EPFL

Exercise Set V

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 *(homework problem from previous year)* **Randomized rounding.** Consider the standard linear programming relaxation of Set Cover that we saw in class. We gave a randomized rounding algorithm for the Set Cover problem. Use similar techniques to give an algorithm that, with probability at least a positive constant, returns a collection of sets that cover at least 90% of the elements and has cost at most a constant factor larger than the LP solution.

Solution: Using Claim 7 and Corollary 8 from the Lecture 7 notes, the expected cost of collection C after d executions of Step 3 of the algorithm for set cover given in Lecture 7 notes is at most $d \cdot LP_{OPT}$.

Let X_i be a random variable corresponding to the event whether i-th element is covered by the output C or not. Specifically, let $X_i = 1$ if the given constraint is not satisfied (therefore given element is not covered) and $X_i = 0$ if it is satisfied (therefore the element is covered) after d executions of Step 3. Let X denote the total number of constraints that are not satisfied. Therefore we write,

$$X = X_1 + X_2 + \cdots + X_n.$$

From Claim 9 in the Lecture 7 notes, we know that the probability that a constraint remains unsatisfied after a single execution of Step 3 is at most $\frac{1}{e}$. In addition, from the first step in the proof of Claim 10, the probability that a constraint is unsatisfied after d executions of Step 3 is at most $\frac{1}{e^d}$. In other words, we have in our notation that

$$\mathbb{P}(X_i = 1) \leq \frac{1}{e^d}.$$

Since each X_i is a Bernoulli random variable, we can write their expectation as

$$\mathbb{E}(X_i) \leq \frac{1}{e^d}.$$

We want to bound the probability that more than 10 % of the elements are not covered (which also means less than 90 % of the elements are covered). We can use Markov's Inequality to write,

$$\begin{aligned}\mathbb{P}\left(X \geq \frac{n}{10}\right) &\leq \frac{\mathbb{E}(X)}{n/10} \\ &= \frac{\mathbb{E}(X_1) + \mathbb{E}(X_2) + \cdots + \mathbb{E}(X_n)}{n/10} \\ &\leq \frac{n \cdot \frac{1}{e^d}}{n/10} \\ &= \frac{10}{e^d}.\end{aligned}$$

Lastly, similar to Claim 11 in the lecture notes, we will bound the probability of bad events (namely, the cost is high or less than 90 % of elements are covered). Firstly, we found that expected cost after d executions is at most $d \cdot LP_{OPT}$. We can write using Markov's Inequality that,

$$\mathbb{P}(cost \geq 5d \cdot LP_{OPT}) \leq \frac{1}{5}.$$

Secondly, we bound the probability of event that less than 90 % of elements are covered. We did it above and showed that probability that more than 10 % of the elements are not covered is at most $\frac{10}{e^d}$. In the worst case, these bad events are completely disjoint. Therefore, the probability that no bad event occurs is at least $1 - \frac{1}{5} - \frac{10}{e^d} > \frac{1}{2}$ for some large constant d. Therefore, the algorithm, with probability at least $\frac{1}{2}$, will return a collection of sets that cover at least 90 % of the elements and has cost at most $5dLP_{OPT} \leq 5d\text{OPT}$.

2 (*) Consider the LP-rounding algorithm for Set Cover that works as follows:

1. Solve the LP relaxation to obtain an optimal solution x^*.
2. Return the solution $\{S : x^*_S > 0\}$, i.e., containing all sets with a positive value in the fractional solution.

Use the complementarity slackness conditions to prove that the algorithm is an f-approximation algorithm, where f is the frequency (i.e., the maximum number of sets that any element belongs to).

Solution: Let y be an optimal dual solution. By complementary slackness we have that for each set S, either $x^*_s = 0$ or $\sum_{e \in S} y_e = c(S)$. Let us now compute the cost of our algorithm. The cost of the algorithm is $\sum_{S: x^*_S > 0} c(S)$. By complementarity slackness we get that

$$\sum_{S: x^*_S > 0} c(S) = \sum_{S: x^*_S > 0} \sum_{e \in S} y_e \leq \sum_{S} \sum_{e \in S} y_e = \sum_{e \in U} y_e \sum_{S \ni e} 1 \leq \sum_{e \in U} f \cdot y_e.$$

We also know that (since y is a feasible dual solution) $\sum_{e \in U} y_e \leq OPT$. Therefore the cost of the above algorithm is at most $f \cdot OPT$.

3 Consider the following grumpy commuters problem. There are n persons. Each person $i = 1, \ldots, n$ has to choose a path from his/her home to work from a set $\mathcal{P}_i$ of potential paths (in an underlying graph with n vertices that models the road network). In addition, person i has a personal preference of paths modeled with a cost function $c_i : \mathcal{P}_i \to \mathbb{R}_+$. The goal is to select paths $p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, \ldots, p_n \in \mathcal{P}_n$ (one for each person) so that

1. The paths $p_1, \ldots, p_n$ are edge-disjoint (the commuters are grumpy and do not want to share roads).
2. The total cost $c_1(p_1) + c_2(p_2) + \cdots + c_n(p_n)$ is minimized.

3a Write an exact integer linear program for the grumpy commuters problem.

3b *(*)* Give a randomized rounding algorithm that returns a set of paths $p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, \ldots, p_n \in \mathcal{P}_n$ satisfying:

1. The total cost $c_1(p_1) + c_2(p_2) + \cdots + c_n(p_n)$ is at most a constant factor times the cost of the optimal value of the LP-relaxation.
2. Each edge is used by at most $100 \cdot \log n / \log \log n$ paths.

You may assume that the underlying graph has n vertices.

Hint: for the analysis (of the second condition) the following specialized Chernoff bound[1] *will be useful: Let $X_1, \ldots, X_n$ be n* independent *random variables taking values in $\{0, 1\}$. If $\mathbb{E}[X_1 + \cdots + X_n] \leq 1$, then*

$$\Pr[X_1 + \cdots + X_n > 100 \cdot \log n / \log \log n] < \frac{1}{n^3}\,.$$

Solution: We first define an exact integer linear program for the grumpy commuters problem. We have a variable x_{ij} for each person $i = 1, \ldots, n$ and each path $p_j \in \mathcal{P}_i$. The intuition is that x_{ij} should take value 1 if person i chooses the path $p_j \in P_i$, and 0 otherwise. Let E be the set of edges in the graph.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{p_j \in \mathcal{P}_i} c_i(p_j) x_{ij} \\
\text{subject to} \quad & \sum_{i=1}^{n} \sum_{p_j \in \mathcal{P}_i : e \in p_j} x_{ij} \leq 1 \quad \forall e \in E \\
& \textstyle\sum_{p_j \in \mathcal{P}_i} x_{ij} = 1 \qquad \forall i = 1, \ldots, n \\
& x_{ij} \in \{0, 1\} \qquad \forall i = 1, \ldots, n, p_j \in \mathcal{P}_i.
\end{aligned}$$

Since we do not know how to solve integer programs in polynomial time, we relax the last constraint to $0 \leq x_{ij} \leq 1$. Therefore we get a polynomial size linear program so we can solve it in polynomial time as well. We will explain an algorithm such that the expected cost of its solution is equal to the cost of the optimum LP solution and with probability at least $1 - 1/n$ it gives a feasible solution (uses each edge at most $100 \cdot \log n / \log \log n$ times). The former implies, by Markov's inequality, that the cost of the returned solution is at most $4OPT_{LP}$ with probability

[1] Chernoff bounds are an extremely useful tool for proving concentration of sums of independent random variables taking values in $\{0, 1\}$. We will revisit them later in the course.

at least 3/4. Therefore the probability that something goes wrong (the solution is too costly – more than $4OPT_{LP}$ – or infeasible) is at most $1/4 + 1/n < 1/3$. This is called a Monte Carlo algorithm.[2]

Let us design such an algorithm. To this end, let x^* be the optimal value of the above LP. For each player $i = 1, \ldots, n$ pick one of the paths in $\mathcal{P}_i$ with probability proportional to x^*_{ij} (make this pick independently for each player). Let the selected collection of paths be $t = (t_1, ..., t_n)$. Let us first analyze the expected cost of this algorithm:

$$\text{expected cost of } t = \sum_{i=1}^{n} \sum_{p_j \in \mathcal{P}_i} c_i(p_j) x^*_{ij} = \text{cost of } x^* = OPT_{LP}.$$

Now let us compute the probability that t uses any edge more than $100 \cdot \log n / \log \log n$ times. Consider any edge e and let X_i be a random variable which is 1 if $e \in t_i$ and zero otherwise for $1 \le i \le n$. Then

$$\mathbb{E}[X_i] = \sum_{p_j \in P_i : e \in p_j} x^*_{ij}.$$

Therefore,

$$\mathbb{E}[X_1 + \cdots + X_n] = \sum_{i=1}^{n} \sum_{p_j \in P_i : e \in p_j} x^*_{ij} \le 1$$

by the LP constraint. The variables X_i are independent, because the path of each player was sampled independently. Therefore, by the Chernoff bound from the hint, we get that the probability of using this edge more than $100 \log n / \log \log n$ times is at most $1/n^3$. Since the underlying graph has n vertices, it has at most n^2 edges, and so the probability of violating any edge is at most $1/n$ (using a union bound).

4 Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. Consider the following linear program with n variables:

$$\begin{array}{ll} \textbf{maximize} & c^T x \\ \textbf{subject to} & Ax = b \\ & x \ge 0 \end{array}$$

Show that any extreme point x^* has at most m non-zero entries, i.e., $|\{i : x^*_i > 0\}| \le m$.

Hint: what happens if the columns corresponding to non-zero entries in x^* are linearly dependent?

(If you are in a good mood you can prove the following stronger statement: x^* is an extreme point if and only if the columns of A corresponding to non-zero entries of x^* are linearly independent.)

Solution: Without loss of generality, index x^* so that $x^*_1, \ldots, x^*_k > 0$ and $x^*_{k+1}, \ldots, x^*_n = 0$. Also let $a_1, a_2, \ldots, a_k$ denote the columns of A corresponding to the nonzero variables $x^*_1, \ldots, x^*_k$.

We start by showing that if $a_1, \ldots, a_k \in \mathbb{R}^m$ are linearly dependent then x^* is not an extreme point. This implies that "any extreme point x^* has at most m non-zero entries, i.e., $|\{i : x^*_i >$

[2] However, if something does go wrong, we can actually detect it (it is easy to compare the cost of the returned solution to the LP optimum and to count the number of times each edge is taken) and then rerun the algorithm. By doing so, we get a so-called Las Vegas algorithm, which *always* gives a good solution, but its running time is polynomial *in expectation* (and, in this case, with very high probability as well).

$0\}| \leq m$" since no more than m vectors can be linearly independent in the m-dimensional space $\mathbb{R}^m$.

Since we assume $a_1, \ldots, a_k$ to be linearly dependent we can write $\sum_{i=1}^{k} \lambda_i a_i = 0$ for some scalars λ_i that are not all equal to 0. Now it is easy to verify that for a small enough $\varepsilon > 0$,

$$y := \begin{bmatrix} x_1^* \\ \vdots \\ x_k^* \\ x_{k+1}^* \\ \vdots \\ x_n^* \end{bmatrix} + \varepsilon \cdot \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \text{and} \qquad z := \begin{bmatrix} x_1^* \\ \vdots \\ x_k^* \\ x_{k+1}^* \\ \vdots \\ x_n^* \end{bmatrix} - \varepsilon \cdot \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

are both feasible solutions to the linear program and $x^* = \frac{y+z}{2}$ and thus x^* is not an extreme point.

We now complete the proof of the good mood problem by showing that if x^* is not an extreme point, then $a_1, \ldots, a_k$ are linearly dependent. If x^* is not an extreme point, then there is a vector $y \in \mathbb{R}^n \setminus \{0\}$ such that $x^* + y$ is feasible and $x^* - y$ is feasible. By simple rewriting, $Ax = b$ and $A(x + y) = b$ imply $Ay = 0$, i.e., $\sum_{i=1}^{n} a_i y_i = 0$. By the nonnegativity constraints, we must have $y_i = 0$ for all $i > k$ (for suppose that $y_i > 0$ and $x_i^* = 0$; then either $x^* - y$ or $x^* + y$ would need to be negative on coordinate i and thus infeasible). Hence we have that $\sum_{i=1}^{k} a_i y_i = 0$, which shows that the vectors $a_1, \ldots, a_k$ are linearly dependent.

In the proofs above we used that x^* is not an extreme point (i.e., it can be written as a convex combination of other feasible vectors) if and only if it can be written as a convex combination of **two** other feasible vectors. The proof here is as follows: suppose that $x^* = \sum_{i=1}^{n} \lambda_i y_i$ for $\lambda_1, ..., \lambda_n > 0$, $\sum_i \lambda_i = 1$ and $y_1 \neq x^*$. Then we can rewrite

$$x^* = \sum_{i=1}^{n} \lambda_i y_i = \lambda_1 y_1 + \sum_{i=2}^{n} \lambda_i y_i = \lambda_1 y_i + (1 - \lambda_1) \cdot \left(\sum_{i=2}^{n} \frac{\lambda_i}{1 - \lambda_1} y_i \right)$$

which is a convex combination of two other feasible points. The point $\sum_{i=2}^{n} \frac{\lambda_i}{1-\lambda_1} y_i$ is feasible as a convex combination of feasible points (note that $\sum_{i=2}^{n} \lambda_i = 1 - \lambda_1$).

5 Consider the following quadratic programming relaxation of the Max Cut problem on $G = (V, E)$:

$$\begin{aligned} \textbf{maximize} \quad & \sum_{\{i,j\} \in E} (1 - x_i) x_j + x_i (1 - x_j) \\ \textbf{subject to} \quad & x_i \in [0, 1] \quad \forall i \in V \end{aligned}$$

Show that the optimal value of the quadratic relaxation actually equals the value of an optimal cut. (Unfortunately, this does not give an exact algorithm for Max Cut as the above quadratic program is NP-hard to solve (so is Max Cut).)

Hint: analyze basic randomized rounding.

Solution: We give a rounding algorithm that gives a cut x^* (an integral vector) of expected value equal to the value of an optimal solution x to the quadratic program. The rounding algorithm is

the basic one: set $x_i^* = 1$ with probability x_i and $x_i^* = 0$ with probability $1 - x_i$. The expected value of the objective function is then $\sum_{i,j} \mathbb{E}\left((1 - x_i^*)x_j^* + x_i^*(1 - x_j^*)\right) = \sum_{(i,j)\in E}(1 - x_i)x_j + x_i(1 - x_j)$ by independence of the variables x_i^* and x_j^*. Thus the expected value after rounding is the optimal value of the quadratic relaxation. But as no integral cut can have larger value than the relaxation, the value of the obtained cut x^* must always equal the value of the fractional solution x. It follows that the optimal value of the quadratic relaxation equals the value of an optimal cut.