Complexity - Exercise 4

Oren Roth - 200701068

January 15, 2018

Question 1

a.

Let (S, T) be a random cut, where each node is chosen uniformly i.i.d to be in S or T. We will define indicator for each $(u, v) \in E$, if this edge is in the cut:

$$X_{(u,v)} = \begin{cases} 0 & u, v \in S \\ 0 & u, v \in T \\ 1 & \text{otherwise} \end{cases}$$

Define new random variable $X = |E(S, T)|$:

$$X = \sum_{e \in E} X_e$$

For $X_{(u,v)}$ half of the cases of choosing u, v their sets, give $X_{(u,v)}$ value 1, and we know u, v is drawn uniformly:

$$\mathbb{E}[X_e] = \frac{1}{2}$$

We calculate the expectation of X,

$$\mathbb{E}[X] = \mathbb{E}[\sum_{e \in E} X_e] = \sum_{e \in E} \mathbb{E}[X_e] = \frac{|E|}{2}$$

As the graph is finite, we have at least one cut with cut of size $\frac{|E|}{2}$, which gives us solution of value:

$$\frac{\frac{|E|}{2}}{|E|} = \frac{1}{2}$$

b.

Let $A \in NP - Complete$, we know $NP = PCP^{\neq}_{\alpha,\beta}(\log n, 2)$, and hence:

$$A \in PCP^{\neq}_{\alpha,\beta}(\log n, 2)$$

So we have verifier V for A which, for any input x of length n, uses $c_1 \cdot \log n$ coins. For each choice $\hat{r}$ of coins, it asks two queries $j_1^{\hat{r}}, j_2^{\hat{r}}$. For any proof π we know that the probability V will accept x is:

$$\Pr_{\hat{r} \in_R \{0,1\}^{c_1 \cdot \log n}} [\pi(j_1^{\hat{r}}) \oplus \pi(j_2^{\hat{r}}) = 1]$$

There are at most $2 \cdot 2^{c_1 \log n} = 2 \cdot n^{c_1}$ different values of queries $j_i^{\hat{r}}$ that V can ask ($2^{c_1 \log n}$ options for the coins and 2 for the queries). So we can create the following graph $G = (V, E)$ in polynomial time in n:

$$V = \{1, 2, \ldots, 2 \cdot n^{c_1}\}$$
$$E = \{(j_1^{\hat{r}}, j_2^{\hat{r}}) : \hat{r} \in \{0,1\}^{c_1 \cdot \log n}\}$$

There are $poly(n)$ vertices and edges in G. We can think of a proof π as partition of V to a cut (S, T) like this:

$$\pi(j_i^{\hat{r}}) = 0 \Rightarrow j_i^{\hat{r}} \in S$$
$$\pi(j_i^{\hat{r}}) = 1 \Rightarrow j_i^{\hat{r}} \in T$$

And the probability that V will accept x ($\Pr_{\hat{r} \in_R \{0,1\}^{c_1 \cdot \log n}}[\pi(j_1^{\hat{r}}) \oplus \pi(j_2^{\hat{r}}) = 1]$) is exactly the number of edges crossing a cut define by π, divide by the number of edges. So if we define OPT to be the value of $Max\ Cut$ problem, we are having,

$$OPT(G) = \max_{\pi}\{\Pr[\pi \text{ make } V \text{ accept } x]\}$$

By the definition of $PCP^{\neq}_{\alpha,\beta}(\log n, 2)$ we know:

- If $x \in A$ we have G with $OPT \geq \beta$
- If $x \notin A$ we have G with $OPT \leq \alpha$

And we showed a gap reduction for an $NP-complete$ problem to $Max\ Cut$. So we conclude, by the notice we showed in class, that for every $\gamma > \frac{\alpha}{\beta}$, there is no $\gamma - apprixmation$ algorithm to $Max\ Cut$ unless $P = NP$.

c.

Let $A \in PCP^{\neq}_{\alpha,1}(\log n, 2)$, so A has a verifier V which, for any input x of length n, uses $c_1 \cdot \log n$ coins. Let be x an input, there are $2^{c_1 \log n} = n^{c_1}$ computation paths for V on x (for each choice of coins). In each path, V choose two queries. Denote the pair of queries of all computational paths:

$$\{< X_i^1, X_i^2 >\}_{i=1}^{n^{c_1}}$$

By the definition of $PCP^{\neq}_{\alpha,1}(\log n, 2)$, $x \in A$ if and only if there is a proof π which satisfy:

$$\forall 1 \leq i \leq n^{c_1} : \quad \pi(X_i^1) \oplus \pi(X_i^2) = 1$$

Existence of π which satisfy the above is equivalent to check if the following $2 - SAT$ is satisfied:

$$\psi = \bigwedge_{i=1}^{n^{c_1}} [(X_i^1 \vee X_i^2) \wedge (\overline{X_i^1} \vee \overline{X_i^2})]$$

Due to the equality:

$$X \oplus Y = 1 \iff (X \vee Y) \wedge (\overline{X} \vee \overline{Y}) = 1$$

Checking whereas ψ is satisfiable, can be checked in polynomial time ($2 - SAT \in P$) and which will determine if $x \in A$.
Conclusion: For a given x, we can reduce it to $2 - SAT$ in polynomial time (going over $poly(n)$ calculation paths and build $poly(n)$ clauses to create ψ). As $2 - SAT \in P$ we conclude $A \in P$.

Question 2

a.

Define L_i as all the nodes in distance i from s in G'. So for $i \neq j$ we know $L_i \cap L_j = \emptyset$, we conclude:

$$\sum_{i=0}^{k} |L_i| \leq |V'|$$

We can use it to conclude the last inequality here:

$$k \cdot \min_{i=1}^{k}\{|L_{i-1}| + |L_i|\} \leq \sum_{i=1}^{k}\{|L_{i-1}| + |L_i|\} = |L_0| + |L_k| + 2 \cdot \sum_{i=1}^{k-1}\{|L_i|\} \leq$$
$$\leq 2 \cdot \sum_{i=0}^{k}\{|L_i|\} \leq 2 \cdot |V'|$$

Which from this we conclude:

$$\min_{i=1}^{k}\{|L_{i-1}| + |L_i|\} \leq \frac{2 \cdot |V'|}{k}$$

Let $j \in argmin_{i=1}^{k}\{|L_{i-1}| + |L_i|\}$, we know:

$$|L_{j-1}| + |L_j| \leq \frac{2 \cdot |V'|}{k}$$

So we conclude:

$$(|L_{j-1}| + |L_j|)^2 \leq \frac{4 \cdot |V'|^2}{k^2} \quad \Rightarrow$$
$$|L_{j-1}|^2 + |L_j|^2 + 2|L_{j-1}| \cdot |L_j| \leq \frac{4 \cdot |V'|^2}{k^2}$$

But we know:

$$(|L_{j-1}| - |L_j|)^2 \geq 0 \quad \Rightarrow$$
$$|L_{j-1}|^2 + |L_j|^2 - 2|L_{j-1}| \cdot |L_j| \geq 0 \quad \Rightarrow$$
$$|L_{j-1}|^2 + |L_j|^2 \geq +2|L_{j-1}| \cdot |L_j|$$

And together with the last two inequalities we conclude:

$$4|L_{j-1}| \cdot |L_j| \leq \frac{4 \cdot |V'|^2}{k^2} \quad \Rightarrow$$
$$|L_{j-1}| \cdot |L_j| \leq \frac{|V'|^2}{k^2}$$

Define cut $F = \{(u, v) : u \in L_{j-1},\ v \in L_j\}$, we have at most L_{j-1} times L_j edges in F, together with previous inequality we get:

$$|F| \leq |L_{j-1}| \cdot |L_j| \leq \frac{|V'|^2}{k^2}$$

Because any path from s to t passing through the cut F, F is separating s and t.

b.

Let F be the solution the algorithm return, we denote:

$$F = F_1 \cup F_2$$

Where F_1 are the set of edges which have been deleting in the while loop (refer as **first step**), and F_2 the set of edges deleting after (refer as **second step**). Denote F_{OPT} the optimal solution. We want to show:

$$|F| \leq 2|V|^{\frac{2}{3}} \cdot |F_{OPT}|$$

In the first step we deleted edges in paths, denote those k paths:

$$F_1 = \{p_1, p_2, \dots, p_k\}$$

Denote that two different paths in F_1 are disjoint, as we delete all the edges of one before choosing the next path. Each path p_i has to intersect with at least one edge of F_{OPT}, as if this is not the case we will have a path smaller than L still in G after deleting F_{OPT} and contradiction to F_{OPT} being a legal solution. As the while loop goes on up to length $|V|^{\frac{2}{3}}$ we are sure $p_i \leq |V|^{\frac{2}{3}}$ for all $1 \leq i \leq k$. Summaries it up:

$$|F_1| = \sum_{i=1}^{k} |p_i| \leq k \cdot |V|^{\frac{2}{3}} \leq |F_{OPT}| \cdot |V|^{\frac{2}{3}}$$

If $L \leq |V|^{\frac{2}{3}}$ we will not do step 2, and thus $|F_2| = 0$, and $|F| = |F_1|$ is a legal solution which will achieve:

$$|F| = |F_1| \leq |F_{OPT}| \cdot |V|^{\frac{2}{3}}$$

Otherwise, $L > |V|^{\frac{2}{3}}$ so we do step 2, and by the last bound we showed in [2a] we know:

$$|F_2| \leq \frac{|V|^2}{(|V|^{\frac{2}{3}})^2} = |V|^{\frac{2}{3}}$$

And we got,

$$\begin{aligned} &|F| = |F_1| + |F_2| \leq |F_{OPT}| \cdot |V|^{\frac{2}{3}} + |V|^{\frac{2}{3}} = \\ &|V|^{\frac{2}{3}}(|F_{OPT} + 1) \leq |V|^{\frac{2}{3}}(|F_{OPT} + |F_{OPT}|) = 2|V|^{\frac{2}{3}} \cdot |F_{OPT}| \end{aligned}$$

As in order to arrive to step 2, it has to be that $|F_{OPT}| \geq 1$. By [2a] we sure that deleting F will have no longer path between s and t so F is legal solution, which achieve the desired approximation ratio.