AM 221: Advanced Optimization Spring 2016

Prof. Yaron Singer *Problem Set 12 — Due Wed, Apr. 27th at 23:59*

Instructions: All your solutions should be prepared in LaTeX and the PDF and .tex should be submitted to canvas. For each question, the best and correct answers will be selected as sample solutions for the entire class to enjoy. If you prefer that we do not use your solutions, please indicate this clearly on the first page of your assignment.

The programming parts can be written in the programming language of your choice and the code should be submitted alongside your solutions.

1. Randomized rounding for Max-Cut. Remember the Max-Cut problem: the input is a graph $G = (V, E)$ and the goal is to find a subset $S \subseteq V$ so as to maximize the number of edges which cross from S to $V \setminus S$. For a cut S, we write $C(S)$ the value of the cut (the number of cut edges).

Max-Cut can be formulated as an IP in the following way: let us introduce one variable y_e for each edge $e \in E$ and a variable x_u for each $u \in V$. $x_u = 1$ will express that $u \in S$. Max-Cut can then be written as:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}} & \sum_{e \in E} y_e \\ \text{s.t. } & y_{uv} \leq x_u + x_v, \ (u,v) \in E \\ & y_{uv} \leq 2 - (x_v + x_u), \ (u,v) \in E \\ & y_{uv} \in \{0,1\}, \ (u,v) \in E \\ & x_u \in \{0,1\}, \ u \in V \end{aligned}$$

a. Let $(\mathbf{x}, \mathbf{y})$ be an optimal solution to the LP relaxation of this IP (where the constraints $\in \{0,1\}$ are replaced by $\in [0,1]$). Show that:

$$y_{u,v} = \min\{x_u + x_v, 2 - (x_u + x_v)\} \ (u,v) \in E$$

b. Let $(\mathbf{x}, \mathbf{y})$ be a feasible solution to the LP relaxation, show that the random set $S_{\mathbf{x}}$ constructed by included each vertex $u \in V$ in $S_{\mathbf{x}}$ independently with probability x_u is such that:

$$\mathbb{E}[C(S_{\mathbf{x}})] \geq \frac{1}{2} \sum_{(u,v) \in E} \min\{x_u + x_v, 2 - (x_u + x_v)\}$$

c. Using part a. and b., describe a randomized algorithm for Max-Cut such that the expected value of the solution is a $\frac{1}{2}$-approximation to the optimal solution.

2. Ellipsoid method for combinatorial auctions. Remember the combinatorial auction problem: there is a set m items and n bidders. Each bidder $i \in [n]$ has a valuation function $v_i : 2^{[m]} \to \mathbb{R}$ which maps a subset $S \subseteq [m]$ of items to a real value. We assume that the valuation functions are normalized: $v_i(\emptyset) = 0$ and monotone: $v_i(S) \leq v_i(T)$ when $S \subseteq T$.

An allocation of items to the bidders is a partition $S_1, \ldots, S_n$ (with $S_i \cap S_j \neq \emptyset$ for all $i \neq j$). The social welfare associated with this allocation is $\sum_{i=1}^{n} v_i(S_i)$. Finding the optimal allocation can be formulated as an IP by introducing a variable $x_{i,S} \in \{0,1\}$ for each bidder $i \in [n]$ and each subset of items S. We then want to solve:

$$\begin{aligned}
\max \quad & \sum_{i \in [n], S \subseteq [m]} x_{i,S}\, v_i(S) \\
& \sum_{i \in [n], S : j \in S} x_{i,S} \leq 1,\ j \in [m] \\
& \sum_{S \subseteq [m]} x_{i,S} \leq 1,\ i \in [N] \\
& x_{i,S} \in \{0,1\}
\end{aligned}$$

a. Explain the first two constraints in the integer program above.

b. Consider the LP relaxation of the integer program where the constraint $x_{i,S} \in \{0,1\}$ is replaced with $x_{i,S} \geq 0$. How many variables does it have? Can it be solved in time polynomial in $n+m$?

c. Show that the dual of this LP relaxation takes the following form:

$$\begin{aligned}
\min \ & \sum_{i \in [n]} u_i + \sum_{j \in [m]} p_j \\
\text{s.t. } & u_i + \sum_{j \in S} p_j \geq v_i(S),\ i \in [n], S \subseteq [m] \\
& u_i \geq 0,\ p_j \geq 0
\end{aligned}$$

Note that there is one variable for each $i \in [n]$ and one variable for each $j \in [m]$ (this is polynomial in $n+m$), but the number of constraints is exponential in $n+m$.

In the remaining part of this problem we will show how to solve the dual of the LP relaxation in polynomial time. Remember from section that an LP with polynomially many variables can be solved in polynomial time given access to a polynomial time separation oracle (and this, even if the number of constraints is exponential).

For the above dual LP, a polynomial time separation oracle is simply a polynomial time algorithm which given a pair $(\mathbf{u}, \mathbf{p})$ outputs either "$(\mathbf{u}, \mathbf{p})$ is feasible" or identifies a constraint which is violated. To construct the separation oracle, we will assume that we can make "demand queries" to the valuation functions.

Definition. *A **demand query** presents a set of prices $p_1, \ldots, p_m$ to bidder $i \in [n]$. The bidder answers the query with a set S such that:*

$$v_i(S) - \sum_{i \in S} p_i \geq v_i(S') - \sum_{i \in S'} p_i,\ S' \subseteq [m]$$

Demand queries model the real-life scenario where the bidders do not reveal their entire valuation functions to the auctioneer. Instead, given a set of prices, they can report the most valuable set of items under those prices.

d. Construct a separation oracle for the above dual LP. The separation oracle should make a number of demand queries which is polynomial in $n+m$. Assuming that demand queries take time $O(1)$ conclude that the dual LP can be solve in polynomial (in $n+m$) time.

3. Lovász extension and submodular minimization. In this problem, we will introduce the Lovász extension, which is another way to extend a submodular function into a continuous relaxation. As we will see, it is a powerful tool when doing submodular minimization.

Let $f : 2^N \to R$ be a submodular function defined on subsets of N with $|N| = n$. We will assume that f is normalized, *i.e* $f(\emptyset) = 0$. The Lovász extension f_L of f is defined over $[0,1]^n$ by:

$$f_L(\mathbf{x}) = \int_0^1 f\big(\{i : x_i \geq \lambda\}\big) d\lambda, \quad \mathbf{x} \in [0,1]^n$$

a. For a vector $\mathbf{x} \in [0,1]$ let us denote by $i_1, \ldots, i_n$ an ordering of $[n]$ such that:

$$x_{i_1} \geq x_{i_2} \geq \cdots \geq x_{i_n}$$

and let us define $S_k \subseteq N$ by $S_k = \{i_1, \ldots, i_k\}$ (with the convention $S_0 = \emptyset$). Show that the Lovász extension can be rewritten:

$$f_L(\mathbf{x}) = x_{i_n} f(S_n) + \sum_{k=1}^{n-1} (x_{i_k} - x_{i_{k+1}}) f(S_k)$$

with the convention $x_0 = 0$.

b. Show that f_L is an extension of f: for $S \subseteq N$, let us denote by $\mathbf{1}_S$ the indicator vector of S (the ith coordinate of $\mathbf{1}_S$ is 1 if $i \in S$ and 0 otherwise), then: $f_L(\mathbf{1}_S) = f(S)$.

c. Show that if f_L is convex then f is submodular. [**Note:** the converse is also true; if f is submodular then f_L is convex, but we will not prove it here, you can however assume that it is true].

d. Show that:

$$\min_{\mathbf{x} \in [0,1]^n} f_L(\mathbf{x}) = \min_{S \subseteq N} f(S)$$

what does this imply for the computational complexity of minimizing a submodular function?