Algorithmic Lower Bound Techniques	**Winter 2015**

Lecture 1: October 29

Lecturer: Parinya Chalermsook *Scribe: Parinya Chalermsook*

Reduction is the main technique in proving conditional lower bounds (a.k.a. hardness results). Depending on the types of hardness results we are aiming at, there are certain parameters we need to pay attention to. In this lecture, we will start by discussing the *size* parameter, which plays a central role in proving *running time* lower bounds of *exact algorithms*. In the next lectures, we will discuss the most important player in *hardness of approximation*, the "gap" parameter.

1 Maximum Independent Set

Given graph G, we say that $J \subseteq V(G)$ is an independent set if there is no edge in J. In Maximum Independent Set (MIS), we are given graph $G = (V, E)$, and our goal is to find a maximum cardinality subset $J \subseteq V(G)$ such that J is an independent set. Let us consider a reduction from 3SAT to MIS.

We will show two reductions. The first reduction will imply the following result.

Theorem 1. *There is a polynomial-time reduction from m-clause SAT formula ϕ to a graph G of size $|V(G)| = O(n)$ such that ϕ is satisfiable if and only if $\alpha(G) \geq m$.*

Corollary 1. *3SAT has a polynomial time algorithm if MIS admits a poly-time algorithm. In particular, assuming $P \neq NP$, MIS does not have poly-time algorithm.*

Notice that, in order to derive Corollary 1, any reduction that blows up the size by a polynomial factor would do the job. What if we blow up the instance super polynomially? For instance, if we have a reduction from 3SAT to problem Π with size $n^{\log n}$, what does it mean? Well, the polynomial time algorithm that solves Π would imply an $n^{O(\log n)}$ algorithm that solves SAT, so this means that unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{poly \log n})$, there is no polynomial time algorithm for Π. In other words, we use a *stronger assumption* to rule out polynomial-time algorithms for Π.

Let's assume the strongest possible assumption that 3SAT does not have $2^{o(n)}$ time algorithm. Then, this will imply the same for MIS. This assumption is referred to as *Exponential Time Hypothesis (ETH)* (Impagliazzo and Paturi).

1.1 Reduction 1

Given a 3SAT formula $\phi = \bigwedge_{i=1}^{m} C_i$ where C_i is a disjunctive of at most 3 variables. For each C_i and satisfying assignment A, we create a vertex $v(i, A)$; choosing this vertex in the independent set is supposed to encode "satisfy the clause C_i by A". Now the edges: For each pair $v(i, A)v(i, A')$ such that $A \neq A'$, we connect them by an edge. Also, we connect each pair $v(i, A)v(i', A')$ such that $i \neq i'$, and C_i shares some variable x_j with $C_{i'}$, but the assignments A and A' give x_j the opposite values. Let G_ϕ denote the resulting graph.

Theorem 2. *The maximum number of clauses satisfiable in ϕ is exactly $\alpha(G_\phi)$.*

Proof. We need to prove two directions. First, assuming that there is an assignment $\sigma : \{x_1, \ldots, x_n\} \to \{true, false\}$, that satisfies m' clauses. We construct an independent set J by choosing, for each i such that C_i is satisfied, the vertex $v(i, A_i)$ where A_i is the "projection" of the assignment σ on those variables appearing in C_i. The size $|J|$ is exactly the number of satisfied clauses. It is easy to check that J is independent.

Now, consider any independent set $J \subseteq V(G_\phi)$. We will show that there is an assignment that satisfies at least $|J|$ clauses. $\square$

1.2 Reduction 2: Size and extra parameter k

In the previous reduction, we see that MIS in general requires exponential time (assuming ETH). What if I know that I only want to find an independent set of size k? How hard would it be? Simply doing bruteforce already gives $\binom{n}{k} \leq n^k$ running time. Can we do it faster? We will show below that, assuming ETH, this is likely the best you can do.

Theorem 3. *Let k be any integer. There is a reduction that takes formula ϕ and produces a graph G_k^ϕ such that*

- $|V(G_k^\phi)| = k2^{O(m/k)}$.
- $\alpha(G_k) \geq k$ *if and only if ϕ is satisfiable.*

Proof. We group the clauses $C_1, \ldots, C_m$ into k groups, each having m/k clauses (to get rid of stupid technicalities, let's assume that m/k is integer.) Let us refer to each group $\bigwedge_{i=(\alpha-1)(m/k)+1}^{\alpha m/k} C_i$ as a super clause $\mathcal{C}_\alpha$. For each super-clause $\mathcal{C}_\alpha$, for each assignment A that satisfies this super-clause, we have a vertex $v(\alpha, A)$. The rest of the reduction closely follows the previous section. $\square$

From this, it is not hard to argue the following: Assuming that we have $|V(G)|^{o(k)}$-time algorithm for MIS, this would imply $2^{o(m)}$-time for solving 3SAT.