shortest path from s to any vertex t in G is exactly the unique path from s to t in T.

We can formulate a fractional relaxation of the problem as follows.

$$\min \sum_{e \in E} L(e) \cdot f_e$$

$$
\begin{aligned}
\text{s.t.} \quad Bf &= \chi \\
f_e &\geq 0 && \forall e \in E
\end{aligned}
$$

Here $\chi \in \mathbb{R}^n$ is the vector $(n-1)\mathbf{1}_s - \sum_{v \in V \setminus \{s\}} \mathbf{1}_v$, where $\mathbf{1}_i$ is the ith standard basis vector as a column vector. $B \in \mathbb{R}^{n \times m}$ is the matrix whose columns are indexed by edges, where the column corresponding to $e = (u, v)$ equals $\mathbf{1}_u - \mathbf{1}_v$. Essentially we should view the LP as finding a "flow" with s as the source, shipping out $n-1$ units of flow, and where each

other vertex absorbs one unit of flow. Thus in an integral flow solution, we can decompose the flow into paths, corresponding to the *s-t* paths for each other $t \in V$.

1. (4 points) As it will turn out, there always is an optimal integral solution to the above LP. This doesn't mean that *every* optimal solution is integral though (note the optimal solution might not be unique). Give an example input graph with nonnegative integral length function L such that there exists an optimal solution which is not integral.

2. (3 points) Write the dual of the above LP.

3. (8 points) Prove the correctness of Dijkstra's algorithm via a primal/dual analysis, i.e. by building primal and dual feasible solutions where the primal is integral.

Note the edges in the shortest path tree after finding an integral solution to the primal LP are simply the edges e with $f_e > 0$.