18.218 Topics in Theoretical Computer Science Fall 2022
Problem Set 3

1. In this question, we show the equivalence between the verifier view of PCPs and the combinatorial view.

   (a) Suppose that gap-CSG$[1, \delta]$ is NP-hard on instances of q queries and alphabet size h. Show that there exists a probabilistic verifier for 3-SAT that on a 3CNF formula ϕ and a witness w, uses $O(\log n)$ random bits and reads at most $q \log h$ bits from the witness w such that:
   Completeness: If $\phi \in$ 3-SAT, then there is a w such that the verifier accepts with probability 1.
   Soundness: If $\phi \notin$ 3-SAT, then for all w, the probability that the verifier accepts is at most δ.

   (b) Show the other direction. Namely, show that if 3-SAT has a probabilistic verifier for 3-SAT as above reading at most q bits from the witness, then gap-CSG$[1, \delta]$ is NP-hard on instances of q queries and alphabet size 2.

2. In this question, we show that gap-3SAT$[1, 1 - \varepsilon]$ is NP-hard for some absolute constant $\varepsilon > 0$.

   (a) Let $P\colon \{0,1\}^k \to \{0,1\}$ be any function whose input is $x \in \{0,1\}^k$. Show that there exists a k-CNF formula $\phi\colon \{0,1\}^k \to \{0,1\}$ such that $\phi(x) = P(x)$ for all $x \in \{0,1\}^k$.

   (b) Let $P\colon \{0,1\}^k \to \{0,1\}$ be any function whose input is $x \in \{0,1\}^k$. Show that there exists a 3-CNF formula $\phi\colon \{0,1\}^{k+m} \to \{0,1\}$ of size at most $2^{O(k)}$ satisfying:
   Completeness: if $x \in \{0,1\}^k$ is such that $P(x) = 1$, then there is $y \in \{0,1\}^m$ for which $\phi(x, y) = 1$.
   Soundness: if $x \in \{0,1\}^k$ is such that $P(x) = 0$, then for all $y \in \{0,1\}^m$, $\phi(x, y) = 0$.

   (c) Assuming the PCP theorem, namely that gap-CSG$[1, 1 - \varepsilon]$ is NP-hard on instances with $O(1)$ queries, alphabet size 2 and $\varepsilon > 0$ absolute constant, show that there is $\varepsilon' > 0$ such that gap-3SAT$[1, 1 - \varepsilon']$ is NP-hard.

3. In the next two problems, we convert the PCP with poly-logarithmically many queries to a corresponding hardness result about Quadratic Solvability with poly-logarithmically many queries.

   Circuits. Recall that a circuit $C(x_1, \ldots, x_m)$ on m variables is a directed acyclic graph. m of its nodes have in-degree 0 and are labeled by input variables x_i; any other node is labeled by one of: OR, AND, NOT. A circuit has a unique node of out-degree 0, which is the output of the circuit; each node labeled by OR or AND has in-degree 2, and each node labeled by NOT has in-degree 1. On input $x \in \{0,1\}^m$, the value is computed from the leafs (the input nodes) to the root. The value of a node is the logical value of the operation labeling it when applied on the Boolean values of its children. The size of a circuit C is defined to be the number of nodes plus the number of edges in the graph.

   Let $P\colon \{0,1\}^m \to \{0,1\}$ be a function computable by a circuit C of size at most $\mathsf{poly}(m)$. Show that there is a system of quadratic equations (X, E) in x and $m' = \mathsf{poly}(m)$ auxiliary variables y that has $\mathsf{poly}(m)$ many equations such that for every $x \in \{0,1\}^m$,

- If $P(x) = 1$, then there is $y \in \{0,1\}^{m'}$ such that x, y solve (X, E).
- If $P(x) = 0$, then for all $y \in \{0,1\}^{m'}$ it holds that x, y does not solve (X, E).

4. We are now going to show a reduction from CSG to QS that preserves poly-log query complexity.
   (a) Starting with the fact that for all $\varepsilon > 0$ the problem gap-CSG$[1, \varepsilon]$ is NP-hard for instances with $\mathsf{poly}(\log n)$ queries and alphabet size $\mathsf{poly}(\log n)$, show that for all $\delta > 0$ the problem gap-CSG$[1, \delta]$ is NP-hard on instances with $\mathsf{poly}(\log n)$ queries and alphabet size 2.
   (b) Show that for all $\varepsilon > 0$ there is $q > 0$ such that the problem gap-QS$_{q,r=\mathsf{poly}(\log n)}[1, \varepsilon]$ is NP-hard. Here, q is the field size and r is the number of variables appearing in each equation.

5. In this question we show an improved hardness of approximation result for clique via randomized reductions. Let $\Psi = (G = (L \cup R, E), \Sigma_L, \Sigma_R, \Phi = \{\Phi_e\}_{e \in E})$ be a label cover instance with alphabet size at most C, and denote $t = 100C \log n$. Let $\varepsilon > 0$ and denote $M = \lceil 10^t (1-\varepsilon)^{-t} \rceil$. Consider the following randomized set of constraints on $L \cup R$: for $i = 1, \ldots, M$, sample t of the edges $e_1, \ldots, e_t \in E$ randomly, and let the constraint C_i be the AND of the constraints $\Phi_{e_1}, \ldots, \Phi_{e_t}$.
   (a) Show that if Ψ is satisfiable, then there is an assignment to $L \cup R$ satisfying all of the C_i's.
   (b) Show that if Ψ is at most $(1-\varepsilon)$-satisfiable, then with probability $1 - o(1)$ at most $2(1-\varepsilon)^t M$ of the constraints $C_1, \ldots, C_M$ can be satisfied.
   (c) Construct the graph $G = (V, E)$ from the constraints $C_1, \ldots, C_M$ as in class: for each $i = 1, \ldots, M$ and a satisfying assignment α to the variables of C_i, construct a node $v_{i,\alpha}$ in G. Connect $v_{i,\alpha}$ and $v_{j,\beta}$ by an edge if they are consistent on all variables shared between C_i and C_j.
      i. Show that if Ψ is satisfiable, then G contains a clique of size at least M.
      ii. Show that if Ψ is at most $(1-\varepsilon)$ satisfiable, then with probability $1 - o(1)$ the graph G doesn't contain a clique of size $2(1-\varepsilon)^t M$.
      iii. Show that the graph G has at most $N = C^t M$ nodes.
      iv. Conclude that there is an absolute constant $\delta > 0$, such that approximating the Maximum-Clique problem on N vertices within factor N^δ is NP-hard under polynomial time randomized reductions.

6. (*) In the next problem, we will complete the missing ingredient in the proof of the PCP theorem by explaining the composition step that reduces from $\mathsf{poly}(\log n)$ queries to $\mathsf{poly}(\log \log n)$ queries.

   Explain the steps of a reduction from instances of gap-CSG$[1, 1-\varepsilon]$ with $\mathsf{poly}(\log n)$ queries and alphabet size 2, to the problem gap-CSG$[1, 1-\varepsilon']$ with $\mathsf{poly}(\log \log n)$ queries with alphabet size $\mathsf{poly}(\log \log n)$, where $\varepsilon' > 0$ is an absolute constant depending only on ε and k.

   In your answer you should explain how the construction works, and explain how each part of it relates to things we have seen in class, as well as the necessary modifications (in high level). You do not have to give fully detailed proofs, but are encouraged to give a rough overview of how a proof would work.

18.408 Topics in Theoretical Computer Science: Probabilistically Checkable Proofs
Fall 2022