PCPs and Inapproximability
Gap-producing and Gap-Preserving Reductions

My T. Thai

1 Hardness of Approximation

Consider a maximization problem Π such as MAX-E3SAT. To show that it is NP-hard to approximation Π to an approximation ratio ρ for all ϵ, we need to define the GAP-$\Pi_{c,s}$ where $\rho = s/c$ and prove that GAP-$\Pi_{c,s}$ is NP-hard.

How to prove a gap problem is NP-hard? Basically, there are two main methods: (1) Gap-producing reduction and (2) Gap-preserving reduction.

2 Gap-Producing Reduction

2.1 Max-Clique

In the previous lecture note, we saw an example of proving GAP-MAX-E3SAT$_{c,s}$ using a gap-producing reduction. Now, let us consider another example: Max-Clique problem.

Definition 1 GAP-Max-Clique$_{c,s}$: *Given $0 < s \leq c \leq 1$, a graph G and an positive integer k, the YES and NO instances of GAP-Max-Clique$_{c,s}$ are defined as follows:*

- *Output YES: if $\omega(G) \geq ck$, that is G has a clique of size ck*
- *Output NO: if $\omega(G) < sk$, that is G has no clique of size $\geq sk$.*

where $\omega(G)$ is the size of the maximum clique in G.

Again, if GAP-Max-Clique$_{c,s}$ is NP-hard, then approximating Max-Clique to s/c is NP-hard for any ϵ.

To show a gap-problem NP-hard via gap-producing reduction, we need to reduce from a NP-hard problem L to the gap-problem in polynomial time. From the PCP theorem, we know that for any NP-complete L, there exists an $(O(\log n), O(1))$-restricted verifier V. Use V as the sub-routing to construct a gap-producing reduction.

For example, consider the GAP-Max-Clique$_{c,s}$ problem and consider an NP-hard problem L. We need to construct a reduction R such that for each input $x \in L$, R will construct an instance (G, k) of GAP-Max-Clique$_{c,s}$ satisfying the following two constraints:

- If $x \in L$, then $\omega(G) \geq ck$
- If $x \notin L$, then $\omega(G) < sk$

Lemma 1 *[1] If 3SAT $\in PCP_{c,s}[r, q]$, then there exists a deterministic reduction running in time poly(2^{r+q}) reducing 3SAT to GAP-Max-Clique$_{c,s}$*

Before proving Lemma 1, we would like to show the following construction.

Consider $\varphi \in E3SAT$, let φ be on n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$. We create m rows of vertices, one for each clause. For each row, there are 7 vertices corresponding to all the satisfying assignments on that respective clause. Note that there are 7 partial assignments on 3 variables that make a clause satisfiable (and 1 make it unsatisfiable). For example, consider the clause $(x_1, x_2, \bar{x_3})$, then the seven vertices correspond to the following assignments: 000, 010, 011, 100, 101, 110, 111.

Now, let see how we assign edges on these vertices. Each row in the graph will be an independent set. Edges between rows are joining vertices whose assignments are consistent. The consistent assignments are defined as follows: (1) two vertices shared a variable which has the same assignment; (2) two vertices with disjoint set of variables. For example, consider clause $C_1 = (x_1, x_2, \bar{x_3})$ and clause $C_2 = (x_1, \bar{x_4}, x_5)$. Let vertex u in row 1 correspond to an assignment 011 for C_1 and v in row 2 correspond to an assignment 000 for C_2, then there is an edge (u, v) as they share the same assignment on x_1.

Observation: Since every row is an independent set, it is easy to see that the largest clique in the graph will be of size at most m (spanning all rows). If $\varphi \in E3SAT$, it is easy to see that by picking one vertex from each row corresponding to satisfying assignment, thus we will form a clique of size m.

Now, putting in the language of verifier V, how can you prove Lemma 1 using the above ideas?

Proof. If 3SAT $\in$ PCP$_{c,s}[r, q]$, then there exists a (r, q)-restricted verifier V satisfying the 2 conditions (completeness and soundness). Now, we are using V to reduce an instance $\varphi \in 3SAT$ to an instance (G, k) of Gap-Max-Clique$_{c,s}$.

By now, you should know what V supposes to do (from R random strings of r random bits, query q locations on proof π to accept or reject....) We call the q bits that V queries in π is the view. Note that there are at most 2^q views for each random string R (and there are 2^r random strings).

We are now ready to give the description of the graph G. The graph consists of $R = 2^r$ rows of vertices, each row with $\leq 2^q$ vertices. The 2^r rows correspond to 2^r random strings and the vertices on each row correspond to the possible 2^q views on queried bits that make the verifier V accept. We place edges between vertices similar to the above construction. That is, two vertices share the consistent partial views will form an edge. Note that there is no edge within a row. Two rows that query disjoint sets of bits have a complete bipartite graph between them.

Now we need to show:

- Completeness: If $\varphi \in 3SAT$, (exists π such that $\text{Prob}[V(x,\pi) = 1] \geq c$), then $\omega(G) \geq c2^r$. This is easy to prove. Note that a clique in G correspond to the accepting view of the same proof π and their size corresponds to the number of random strings of V for which it accepts this proof. Also note that the views are consistent. Since V accepts φ with probability $\geq c$, the size of clique, i.e. $\omega(G)$ must be $\geq c2^r$.
- Soundness: If $\varphi \notin 3SAT$, (for any π, $\text{Prob}[V(x,\pi) = 1] \leq s$), then $\omega(G) < s2^r$. Prove this by contradiction. Assume that there exists $\omega(G) \geq s2^r$. Note that the views for all vertices in this clique are consistent (else, we don't have edges between them). Then we can construct a proof π from those assignments on this clique such that verifier can accept φ with a probability $> s$, contracting to the fact that $\varphi \notin 3SAT$.

$\square$

Question: Can we pick 3-COLOR instead of 3SAT? Can we pick any language in NP?

The above Lemma implies the following:

Theorem 1 *If $NP \subseteq PCP_{c,s}[r,q]$ and $2^{r+q} = poly(n)$, then we cannot approximate Max-Clique to a ratio $s/c + \epsilon$ for any $\epsilon > 0$.*

Question: Why in the above Lemma, we require $2^{r+q} = poly(n)$. (Hint: How many nodes in the above constructed G?)

Notice that this proof always carries the ratio s/c. How do we prove that there does not exist any ρ-approximation algorithm for Max-Clique for all $\rho > 0$

We can run PCP verifier k times independently and accept only if all k views are accepting, then we have $PCP_{c,s}[r,q] \subseteq PCP_{c^k,s^k}[kr,kq]$

From the PCP theorem, we have $NP \subseteq PCP_{1,1/2}[O(\log n), O(1)]$. Run it $\log_2(1/\rho)$ times independently, we have $NP \subseteq PCP_{1,\rho}[O(\log n), O(1)]$. Then apply the above proof, we can easily obtain the following theorem:

Theorem 2 $\forall \rho > 0$, *gap-clique*$_\rho$ *is NP-hard.*

Question: Can you show the following:

There exists a $\delta > 0$, gap-clique$_{n^{-\delta}}$ is NP-hard. That is, approximating max-clique to a factor better than $n^{-\delta}$ is NP-hard.

Hint: We can choose k to be super-constant. But the running time of V no longer polynomial (becomes supper-polynomial). Can we derandomize it?

The inapproximability factor of max-clique can be improved to $n^{-(1-\epsilon)}$ for any ϵ due to the work of [2]. We will cover this part in the second half of this semester.

2.2 Max k-Function SAT

There is a gap-producing reduction from SAT to MAX k-FUNCTION SAT using V as the sub-routine. This problem will be presented by Thang N. Dinh.

3 Gap-Preserving Reduction

Another way to prove the hardness of approximation is to reduce a known gap-version of problem Π_1 to the gap-version of problem Π_2 which we are considering. This reduction is called a gap-preserving reduction from Π_1 to Π_2. Depending on whether Π_1 and Π_2 are minimization or maximization problems, we have slightly different definitions.

Let Π_1 and Π_2 be the maximization problems. A gap-preserving reduction from Π_1 to Π_2 is a polynomial time algorithm with given an instance $x \in \Pi_1$, produces an instance $y \in \Pi_2$ such that if

- $OPT(x) \geq h(x)$, then $OPT(y) \geq h'(y)$
- $OPT(x) < g(|x|)h(x)$, then $OPT(y) < g'(|y|)h'(y)$

for some functions $h(x), g(|x|), h'(y), g'(|y|)$ with $g(|x|), g'(|y|) \leq 1$

Likewise, if Π_1 is a minimization problem and Π_2 is the maximization problem. We have:

- $OPT(x) \leq h(x)$, then $OPT(y) \geq h'(y)$
- $OPT(x) > g(|x|)h(x)$, then $OPT(y) < g'(|y|)h'(y)$

for some functions $h(x), g(|x|), h'(y), g'(|y|)$ with $g(|x|) \geq 1$ and $g'(|y|) \leq 1$

You can produce the other two cases.

We now consider several examples, increasing in their difficulties.

3.1 Independent Set (IS)

Let us first consider this simple example. Let $G = (V, E)$ be an undirected graph. An IS set of G is a set $S \subseteq V$ such that for every pair of vertices $u, v \in S$, the edge $(u, v) \notin E$. We will see that the usual reduction from MAX-3SAT to IS (the one we use to prove NP-completeness of IS) is gap preserving.

Let φ be an instance of MAX-3SAT with n variables and m clauses. We construct the graph G from φ as follows. G has a vertex v_{ij} for every occurrence of variables x_i in clause C_j. All the vertices corresponding to literals from the same clause are joined by an edge. Also, if $x_i \in C_j$ and $\bar{x}_i \in C_{j'}$, we joint the vertices v_{ij} and $v_{ij'}$ by an edge. It is easy to see that an IS of size $\geq k$ in G iff there is an assignment which satisfies $\geq k$ of the clauses of φ. Thus we have:

- $OPT(\varphi) = m \Rightarrow OPT(G) \geq m$
- $OPT(\varphi) < sm \Rightarrow OPT(G) < sm$

where $s < 1$.

Thus we conclude:

Theorem 3 *IS is hard to approximate within s for $0 < s < 1$*

We can further amplify this constant by a reduction from an instance of IS with a gap of s/c to an instance of IS with a gap $(s/c)^2$. By repeating this reduction k times, we can obtain the following theorem:

Theorem 4 *IS is hard to approximate within a factor of s^k for every constant integer $k \geq 1$.*

Proof. Given an instance $G = (V, E)$, construct the graph $G' = (V', E')$ where $V' = V \times V$ and $E' = \{((u, v), (u', v'))|(u, u') \in E \text{ or } (v, v') \in E\}$. Let $I \subseteq V$ be an independent set of G. It is easy to see that by construction, the $I \times I$ is an IS of G'. Therefore, $OPT(G') \geq OPT(G)^2$. Now, let I' be an optimal IS of G' with vertices $(u_1, v_1), \ldots, (u_k, v_k)$. By construction, $u_1, \ldots, u_k$ and $v_1, \ldots, v_k$ are ISs in G, each contains at most $OPT(G)$ distinct vertices. Therefore, $OPT(G') \leq OPT(G)^2$. Thus we have $OPT(G') = OPT(G)^2$. Hence, we obtain the following:

- $OPG(G) \geq cn \Rightarrow OPT(G') \geq c^2n^2$
- $OPT(G) < sn \Rightarrow OPT(G') < s^2n^2$

□

Question: Can we show the hardness of approximation of IS using a direct reduction from PCP? (Similar to the idea of clique but adding edges in an opposite way?)

3.2 Vertex Cover

Prepared and presented by Nam Nguyen.

3.3 Hardness of Steiner Tree problem

Prepared and presented by Nam Nguyen.

3.4 MAX-3SAT with Bounded Occurrences

Definition 2 MAX-3SAT(k): *For each fixed k, define MAX-3SAT(k) to be restriction of MAX-3SAT in which each variable occurs at most k times.*

We are going to prove the following theorem:

Theorem 5 *There is a gap preserving reduction from MAX-E3SAT to MAX-3SAT(29) that transforms $\phi \in$ MAX-E3SAT to $\varphi \in$ MAX-3SAT(29) such that:*

- *If $OPT(\phi) = m \Rightarrow OPT(\varphi) = m'$*
- *If $OPT(\phi) < (1-\delta)m \Rightarrow OPT(\varphi) < (1-\eta)m'$*

where m and m' are the number of clauses in ϕ and φ respectively, $1 > \delta$ and $\eta > 0$ such that the GAP-MAX-E3SAT$_{1,\delta}$ is NP-hard, and $\eta = \delta/(1+3d)$ where $d = 14$

In other way, there exists a constant $0 < \eta < 1$ such that GAP-MAX-3SAT$(29)_{1,\eta}$ is NP-hard.

To prove it, we will need *Expander graphs* of the following type:

Definition 3 Expander Graph *Let c be a constant. An undirected graph $G = (V, E)$ is an $c-$edge expander if for every subset $S \subset V$, $|S| \leq |V|/2$, the number of edges in the cut $[S, V-S]$ is at least $c|S|$. (note that the cut $[S, V-S]$ is the set of all edges in G having one endpoint in S and another one in $V - S$.*

It is not very hard to find such a graph. For example, the complete graph has $c \geq \Omega(n)$. However, we are more interested in finding a sparse expanders, especially d-regular expanders for some constant d. How difficult is it? In fact, such sparse expanders can be constructed explicitly as follows:

Theorem 6 *There exist a constant $d = 14$ and an algorithm that runs in poly(n) time abd returns a regular graph of degree d with n vertices that is a 1-edge expander for all $n \geq 2$ [4]*

Note that if there is an expander for $d = d_0$ then there exists an expander for $d \geq d_0$. We can have the following more general theorem:

Theorem 7 *For every constant c, there is a constant $d = d(c)$ and an algorithm that runs in poly(n) time and returns a regular graph of degree d with n vertices that is a c-edge expander.*

We are now ready to prove Theorem 5 using the 1-edge expander as follows.

Proof. Let ϕ be an instance of GAP-MAX-E3SAT with n variables $x_1, \ldots, x_n$ and m clauses. Let f_i denote the number of occurrences of x_i in ϕ, that is, the number of clauses that involve the literal x_i or $\bar{x}_i$. Note that $\sum_i f_i = 3m$. We will construct an instance φ of GAP-MAX-3SAT$(2d+1)$ as follows:

φ will have $3m$ variables y_{ij} such that each variable y_{ij} corresponding to each variable $x_i \in C_j$. Next, we will construct the clauses of φ. φ will consists of two types of clauses:

- Primary clauses: For each clause $C_j \in \phi$, we create an equivalent clause in φ. That is, if $C_j = (x_a \vee x_b \vee x_c)$ is a clause in ϕ, then $(y_{aj} \vee y_{bj} \vee y_{cj})$ is a clause in φ
- Consistency clauses: Next, we need to find a relationship of a truth assignment of y_{ij} based on the assignment of x_i. For each variable x_i, we construct a d-regular 1-edge expander G_i with f_i vertices. Label each vertex in G_i as $[i,1], [i,2], \ldots, [i, f_i]$. For each edge $([i,j],[i,j'])$ in G_i, we add two clauses $(y_{ij} \vee \bar{y}_{ij'})$ and $(\bar{y}_{ij} \vee y_{ij'})$ to φ.

Question. Why we need to introduce the consistency clauses and what the use of expanders here? Note that if $y_{ij} = y_{ij'}$ then both consistency clauses are satisfied whereas if $y_{ij} \neq y_{ij'}$, then one of the two consistency clauses is not satisfied. Thus the truth assignment to y_{ij} is said to be consistent if either all the variables are set to true or all are set to false. An inconsistent truth assignment partitions the vertices of G_i into two sets, say S and $\bar{S}$ where $|S| \leq |\bar{S}|$. Then corresponding to each edge in the cut $[S, \bar{S}]$, φ will have an unsatisfied clause. Thus the number of unsatisfied clauses $\geq |S|$.

Also note that by construction, every variable occurs in at most $2d+1$ clauses of φ and φ has $m' = m + 3dm$ clauses. (**Show it!**) Thus φ is an instance of MAX-3SAT$(2d+1)$.

Now we are going to prove the two claims in Theorem 5. The first one is very easy. Take the assignment τ for ϕ and then for every variable y_{ij} of φ, assign to it the value that the assignment τ gives to x_i. This assignment will satisfy all clauses in φ. (since for all j, the assignments of y_{ij} are consistent, so all consistency clauses are satisfied. Plus, all m clauses in ϕ are satisfied by τ, then all primary clauses in φ must be satisfied.)

The proof is quite interesting for the second one, which we need to prove that if $OPT(\phi) < (1-\delta)m \Rightarrow OPT(\varphi) < (1-\eta)m'$. Assume that there exists a truth assignment τ of φ such that more than $(1-\eta)m' = (1-\eta)(3d+1)m$ clauses of φ satisfied. We will show that there will be more than $(1-\delta)m$ clauses of ϕ will be satisfied by some truth assignment, thus contradiction.

Consider such a truth assignment τ of φ, we will construct a new truth assignment τ' for φ as follows: For each i, assign all variables y_{ij} either True or False based on the majority assignment of y_{ij} in τ. That is, under an assignment τ, for each i, we have all variables y_{ij} partition into two sets, say S and $\bar{S}$. If $|S| \leq |\bar{S}|$, i.e. the majority assignment of y_{ij} is in $\bar{S}$, then we flip all the assignments of y_{ij} in S. (For example, for each i, we have three variables and their assignment: $y_{i1} = T, y_{i2} = F, y_{i3} = F$, then $S = \{y_{i1}\}$, and $\bar{S} = \{y_{i2}, y_{i3}\}$. Then we flip the assignment of y_{i1} from T to F. We are going to prove that the number of clauses in φ satisfied by τ' is at least that of by τ. After we flip at most $s = |S|$ assignments, then there at most s primary clauses that satisfied under τ is now not satisfied under τ'. That is, the number of satisfied primary clauses will be decreased by at most s. However, the number of satisfied consistency clauses will be increased by at least s according to the expanders as mentioned above. (Each edge in the cut is corresponding to 2 consistency clauses by our construction. Under τ, for each of these 2, only 1 is satisfied. However, after we flip the assignment in S, these two consistency clauses will be satisfied.) Therefore, the number of clauses satisfied by $\tau' \geq$ the number of clauses satisfied by τ.

From τ', we will construct an assignment τ'' for ϕ by letting $x_i = y_{ij}$ for each i. (Consider the above example, under τ', $y_{i1} = y_{i2} = y_{i3} = F$, then we have $x_i = F$.) Then under this truth assignment τ'', the number of satisfied clauses in ϕ is more than $(1-\eta)(3d+1)m - 3dm$. Thus, choosing $\eta = \delta/(1+3d)$, we have the contradiction. (That is, the number of satisfied clauses in ϕ is more than $(1-\delta)m$).

$\square$

In general, we have: For all constant $k \geq 29$, there exists a constant $0 < \eta < 1$ such that GAP-MAX-3SAT$(k)_{1,\eta}$ is NP-hard.

Note that Feige [3] has reduced k to $k \geq 5$.

3.5 Label Cover

Definition 4 Max-Label-CoverΣ *An instance of Label-CoverΣ is denoted by $\mathcal{L}(G = (U, V; E), \Sigma, \Pi)$ where $G = (U, V; E)$ is a regular bipartite graph, alphabet Σ, and $\Pi = \{\Pi_{uv} : \Sigma \to \Sigma \mid uv \in E\}$ is a set of all constraints in G. Each constraint Π_{uv} on each edge uv is a function from Σ to itself, also referred as the projection property. A labeling of the graph G is a mapping $L : U \cup V \to \Sigma$ which assigns a label for each vertex of G. A labeling L is said to satisfy an edge uv iff $\Pi_{uv}(L(u)) = L(v)$. A problem asks us to find L to maximize the number of satisfied edges.*

Theorem 8 *There exists a constant $0 < \eta < 1$ and Σ of constant size such that Gap-Max-Label-Cover$\Sigma_{1,\eta}$ is NP-hard.*

Proof To prove Theorem 8, we will show that there is a gap preserving reduction from GAP-MAX-E3SAT(Ed)$_{1,\delta}$ to Gap-Max-Label-Cover$\Sigma_{1,\eta}$ (such that it will transform an instance ϕ of MAX-E3SAT(Ed) to $\varphi \in$ MAX-Label-CoverΣ satisfying the following constraints: (1) If $OPT(\phi)m \Rightarrow OPT(\varphi) = m'$ and (2) If $OPT(\phi) < \delta m \Rightarrow OPT(\varphi) < \eta m'$ where m and m' are the number of clauses in ϕ and the number of edges φ respectively.)

Given an instance $\phi \in E3SAT$ with m clauses C_j and n variables x_i. Construct an instance $\varphi \in Label-Cover$ over Σ with $|\Sigma| = 7$ as follows: Let U and V be two sets of vertices in bipartite graph G where each vertex in U represents a variable x_i and each vertex in V represents a clause C_j. If $x_i \in C_j$, then there is a corresponding edges (x_i, C_j) in G. Note that $G = (U, V; E)$ is a regular bipartite graph (with left degree d and right degree 3) and $m' = 3m$.

We now define the constraint function Π. For each C_j, there are 7 assignments corresponding to seven symbols in Σ that satisfy C_j. Define $\Phi_{x_i,C_j} : \Sigma \to \{True, False\}$ where $True$ and $False$ correspond to two symbol in Σ (thus we still have $Pi_{x_i,C_j} : \Sigma \to \Sigma$) and the label on x_i is exactly the assignment of x_i implied by the label on C_j. The constraint for any edge (x_i, C_j) is satisfied if the labels on C_j and x_i are consistent. For example, let consider $C_j = (x_1 \vee x_2 \vee \bar{x}_3)$ and $\Sigma = \{000, 010, 011, 100, 101, 110, 111\}$. Let False $= 000$ and True $= 001$. Then we have $\Pi_{x_1,C_j}(000) = False = 000$, $\Pi_{x_1,C_j}(010) = False = 000$, $\Pi_{x_1,C_j}(011) = False = 000$, $\Pi_{x_1,C_j}(100) = True = 111$, $\Pi_{x_1,C_j}(101) = True = 111$, etc, $\Pi_{x_3,C_j}(000) = True = 111$

since $\bar{x}_3$ occurs in C_j, not x_3 itself.

Now, we prove that if $OPT(\phi) = m$, then $OPT(\varphi) = m'$. It is easy to see that if there is an assignment τ satisfying all m clauses in ϕ, then we just need label x_i as per the satisfying assignment, and giving C_j the label corresponding to the satisfying assignment, then this labeling will satisfy all edges, thus $OPT(\varphi) = m'$.

Now we prove that if $OPT(\phi) < \delta m \Rightarrow OPT(\varphi) < \eta m' = 3\eta m$. Consider any labeling L. Since the labeling of all nodes in U can be thought of as an assignment. Then by this assignment L, at most δm clauses are satisfied, that is, at least $(1-\delta)m$ clauses are unsatisfied. Consider one such unsatisfied clause C_j in V, one of its three neighbors nodes in U must be violating the constraint (Otherwise, C_j should be satisfied). It means that at least $(1-\delta)m$ edges are not satisfied, thus $OPT(\varphi) < 3m-(1-\delta)m = 3\eta m$ where $\eta = 1-(1-\delta)/3 = (2+\delta)/3$.

□

3.6 Set Cover

Prepared and Presented by Ying Xuan.

References

[1] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy, "Interactive Proofs and the Hardness of Approximating Cliques," *J. ACM*, 43(2):268-292, March 1996. (Preliminary version in 32nd FOCS, 1991)

[2] J. Hastad, "Clique is Hard to Approximate within $n^(1-\epsilon)$", *Acta Matehmatica*, 182:105-142, 1999

[3] U. Feige, "A Threshold of $\ln n$ for Approximating Set Cover", *J. of the ACM*, 45(4):634–652, 1998.

[4] S. Hoory, N. Linial, and A. Wigderson, "Expander Graphs and Their Applications," *Bulletin of the American Mathematical Society,* 43(4):439–561, 2006.