Paul Pham
ppham@cs.washington.edu
CSE 533 Problem Set 1

Problem 1 Hardness-of-approximation reductions

a) Perfect Completeness of Håstad PCP. We define a *Håstad* (c, s)*-PCP-system* as a PCP system for NP that makes 3 queries from the alphabet $\{0, 1\}$ with predicates ϕ of the form $x_{i_1} + x_{i_2} + x_{i_3} = b \mod 2$, completeness c, and soundness s.

To show that a Håstad $(1, \frac{1}{2} + \epsilon)$-PCP-system does not exist unless $\mathrm{P} = \mathrm{NP}$, it suffices to show that such a system would give a deterministic polytime algorithm for E3SAT, which is NP-complete.

For the language E3SAT, the verifier and its predicates ϕ reduce proofs to assignments to Boolean variables in a E3CNF formula ψ as part of their deterministic polytime computation. Therefore the three chosen proof bits x_i correspond to the truth assignment to three variables whose literals appear in the same E3CNF clause, chosen randomly from within ψ.

Having perfect completeness means that for all random coin tosses and choices of proof bits, $\psi \in$ E3SAT $\Rightarrow \Pr[\phi(x_{i_1}, x_{i_2}, x_{i_3})\text{accepts}] = 1$. Then we will construct an E3SAT solver.

For $r = O(\log n)$ random bits, enumerate all $2^r = O(n)$ possible random choices of bits and their corresponding predicates in deterministic polytime. This will cover all E3CNF clauses in ψ. Run each ϕ on its corresponding clause. If all ϕ's accept, then the overall verifier should accept, otherwise it should reject. This verifier is an E3SAT solver. □

b) Hardness of MAX-3LIN. Using a Håstad $(1 - \epsilon, \frac{1}{2} + \epsilon)$-PCP-system, we can solve MAX-3LIN directly by using the verifier to enumerate over all random coin tosses and generate predicates to check every 3LIN equation in our system. A PCP system solving a gap decision problem with completeness c and soundness s gives an $\frac{s}{c}$-approximation algorithm for the corresponding optimization problem.

For the above Håstad PCP system, we have:

$$\begin{aligned}
\frac{s}{c} &= \\
\frac{\frac{1}{2}+\epsilon}{1-\epsilon} &= \\
\frac{\frac{1}{2}-\frac{\epsilon}{2}}{1-\epsilon} + \frac{\frac{3}{2}\epsilon}{1-\epsilon} &= \\
\frac{\frac{1}{2}(1-\epsilon)}{1-\epsilon} + \frac{\frac{3}{2}\epsilon}{1-\epsilon} &= \\
\frac{1}{2} + \frac{\frac{3}{2}\epsilon}{1-\epsilon} &= \\
\frac{1}{2} + \epsilon'
\end{aligned}$$

This gives us a $(\frac{1}{2} + \epsilon')$-approximation algorithm for all $\epsilon' > 0$, a new positive constant we have defined in terms of the PCP constant ϵ above. The prover for this PCP runs in nondeterminisic polytime because it can "guess" an optimal assignment which the verifier will accept. If $\mathrm{P} = \mathrm{NP}$, then this PCP system gives a polytime approximation algorithm for MAX-E3LIN. □

c) Hardness of MAX-E3SAT. We show the hardness of a $(\frac{7}{8}+\epsilon)$-approximation algorithm for MAX-E3SAT by reduction from MAX-3LIN.

First, note that of the eight possible truth assignments to three Boolean variables, there are four with even parity and four with odd parity.

$b = 0$	$b = 1$
$000, 011, 110, 101$	$111, 001, 010, 100$

Given as input a system of 3LIN equations, convert each equation $x_{i_1}+x_{i_2}+x_{i_3} = b \mod 2$ into four E3CNF clauses (gadgets) according to the parity bit b as follows.

$b = 0$	$b = 1$
$(\overline{x_{i_1}} \vee \overline{x_{i_2}} \vee \overline{x_{i_3}})$	$(x_{i_1} \vee x_{i_2} \vee x_{i_3})$
$(\overline{x_{i_1}} \vee x_{i_2} \vee x_{i_3})$	$(\overline{x_{i_1}} \vee \overline{x_{i_2}} \vee x_{i_3})$
$(x_{i_1} \vee \overline{x_{i_2}} \vee x_{i_3})$	$(\overline{x_{i_1}} \vee x_{i_2} \vee \overline{x_{i_3}})$
$(x_{i_1} \vee x_{i_2} \vee \overline{x_{i_3}})$	$(x_{i_1} \vee \overline{x_{i_2}} \vee \overline{x_{i_3}})$

If a 3LIN equation system is satisfiable, then for every equation the corresponding assignment will have the right parity b and satisfy all four clauses above for that parity. This gives the reduction a completeness of 1.

If a 3LIN equation system is unsatisfiable, then all assignments will have the wrong parity for at least one equation. All assignments of odd parity will fail exactly one clause above for even parity, and all assignments of even parity will likewise fail exactly one clause above for odd parity. This means $\frac{1}{4}$ of the E3CNF clauses cannot be satisfied, and by the soundness of the Håstad PCP system, $(\frac{1}{2}-\epsilon)$ fraction of these fail the predicate check. This gives an overall soundness of:

$$\begin{aligned} 1-\tfrac{1}{4}\left(\tfrac{1}{2}-\epsilon\right) &= \\ 1-\tfrac{1}{8}+\epsilon &= \\ \tfrac{7}{8}+\epsilon \end{aligned}$$

From the calculation in part (a), we know that this completeness and soundness give us a $(\frac{7}{8}+\epsilon')$-approximation algorithm for MAX-E3SAT for all $\epsilon' > 0$, where ϵ' is defined in terms of the original PCP's ϵ. This algorithm does not run in polytime unless $\mathrm{P} = \mathrm{NP}$. $\square$

d) Hardness of MAX-3MAJ. We show the hardness of a a $(\frac{2}{3}+\epsilon)$-approximation algorithm for MAX-3MAJ by reduction from MAX-E3LIN.

Given as input a system of 3LIN equations, convert each equation $x_{i_1}+x_{i_2}+x_{i_3} = b \mod 2$ into four 3MAJ constraints (gadgets) according to the parity bit b as follows.

$b = 0$	$b = 1$
Maj$(x_{i_1}, x_{i_2}, x_{i_3}) = 1$	Maj$(\overline{x_{i_1}}, \overline{x_{i_2}}, \overline{x_{i_3}}) = 1$
Maj$(x_{i_1}, \overline{x_{i_2}}, \overline{x_{i_3}}) = 1$	Maj$(\overline{x_{i_1}}, x_{i_2}, x_{i_3}) = 1$
Maj$(\overline{x_{i_1}}, \overline{x_{i_2}}, x_{i_3}) = 1$	Maj$(x_{i_1}, x_{i_2}, \overline{x_{i_3}}) = 1$
Maj$(\overline{x_{i_1}}, x_{i_2}, \overline{x_{i_3}}) = 1$	Maj$(x_{i_1}, \overline{x_{i_2}}, x_{i_3}) = 1$

If a 3LIN equation is satisfiable, then the corresponding assignment will have the right parity b and satisfy exactly three of the constraints above for that parity (and fail one). This gives the reduction a completeness of $\frac{3}{4}$.

If a 3LIN equation is unsatisfiable, then it will have the wrong parity. All assignments of odd parity will satisfy exactly one clause above for even parity, and all assignments of even parity will likewise satisfy exactly one clause above for odd parity. This means at most (in fact, exactly) $\frac{1}{4}$ of the 3MAJ constraints are satisfied, giving the reduction a soundness of $\frac{1}{4}$. $\square$

Problem 2 2-Prover 1-Round Games

a) Randomized Provers. We would like to show that randomness does not increase the probability of provers winning a 2P1R by comparing the expectation of the game's value in the first case, when the provers are not given randomness, and in the second case, when they are.

Because the provers P_1 and P_2 are all-powerful, they can simulate V before the game starts on all possible random coin flips. We define a random variable to be the value of a game where V has random string r, P_1 gives optimal answer a_1 with randomness r_1, and P_2 gives optimal answer a_2 with randomness r_2.

$$X_{r,r_1,r_2} = \Pr\left[(q_1, q_2) \leftarrow V(r); a_1 \leftarrow P_1(q_1, r_1); a_2 \leftarrow P_2(q_2, r_2); V'(a_1, a_2) = 1\right] \tag{1}$$

The expected value of provers winning the game with an optimal strategy is taken over random coin flips of the verifier, which determines the questions and the provers' answers. In the first case, the provers are not allowed any randomness, denoted by $X_{r,0,0}$.

$$\mathop{\mathrm{E}}_{r}\left[X_r\right] = \sum_{r} \Pr\left[r\right] X_{r,0,0} \tag{2}$$

In the second case, the provers are given random strings r_1 and r_2 after the game has started. The new expectation of winning is:

$$\mathop{\mathrm{E}}_{r,r_1,r_2}\left[X_{r,r_1,r_2}\right] = \sum_{r} \Pr\left[r \wedge r_1 \wedge r_2\right] X_{r,r_1,r_2} \tag{3}$$

Since in the best case for the provers, the choices of randomness are independent, 3 can be written as:

$$\sum_{r,r_1,r_2} \Pr\left[r_1\right] \Pr\left[r_2\right] \Pr\left[r\right] X_{r,r_1,r_2} \tag{4}$$

For a given choice of verifier random string r, we can see that the provers can always pre-compute or guess optimal random strings r'_1 and r'_2 which are at least as good in expectation as using any given random string.

$$\sum_{r_1,r_2} \Pr[r_1]\Pr[r_2]X_{r,r_1,r_2} \leq X_{r,r'_1,r'_2} \tag{5}$$

Therefore 3 will always be less than or equal to 2, and it does not help the provers to have randomness.

b) Bipartite GAP-CG$_1, \epsilon(\Sigma)$ **is NP-hard.**

In lecture we showed that GAP-CG$_{1,\epsilon}(\sigma)$ is NP-hard and that 2P1R games are equivalent to bipartite constraint-graph problems. We can then reduce any constraint graph problem to a bipartite version, which can then be solved by some 2P1R game. This shows how a 2P1R game can decide any language $L \in \mathrm{NP}$ with the required completeness and soundness.

Given an input to GAP-CG$_{1,\epsilon}(\sigma)$, $G = (V, E)$ and constant-size For every vertex $x \in G$, create two vertices x_1 and x_2 in G' connected by an equality constraint to enforce the reduction. For every edge (u, v), create two "diagonal" edges (u_1, v_2) and (u_2, v_1) with the same constraint.

If $x \in L$ and all constraints in G are satisfied, then all constraints in G' will also be satisfied, giving completeness 1. Each constraint that is unsatisfied in G must fail one of the diagonal edges in G', $\frac{1}{4}$ of the constraints. If $x \notin L$ then at most ϵ fraction of constraints are satisfied in G and therefore $\epsilon' = \frac{\epsilon}{4}$ constraints unsatisfied in G'.

Therefore if there exists a 2P1R game to solve the gap problem for any bipartite constraint graph, all constraint graphs can be made bipartite and their gap problem is NP-hard, then a 2P1R game can solve any gap problem for languages in NP. □

Problem 7 A long code test

a) In general, a local test for a function set C may not be a local test for a subset $C' \subseteq C$ because the distribution of functions may be different in the subset. For example, consider the function $f : \{1, -1\}^n \to 1$ that maps every input to 1. It is at least $\frac{1}{2}$-far from every function in $\mathcal{D}$, since the dictator bit is always balanced (half 1's and half -1's). However, it will always pass the BLR test.

b)

$$\frac{3}{4} - \frac{ab + bc + ca}{4}$$

If $a = b = c$ then $ab = bc = ca = 1$ and the expression is 0. If $a \neq b$ or $b \neq c$, then one of $\{ab, bc, ca\}$ is 1 and the other two are-1, so the expression is 1.