Due date: October 21, 2024

Problem 1: Let $d \geq 1$ be a constant and $H = \{h_i, \ldots, h_n\}$ be a set of n constraints of the form $h_i = \{x : a_i^\top x \leq b_i\}$, where $b_i \in \mathbb{R}$ and $a_i \in \mathbb{R}^d$ for all $i \leq n$. Assume, given an objective vector $c \in \mathbb{R}^d$, Simplex(H) computes the optimal solution to the linear program $\max c^\top x$ s.t. $x \in h$ for all $h \in H$. Consider the following randomized LP algorithm.

Algorithm 1 LP via MWU

```
$w(h) = 1 \ \ \forall h \in H$
if $|H| \leq 9d^2$ then
    return Simplex$(H)$
end if
repeat
    $R$: random subset of $H$, where $|R| = 9d^2$ and $\Pr[h \in R] = \frac{w(h)}{w(H)}$
    $x^* \leftarrow$ Simplex$(R)$
    $V \leftarrow \{h \in H : x^* \notin h\}$
    if $w(V) \leq \frac{2w(H)}{9d}$ then
        $w(h) \leftarrow 2w(h), \ \forall h \in V$
    end if
until $V = \varnothing$
return $x^*$
```

(i) Let $B \subseteq H$ be a set of d constraints that define x^*. Show that if $V \neq \varnothing$, then $V \cap B \neq \varnothing$.

(ii) Derive an upper bound on $w(H)$ and a lower bound on $w(B)$ after the multiplicative update step $w(h) \leftarrow 2w(h), \ \forall h \in V$ has been executed i times.

(iii) Prove that the multiplicative update step $w(h) \leftarrow 2w(h), \ \forall h \in V$ is executed $O(\log n)$ times. Recall that d is a constant.

(iv) Assuming $\Pr\left[w(V) > \frac{2w(H)}{9d}\right] \leq \frac{1}{2}$, what is the expected running time of the algorithm.

Problem 2: Consider the multiple knapsack problem (MKP): There are k knapsacks with capacities $B_1, B_2, \ldots, B_k$ and there are n items, each of which has a profit p_i and a size s_i. The goal is pack these k knapsacks with the items so that size constraints are met and the profit is maximized.

(i) Describe a pseudo-polynomial time exact algorithm for the problem for $k = 2$.

(ii) Consider the MKP problem as in the previous problem but with uniform capacities. Consider a greedy algorithm that picks each knapsack in turn and packs it using a α-approximate algorithm for the single knapsack problem over the remaining items. Prove that Greedy yields a $(1 - e^{-\alpha})$-approximation.

Problem 3: Consider the following scheduling problem: there are n jobs to be scheduled on a single machine, where each job j has a processing time p_j, a weight w_j, and a due date d_j, $j = 1, \ldots, n$. The objective is to schedule the jobs so as to maximize the total weight of the jobs that complete by their due date.

(i) Prove that there always exists an optimal schedule in which all on-time jobs complete before all late jobs, and the on-time jobs complete in an earliest due date order.

(ii) Use the structural result from the previous part to show how to solve this problem using dynamic programming in $O(nW)$ time, where $W = \sum_j w_j$.

(iii) Use the results from the previous parts to derive a fully polynomial-time approximation scheme.

Problem 4: A *multigraph* $G = (V, E)$ is a weighted graph where each edge $e \in E$ has a non-negative integral weight $w(e) \in \mathbb{Z}_{\geq 0}$. The cost of a cut $(U, V \setminus U)$, where $U \neq \emptyset$, is defined as $\textsc{cost}(U, V \setminus U) = \sum\limits_{e \in E(U, \{V \setminus U\})} w(e)$.

An α-approximate cut is any cut $(U, V \setminus U)$, such that

$$\textsc{Cost}(U, V \setminus U) \leq \alpha \cdot \min_{U' \subset V} \textsc{Cost}(U', V \setminus U').$$

(i) What is the probability that the randomized MinCut algorithm taught in class returns an α-approximate cut when run on G?

(ii) How many unique α-approximate cuts does G have?