Algorithm Design - Homework 2

Sapienza University of Rome

Marco Costa, 1691388

January 10, 2020

Exercise 1

First Problem

We need to model Michele's problem as a integer linear problem, and relax this to a corresponding LP.

Solution

The problem can be modeled as an *ILP* as follow:

$$\begin{cases} minimize & \sum\limits_{i \in F} \sum\limits_{j \in O} x_{ij} w(i,j) \\ s.t. & \sum\limits_{j \in O} x_{ij} = 1, \forall i \in F \\ & \sum\limits_{i \in F} x_{ij} \leq 1, \forall j \in O \\ & x_{ij} \in \{0,1\} \end{cases}$$

For the relaxation of this problem to a LP problem, we need simply to change the last condition in $x_{ij} \in [0,1]$ $(0 \leq x_{ij} \leq 1)$.

Second Problem

Give a polynomial-time algorithm that, from any given optimal LP-solution, computes an optimal integer assignment, knowing that there is no *integrality gap* and for every fractional LP-solution, there exists an integral feasible solution with the same cost.

Solution

Let's assume that $\hat{x}$ is an optimal solution for the LP. If, for each $x_{ij} \in \hat{x}$, we have that $x_{ij} \in \{0,1\}$, then $\hat{x}$ is also an optimal solution for the ILP.
But, if $\exists\, x_{ij} \in [0,1]$ then $\hat{x}$ is fractional so it is not an optimal solution for the ILP problem.
Now, consider the subgraph $H \in G$ (where G is the graph composed by the matching of the two sets F and O) comprising all the fractional values $x_{ij} \in \hat{x}$ (so H will be a graph whose edges are the ones with fractional weights deriving from $\hat{x}$).
Now, either H is a cycle or it is a path whose source and sink vertices have all incident edges $e_{ij} \in G.edges$ with $x_{ij} = 0$ except for one that has weight less then 1. In order to prove this, each edge that belongs to H has a fractional weight by definition. Let $(v_1, v_2) \in H$ be the first edge of a path, obviously we have $0 < x_{ij} < 1$. All the other incident edges of v_1 that are not in H cannot have $x_{ij} = 1$ (otherwise the constraints will be violated) or $0 < x_{ij} < 1$ (otherwise they will be in H), so for all this edges we will have $x_{ij} = 0$. Now, consider a graph H defined as before. We can define the following value at the beginning of each step of the algorithm:

$$\epsilon = \min_{\{i,j\} \in H} (x_{ij}, 1 - x_{ij}) \tag{1}$$

At each step of the algorithm, for each $x_ij \in \bar{x}$ we compute the two following updates:

$$x'_{ij} = \begin{cases} x_{ij} - \epsilon, & \text{if } e_{ij} \text{ has odd idx in G.edges} \wedge e_{ij} \in H.edges \\ x_{ij} + \epsilon, & \text{if } e_{ij} \text{ has even idx in G.edges} \wedge e_{ij} \in H.edges \\ x_{ij}, & \text{if } e_{ij} \notin H.\text{edges} \end{cases} \tag{2}$$

$$x''_{ij} = \begin{cases} x_{ij} - \epsilon, & \text{if } e_{ij} \text{ has odd idx in G.edges} \wedge e_{ij} \in H.edges \\ x_{ij} + \epsilon, & \text{if } e_{ij} \text{ has even idx in G.edges} \wedge e_{ij} \in H.edges \\ x_{ij}, & \text{if } e_{ij} \notin H.\text{edges} \end{cases} \tag{3}$$

We define x' as the solution whose elements are x'_{ij} and x'' the solution whose elements are x''_{ij}. At each step we perform the two updates over the solution with the highest number of integer values. The number of steps required to obtain an integral solution is $O(n)$ where n is the number of fractional solutions in the initial optimal solution $\hat{x}$ of the LP. The most important thing is that, for every intermediate solution computed (so x' and x'') the value of the objective function doesn't change, so also at the end of the algorithm it will be the same as the beginning.

Exercise 2

First Problem

We need to show that $\mathbb{E}[|B'|] \leq 2|OPT|$.

Solution

Let's define two variables X_i a d Y_i, the first is set to 1 if the algorithm adds a vertex belonging to OPT and to 0 otherwise, while the second is set to 1 if the algorithm adds a vertex not belonging to OPT and to 0 otherwise. Let's define also $X = \sum_i X_i$ and $Y = \sum_i Y_i$.

In this way we have that $|B'| = X + Y$ and $X \leq |OPT|$.
Now let's consider this: if an edge e is already covered, we have $X_i = Y_i = 0$, if both vertices of e belong to OPT, then $X_i = 1$ and $Y_i = 0$, and $Pr[X_i] = Pr[Y_i] = \frac{1}{2}$ otherwise.
So we can see that $Pr[Y_i] \leq Pr[X_i]$, and it means that $\mathbb{E}[|Y|] \leq \mathbb{E}[|X|]$.
Finally we have $\mathbb{E}[|B'|] = \mathbb{E}[|X+Y|] = \mathbb{E}[|X|] + \mathbb{E}[|Y|] \leq 2\,\mathbb{E}[|X|] \leq 2|OPT|$, so $\mathbb{E}[|B'|] \leq 2|OPT|$.

Second Problem

We need to show that for every constant $c \geq 1$, the algorithm might produce a B' with $|B'| \geq c|OPT|$.

Solution

Given a star graph of $c+1$ vertices we have that the optimal solution is to choose the center ($|OPT| = 1$). Now suppose that for each edge the algorithm chooses the other vertex (that is not the center). In this way the algorithm chooses c different vertices, so $|B'| = c|OPT|$.
Since it is the worst case, we have that $|B'| \leq c|OPT|$.

Third Problem

We need to say why there is not much hope of deriving an efficient, deterministic version using the method of conditional expectation.

Solution

In order to apply the condition expectation to have a deterministic algorithm, we can do as follow:
When the edge e is not covered and we need to add one of its endpoints, we can analyze how each of the two endpoints impacts the expectation value of the solution $E[|B'|]$. I define a variable $Y = (v_1, v_2, ...)$ for each chosen edge e: it's equal to the vertex chosen.
Let's suppose that i edges are already covered and j vertices already added in B'. Now we have an edge $e = (u, v)$ to cover and we want to add what minimizes the expected $|B'| = S$:

$$E[|B'||y_1 = v_1, ..., y_i = v_j] = \frac{1}{2}E[|B'||y_1 = v_1, ..., y_i = v_j \wedge Y_{i+1} = u] + \frac{1}{2}E[|B'||y_1 = v_1, ..., y_i = v_j \wedge Y_{i+1} = v]$$

Now we can compute $E[|B'||y_1 = v_1, ..., y_i = v_j \wedge Y_{i+1} = v_{j+1}]$ as $j + 1 +$ all the vertices needed to cover the remaining edges. To compute the number of these uncovered edges, we can consider the already covered edges, the new added vertex and all his outgoing edges not covered:
$E - \{e_1, ..., e_i\} - \{e$ — e is outgoing vertex of $v_{j+1} \wedge y_e = 0\}$
The issue, that makes not efficient deterministic problem, is the difficulty and hard computation to find number of vertices needed to cover the uncovered edges.

Exercise 3

Problem

Given $\alpha \geq 0$ calculate Philips payoff at some mixed Nash equilibrium, for $\alpha < 1$ and $\alpha \geq 1$.

Solution for $\alpha < 1$

Let's define as q_r the probability that the opponent plays R, q_p the probability that the opponent plays P and $q_s = 1 - q_r - q_p$ the probability that the opponent plays S. Now we compute Philip's expected payoffs:

- Philip's expected payoff playing R: $\alpha \cdot q_r + (\alpha - 1) \cdot q_p + (\alpha + 1) \cdot (1 - q_r - q_p)$
- Philip's expected payoff playing P: $1 \cdot q_r + 0 \cdot q_p - 1 \cdot (1 - q_r - q_p) = 2q_r + q_p - 1$
- Philip's expected payoff playing S: $-1 \cdot q_r + 1 \cdot q_p + 0 \cdot (1 - q_r - q_p) = -q_r + q_p$

Since the strategy must be best responses, the payoffs must be equal, and we obtain:

$$\begin{cases} \alpha \cdot q_r + (\alpha - 1) \cdot q_p + (\alpha + 1) \cdot (1 - q_r - q_p) = 2q_r + q_p - 1 \\ \alpha \cdot q_r + (\alpha - 1) \cdot q_p + (\alpha + 1) \cdot (1 - q_r - q_p) = -q_r + q_p \\ 2q_r + q_p - 1 = -q_r + q_p \end{cases} \Longrightarrow \begin{cases} q_r = \frac{1}{3} \\ q_p = \frac{\alpha+1}{3} \\ q_s = 1 - q_r - q_p = \frac{1-\alpha}{3} \end{cases}$$

Since Philip is mixing in a Nash Equilibrium, the opponent must mix with weights $\left(\frac{1}{3}, \frac{\alpha+1}{3}, \frac{1-\alpha}{3}\right)$.
Now let's do the same for opponent's expected payoffs, defining as q'_r, q'_p and $q'_s = 1 - q'_r - q'_p$ the probability that Philip plays R, P or S respectively:

- Opponent expected payoff playing R: $0 \cdot q'_r - 1 \cdot q'_p + 1 \cdot (1 - q'_r - q'_p) = -2q'_p - q'_r + 1$
- Opponent expected payoff playing S: $1 \cdot q'_r + 0 \cdot q'_p - 1 \cdot (1 - q'_r - q'_p) = 2q'_r + q'_p - 1$
- Opponent expected payoff playing P: $-1 \cdot q'_r + 1 \cdot q'_p + 0 \cdot (1 - q'_r - q'_p) = -q'_r + q'_p$

Since the strategy must be best responses, the payoffs must be equal, and we obtain:

$$\begin{cases} -2q'_p - q'_r + 1 = 2q'_r + q'_p - 1 \\ -2q'_p - q'_r + 1 = -q'_r + q'_p \\ -2q'_r + q'_p - 1 = -q'_r + q'_p \end{cases} \Longrightarrow \begin{cases} q'_r = \frac{1}{3} \\ q'_p = \frac{1}{3} \\ q'_s = 1 - q_r - q_p = \frac{1}{3} \end{cases}$$

Since the opponent is mixing in a Nash Equilibrium, Philip must mix with weights $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$.
So the MNE is: $\left[\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \left(\frac{1}{3}, \frac{\alpha+1}{3}, \frac{1-\alpha}{3}\right)\right]$.
Finally we can calculate Philip's payoff as follow:

$$\alpha \cdot q_r \cdot q'_r + (\alpha-1) \cdot q_p \cdot q'_r + (\alpha+1) \cdot q_s \cdot q'_r + 1 \cdot q_r \cdot q'_p + 0 \cdot q_p \cdot q'_p + (-1) \cdot q_s \cdot q'_p + (-1) \cdot q_r \cdot q'_s + 1 \cdot q_p \cdot q'_s + 0 \cdot q_s \cdot q'_s = \frac{\alpha}{3}$$

Solution for $\alpha \geq 1$

In this case, looking at the table it's possible to see that simplifications can be made: for Philip, P is dominated by R because of $\alpha \geq 1$, so we can delete the relative row. After that, for the opponent S is dominated by R, so we can delete the relative column.

Ph/Op	R	P	S
R	α, 0	$(\alpha - 1)$, 1	$(\alpha + 1)$, -1
P	1, -1	0 , 0	-1 , 1
S	-1, 1	1 , -1	0 , 0

Let's define as q_r and $q_p = 1 - q_r$ the probability that the opponent plays R or P respectively, while as q'_r and $q'_s = 1 - q'_r$ the probability that Philip plays R or S respectively.

We have the following payoffs:

	Philip	Opponent
R	$\alpha \cdot q_r + (\alpha - 1) \cdot (1 - q_r) = q_r + \alpha - 1$	$0 \cdot q'_r + 1 \cdot (1 - q'_r) = 1 - q'_r$
P	none	$1 \cdot q'_r - 1 \cdot (1 - q'_r) = 2q'_r - 1$
S	$-1 \cdot q_r + 1 \cdot (1 - q_r) = -2q_r + 1$	none

Since the strategy must be best responses, the payoffs of each player must be equal, and we obtain: $q_r = \frac{2-\alpha}{3}$, $q_p = \frac{\alpha+1}{3}$, $q'_r = \frac{2}{3}$ and $q'_s = \frac{1}{3}$.
So the MNE is: $\left[\left(\frac{2}{3}, \frac{1}{3}\right), \left(\frac{2-\alpha}{3}, \frac{\alpha+1}{3}\right)\right]$.
Finally we can calculate Philip's payoff as follow:

$$\alpha \cdot q_r \cdot q'_r + (\alpha - 1) \cdot q_p \cdot q'_r + (-1) \cdot q_r \cdot q'_s + 1 \cdot q_p q'_s = \frac{2\alpha-1}{3}.$$

Exercise 4

Problem

Given n, the number of students, and assuming that each student a_i and b_i is uniformly distributed in $[0, 1]$, show that there are $\mathbf{O}(\log n)$ students worthy of an award, with high probability.

Solution

We can order the students according to a_i in descending order.
The first student will get the award for sure, since he has the highest a_i.
The other students will get the award if and only if $b_i > b_1$.
Now let's define with $X_i \in 0, 1$ the variable that describes if the student i gets the award, and with $X = \sum_{i=1}^{n} X_i$ the number of student that have a reward.
We have that $\mathbb{E}[X_i] = Pr[X_i] = \frac{1}{i}$.
So $\mathbb{E}[X] = \sum_{i=1}^{n} \frac{1}{i} = \ln n + \mathbf{O}(1)$.
Now let's apply Chernoff Bound:
$Pr[X > (1 + \delta)\,\mathbb{E}[X]] < e^{-\delta^2 \cdot \frac{\mu}{3}}$.
Choosing $\mu = \ln n$ and $\delta = \frac{1}{2}$, we have:
$Pr[X > (1 + \delta) \ln n] < e^{-\frac{1}{4} \cdot \frac{\mu}{3}} = \frac{1}{\sqrt[12]{e^{\ln(n)}}} = \frac{1}{\sqrt[12]{n}}$.
Finally, for high values of n this values tents to 0, so there are $\mathbf{O}(logn)$ students that will get an award with high probability.

Exercise 5

First Problem

We need to show that $|T1| + 2|T2| + 3|T3| = |E|(|V|2)$.

Solution

Given an edge $e = (u, v)$ of the graph $G = (V, E)$, we need to build a triple choosing any vertex in the set $V \setminus \{u, v\}$ For every edge $e \in E$ there could be three cases:

- $(e, w) = (u, v, w) \in T1$: count it one time;
- $(e, w) = (u, v, w) \in T2$: count it two times, since one time we have this triple as $(e, w) = (u, v, w)$, and another time when we have the edge (v, w) and the vertex u (or the edge (u, w) and the vertex v);
- $(e, w) = (u, v, w) \in T3$: count it three times, for the same reason as before.

Since the number of triple generated by the algorithm is $|E|(|V| - 2)$ because for every edge we test all possible $|V| - 2$ vertices, the triple in $T1$ are generated one time, in $T2$ two times and in $T3$ three times, we have that $|T1| + 2|T2| + 3|T3| = |E|(|V|2)$.

Second Problem

We need to show that $\frac{1}{s}\sum\limits_{i=1}^{s} x_i \frac{|E|(|V|-2)}{3}$ is an estimator of T3.

Solution

The expected value of the output value x of the algorithm is $\mathbb{E}[x] = \frac{3|T3|}{|T1|+2|T2|+3|T3|} = \frac{3|T3|}{|E|(|V|-2)}$.
So we have $|T3| = \mathbb{E}[x] \cdot \frac{|E|(|V|-2)}{3|T3|}$.
Now let's use $(\frac{1}{s}\sum\limits_{i=1}^{s} x_i)$ as an estimator for $\mathbb{E}[x]$. Finally we have that $\tilde{T}3 = \frac{1}{s}\sum\limits_{i=1}^{s} x_i \frac{|E|(|V|-2)}{3}$ is an estimator for $T3$.

Third Problem

We need to find a nontrivial number s of executions of the algorithm which are sufficient in order obtain an $(1 + \epsilon)$ and an $(1 - \epsilon)$ approximation of T3 with probability at least $1 - \delta$.

Solution

In other words, we need to show that we have $(1 - \epsilon)|T3| < \tilde{T}3 < (1 + \epsilon)|T3|$ with probability $(1 - \delta)$. Using Chernoff Bound we have:

- **Lower tail**: $\mathbf{Pr}[\frac{1}{s}\sum\limits_{i=1}^{s} x_i \leq (1 - \epsilon)\,\mathbb{E}[x]] \leq e^{-\frac{\mathbb{E}[x]\cdot\epsilon^2}{2}}$
- **Upper tail**: $\mathbf{Pr}[\frac{1}{s}\sum\limits_{i=1}^{s} x_i \geq (1 + \epsilon)\,\mathbb{E}[x]] \leq e^{-\frac{\mathbb{E}[x]\cdot\epsilon^2}{3}}$

For $s \geq \frac{3}{\epsilon^2} \cdot \frac{|T1|+2|T2|+3|T3|}{|T3|} \cdot \ln(\frac{2}{\delta})$ the sum of both probabilities is bounded by δ.