EECS 279: Approximation Algorithms — Release: 10-23-2014
Homework 4-B — Due: 3:00pm, Tuesday, 11-4-2014

Make your solutions concise and formal. Your goal is to convince me that you know the solutions. You are highly encouraged to typeset your solutions in Latex. See the course website for further homework policies. Please write down your collaborators names and references if any.

In the Far-far-away country, there is a gas station with n *all-or-nothing* pumps. That is, at pump i, you can buy exactly s_i gallons of gas for w_i dollars, and you get nothing from the pump for less than w_i dollars. We can use each pump at most once. To drive back to Merced, you need to collect at least D gallons of gas, and your goal is to minimize the payment. For simplicity, all input parameters s_i, w_i, and D are assumed to be non-negative integers. It is also assumed that D is polynomially bounded by n.

1. Give a simple 2-approximation. Your algorithm must run in time $O(n^2 \log n)$. Neither LP nor dynamic programming is needed – we can indeed solve this problem optimally in polynomial time via a simple dynamic programming when D is polynomially bounded by n.

2. Let's consider the following IP for this problem.

$$\min \sum_{i \in [n]} w_i x_i$$

$$\sum_{i \in [n]} s_i x_i \geq D$$

$$x_i \in \{0, 1\} \qquad i \in [n]$$

   (a) We can relax this IP to an LP by replacing $x_i \in \{0,1\}$ with $x_i \in [0,1]$. Show that this LP has an integrality gap of at least $\Omega(D)$. You should be able to show a bad integrality gap even for an instance with $n = 1$.

   (b) Consider the following modified IP with each s_i capped at D. Argue that this IP is a valid relaxation. In general, you need to show that for any instance I, $\mathrm{OPT}_{IP}(I) \leq \mathrm{OPT}$, but here it suffices to show that any feasible (integral) solution for the original problem satisfies all IP constraints.

$$\min \sum_{i \in [n]} w_i x_i$$

$$\sum_{i \in [n]} \min\{s_i, D\} \cdot x_i \geq D$$

$$x_i \in \{0, 1\} \qquad i \in [n]$$

   As before, we can obtain an LP relaxation by allowing $x_i \in [0,1]$. Show that this LP still has an integrality gap of at least $\Omega(D)$.

3. We will strengthen the LP as follows. Consider the following IP where $s(A) := \sum_{i\in A} s_i$.

$$\min \sum_{i\in[n]} w_i x_i$$

$$\sum_{i\in[n]\setminus A} \min\{s_i, (D - s(A))_+\} \cdot x_i \geq D - s(A) \qquad\qquad A \subseteq [n]$$

$$x_i \in \{0,1\} \qquad\qquad i \in [n]$$

Here $(y)_+ = \max\{y, 0\}$. Show that why this IP formulation is valid. That is, show that any feasible (integral) solution satisfies all constraints. Note that there are exponentially many constraints that need to be satisfied.

4. As before, we get an LP relaxation by allowing $x_i \in [0,1]$. Show that you can solve this LP in polynomial time in n. In other words, show a separation oracle: for any x_i, show how to find a violated constraint in polynomial time in n if such a constraint exists.

   (a) Explain that the constraint is satisfied for all A with $s(A) \geq D$ if $x_i \geq 0$. One line answer is sufficient.

   (b) Consider any fixed integer $0 \leq k \leq D$ and any $0 \leq x_i \leq 1$, $x_i \in [n]$. We want to see if there is A with $s(A) = k$ for which the constraint is violated. Then, note that the constraint can be rewritten as

   $$\sum_{i\in[n]} \min\{s_i, D-k\} \cdot x_i - (D-k) \geq \sum_{i\in A} \min\{s_i, D-k\} \cdot x_i.$$

   Do you see the left-hand-side is fixed? Also $\min\{s_i, D-k\} \cdot x_i$ is fixed for each i. Note that if the constraint is violated for some A with $s(A) = k$, it must be violated for A with $s(A) = k$ such that the right-hand-side $\sum_{i\in A} \min\{s_i, D-k\} \cdot x_i$ is maximized.

   Give a dynamic programming that finds such a set A. Complete the separation oracle by considering all $k := s(A)$.

5. (Extra credit). Show that the last LP has an integrality gap of 2. Note that you need to show both the upper and lower bounds on the integrality gap.

   Hint. To show the upper bound, if $x_i \geq 1/2$, pick i. Then, use the constraint for $A = \{i \in [n] \mid x_i \geq 1/2\}$. You will need pick more pumps in addition to A.