# CS 6150: HW 6 – Optimization formulations, review
Collaborators: Brian Schnepp, Craig Butler, Yo Office Hours

Submission date: Tuesday, December 14, 2021, 11:59 PM

This assignment has 5 questions, for a total of 50 points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

| Question | Points | Score |
|---|---|---|
| Understanding relax-and-round | 16 | |
| The power of two choices | 10 | |
| Optimal packaging | 10 | |
| Non-negativity in Markov | 4 | |
| Distributed independent set | 10 | |
| Total: | 50 | |

Question 1: Understanding relax-and-round .......................................................... **[16]**

In this question, you will implement the relax and round paradigm using the example of the Set Cover (or hiring) problem. Suppose we have $n$ people (i.e., sets) and $m$ skills that we wish to cover. Let us create an instance with $n = m = 500$. Let $d = 25$ be the size of each skill set.

(a) [**3**] Write code that generates a random instance of set cover, where for each person $i$, the skill set $S_i$ is a random subset of the $m$ skills, with $|S_i| = d$.
See the code

(b) [**3**] Write down the integer linear program using variables $x_i$ that indicate if person $i$ is chosen/hired (abstractly, as we did in class). Then write down the linear programming relaxation. Which one has the lower optimum objective value?
ILP: $\sum_{i=1}^{n} XiAij > 1$ where skill j is covered by more than 1 person
LP: $\sum_{i=1}^{n} XiAij \geq 1$ where skill j is covered by at least 1 person

(c) [**6**] For the instance you created in part (a), solve the linear program from part (b) using an LP solver of your choice, and output the fractional solution.
See the code

(d) [**4**] Round the fractional solution using randomized rounding, i.e., hire person $i$ with probability $\min(1, tx_i)$. Try $t = 1, 2, 4, 8$, and in each case, report the (a) total number of people hired, and (b) number of "uncovered" skills (i.e., skills for which none of the people possessing the skill were hired).
see code

Question 2: The power of two choices .......................................................... **[10]**

As I mentioned in class, one of the classic applications of random hashing is "on-the-fly" load balancing. In this problem, we will empirically see how "balanced" such an assignment is, and how to make it more balanced.

In what follows, set $N = 10^7$, i.e., 10 million. Suppose we have $N$ servers, and $N$ service requests arrive sequentially.

(a) [**4**] When a request arrives, suppose we generate a random index $r$ between 1 and $N$ and send the request to server $r$ (and we do this independently for each request). Plot a histogram showing the distribution of the "loads" of the servers in the end. I.e., show how many servers have load 0, load 1, and so on. [The load is defined as the number of requests routed to that server.]
see code

(b) [**6**] Now, suppose we do something slightly smarter: when a request arrives, we generate *two* random indices $r_1$ and $r_2$ between 1 and $N$, query to find the *current load* on the servers $r_1$ and $r_2$, and assign the request to the server with the *lesser* load (breaking ties arbitrarily). With this allocation, plot the histogram showing the load distribution, as above.
see code

Question 3: Optimal packaging .......................................................... **[10]**

With the holiday season around the corner, company Bezo wants to minimize the number of shipping boxes. Let us consider the one dimensional version of the problem: suppose a customer orders $n$ items of lengths $a_1, a_2, \ldots, a_n$ respectively, and suppose $0 < a_i \leq 1$. The goal is to place them into boxes of length 1 such that the total **number of boxes** is minimized.

It turns out that this is a rather difficult problem. But now, suppose that there are only $r$ *distinct* values that the lengths could take. In other words, suppose that there is some set $L = \{s_1, \ldots, s_r\}$ such that every $a_i \in L$. Let us think of $r$ is as a small constant. Devise an algorithm that runs in time $O(n^r)$, and computes the optimal number of boxes.

[*Hint:* first find all the possible "configurations" that can fit in a single box. Then use dynamic programming.]

Question 4: Non-negativity in Markov .................................................................. **[4]**

Markov's inequality states that for a non-negative random variable $X$, we have $\Pr[X > t \cdot \mathbb{E}[X]] \leq 1/t$, for any $t \geq 1$. The point of this exercise is to show that the non-negativity is important. Give an example of a random variable (that takes negative values), for which (a) $\mathbb{E}[X] = 1$, and (b) $\Pr[X > 5] \geq 0.9$.

Given the condition $a$) we pick some number greater than 5 say 6 and set up the following equation to get $E[X] = 1$. $\rightarrow 6 \cdot 0.9 + (-44) \cdot 0.1 = 1$

Given the following from condition $b$): $Pr[x > 5] = 0.9$ we apply Markov's inequality:

$\rightarrow Pr[X > 5 \cdot E[X]] \leq \frac{1}{5}$ which violates requirement $b$). Thus we have shown a case of a random variable that satisfies both of conditions stated in the problem.

Question 5: Distributed independent set .................................................................. **[10]**

A fundamental problem in distributed algorithms (used in P2P networks, distributed coloring, etc.) is the following: we are given an undirected graph with $n$ vertices where each vertex corresponds to an 'agent'. The goal is to find a large independent set (a set of vertices with no edges between them) in a distributed manner. It turns out that this problem has a nice solution when the degree of each vertex is $\leq d$, for some known parameter $d$.

Consider the following algorithm. (1) Every vertex becomes *active* with probability $\frac{1}{2d}$. (2) Every active vertex queries its neighbors, and if any vertex in the neighborhood is also active, it becomes inactive. (Step (2) is done in parallel; thus if $i$ and $j$ are neighbors and they were both activated in step (1), they both become inactive.) (3) The set of active vertices in the end is output as the independent set.

(a) **[2]** Let $X$ be the random variable that is the number of vertices activated in step (1). Find $\mathbb{E}[X]$.
$n \cdot \frac{1}{2 \cdot d}$

(b) **[3]** Let $Y$ be the random variable that is the number of edges $\{i, j\}$ *both of whose end points* are activated in step (1). Find $\mathbb{E}[Y]$ (in terms of $m$, the total number of edges in the graph).
$m \cdot \frac{1}{2 \cdot d} \cdot \frac{1}{2 \cdot d}$

(c) **[5]** Prove that the size of the independent set output in (3) is at least $X - 2Y$, and thus show that the expectation of this quantity is $\geq n/4d$.
$E[X - 2Y] = \frac{n}{2d} - 2 \cdot \frac{m}{4 \cdot d^2}$
$\geq \frac{n}{2 \cdot d} - \frac{\frac{1}{2} \cdot n \cdot d}{2 \cdot d^2}$
$= \frac{n - \frac{1}{2} \cdot n}{2d} = \frac{n}{4 \cdot d}$