This page is intentionally left blank.
Please do not open this question set before you are allowed to do so.

4810-1183 Approximation and Online Algorithms with Applications (Spring 2018)

Midterm Problem 1

Suppose that we have the following optimization model. We will call it as Model 1.

Input: $\mathcal{A}$

Output: $\mathcal{B}$

Constraint: $\mathcal{C}$

Objective Function: Maximize $\mathcal{D}$

We assume that $\mathcal{D}$ is always positive. Consider Model 2 that has the same input, output, and constraint, but have the following objective function.

Objective Function: Minimize -$\mathcal{D}$

Question 1.1: Discuss why an optimal solution of Model 1 is also an optimal solution of Model 2.

Suppose that we have a 0.5-approximation algorithm for Model 1, and the objective value obtained from the approximation algorithm is SOL, and the optimal value of Model 1 is OPT.

Question 1.2: What is the objective value of the same solution for Model 2? What is the optimal value of Model 2 in that case?

Question 1.3: Discuss why we have 0.5-approximation algorithm also for Model 2.

Question 1.4: Discuss why having approximation ratio equal 0.5 for Model 2 is a problem. Also, discuss why we have the problem.

Now, let consider Model 3. The model has the same input, output, and constraint as Models 1,2, but have the following objective function.

Objective Function: Minimize $1/\mathcal{D}$

Question 1.5: Discuss why an optimal solution of Model 1 is also an optimal solution of Model 3.

Question 1.6: Suppose that we have a 0.5-approximation algorithm for Model 1. Discuss why that algorithm is a 2-approximation algorithm for Model 3.

Question 1.7: Discuss why, when we have to turn our problem to a minimization problem, changing objective function to $1/\mathcal{D}$ is a better choice than $-\mathcal{D}$.

4810-1183 Approximation and Online Algorithms with Applications (Spring 2018)

Midterm Problem 2

In this problem, we will try to use deterministic rounding to solve the knapsack problem (simplified version). Recall that the optimization models for the knapsack problem is as follows:

Input: Weight of each strawberry $w_1, \ldots, w_n$, maximum weight we can eat W

Output: Set of strawberries that we eat $S \subseteq \{1, \ldots, n\}$

Constraint: $\sum_{i \in S} w_i \leq W$

Objective Function: Maximize $\sum_{i \in S} w_i$

Instead of having S as an output, we will have $x_1, \ldots, x_n \in \{0,1\}$ as outputs in the following questions.

Question 2.1: Write an optimization model for the knapsack problem with $x_1, \ldots, x_n$ as an output. You cannot use S in your formulation for constraint and objective function.

We will consider the optimization model in Question 2.1. However, instead of having $x_i \in \{0,1\}$, we will have $x_i \in [0,1]$.

Question 2.2: We will solve the problem using linear program library. When using the library, we have to construct matrix **A**, vectors **b**, **c, x** from the input and output of the knapsack problem. Discuss how we construct those matrix and vectors.

We will call the linear program obtained from Question 2.2 as fractional knapsack problem. Suppose that the optimal output of the fractional knapsack problem is $\mathbf{x} = [x_1, x_2, \ldots, x_n]^t$. Using deterministic rounding, we will construct a solution for the knapsack problem $\mathbf{x}' = [x'_1, \ldots, x'_n]^t$ such that $x'_i \in \{0,1\}$. For a real number $0 \leq t \leq 1$, we will have $x'_i = 1$ when $x_i \geq t$ and we will have $x'_i = 0$ otherwise.

Question 2.3: What is the smallest value of t which we can be sure that $\mathbf{x}'$ will always satisfy the constraint of the knapsack problem?

Question 2.4: Give an example where $\mathbf{x}'$ do not satisfy the constraint of the knapsack problem, when t is *slightly* smaller than your answer in Question 2.3.

Question 2.5: By your answer in Question 2.3, what is the best approximation ratio we can obtain from the deterministic rounding scheme?

Question 2.6: Discuss why we should not use the deterministic rounding technique for devising an algorithm for the knapsack problem.

4810-1183 Approximation and Online Algorithms with Applications (Spring 2018)

Midterm Problem 3

In our discussion on the adaptive Bloom filter, for element i that is in the cache S with probability P_i, we randomly set k_i cells of Hash table to 1. To minimize the probability of having false positive, we optimize the following optimization problem.

Input: Positive integer N (number of possible information)
Positive real numbers $w_i^{(j)} = P_i$ for $1 \le i \le N$ and $1 \le j \le 3$
Positive real number $W = m \cdot \ln 2$, when m is the Hash table size
Positive real number $h_i^{(1)} = 0.5$, $h_i^{(2)} = 0.25$, $h_i^{(3)} = 0.125$ for $1 \le i \le N$

Output: $S \subseteq \left\{p_i^{(j)}: 1 \le i \le N \text{ and } 1 \le j \le 3\right\}$
(Elements i will trigger at least j bits when $p_i^{(j)} = 1$.)

Constraint: $\sum_{p_i^{(j)} \in S} w_i^{(j)} \le W$

Objective Function: Maximize $\sum_{p_i^{(j)} \in S} h_i^{(j)}$

We stop our discussion here, but there is still a lot of issues to consider. When $p_i^{(2)} = 0$ and $p_i^{(3)} = 1$, we will not be able to find an appropriate value for k_i. The output $p_i^{(2)} = 0$ will be violated if $k_i \ge 2$, but $p_i^{(3)} = 1$ will be violated when $k_i < 2$. In the next question, we will show that the situation will not happen if we use the greedy algorithm for the knapsack problem.

Question 3.1: Discuss why, when we use the 0.5-approximation algorithm for the knapsack problem to solve the above optimization model, we will have $p_i^{(j)} = 1$ only if $p_i^{(j')} = 1$ for all $j' < j$.

Another issue from the above optimization model is: we set the maximum value of k_i is no more than 3 in the above optimization model. In reality, we might have k_i much larger than 3, and what we have from the optimization model might be far from optimal. From the next question, we will show that assuming $k_i \le 3$ is not that bad idea.

Consider the second optimization model, which have the same input, constraint, and objective function, but the following output:

Output: $S \subseteq \{p_i^{(j)}: 1 \le i \le N \text{ and } j \ge 1\}$

We will call the first problem as $Bloom_3$ and the second optimization problem as $Bloom_\infty$.

Question 3.2: Suppose that S^* is an optimal solution of $Bloom_\infty$, and $S' = \left\{p_i^{(j)} \in S^*: 1 \le i \le N \text{ and } 1 \le j \le 3\right\}$. Discuss why the objective value of S' is no larger than the optimal value of $Bloom_3$.

Question 3.3: Discuss why the objective value of S^* is no larger than 8/7 times of the objective value of S'.

Question 3.4: Discuss why the optimal value of $Bloom_3$ is no smaller than 7/8 times of the optimal value of $Bloom_\infty$.

Question 3.5: Discuss why a 0.5-approxiation algorithm for $Bloom_3$ is a 7/16-approximation algorithm for $Bloom_\infty$.

Question 3.6: Devise a 0.5-approximation algorithm for $Bloom_\infty$ based on the greedy algorithm for the Knapsack problem.