Connections between exponential time and polynomial time problem
Lecture Notes for CS294

Lecturer: Russell Impagliazzo
Scribe: Stefan Schneider

November 10, 2015

We connect problems that have exponential time-complexity with much easier polynomial time problems using reductions that preserve their *fine-grained complexity*.

1 Subset-Sum and k-Sum

Consider the Subset-Sum problem we have seen before.

Definition 1. *The* Subset-Sum *problem is given integers $a_1, \dots, a_n \in [-2^l, 2^l]$ and $T \in \mathbb{N}$, find a set $S \subseteq \{1, \dots, n\}$ such that*

$$\sum_{i \in S} a_i = T \tag{1}$$

The trivial algorithm for subset sum tests all subsets of numbers and takes time $O(2^n)$, but we have seen in earlier lectures that we can improve this algorithm to $O(2^{n/2})$. In fact, we showed that Subset-Sum is in time $\min\{2^{n/2}, 2^l\}$.

The k-Sum problem is in some sense the polynomial time equivalent of subset sum.

Definition 2. *The k-*Sum *problem is given k sets of integers $A_1, \dots, A_k$ with $|A_1| = \dots = |A_k| = n$ and $T \in \mathbb{N}$, find $a_i \in A_i$ for all i such that*

$$\sum_{i=1}^{k} a_i = T \tag{2}$$

Using a similar idea as used for Subset-Sum, we can solve k-Sum in time $O(n^{\lceil k/2 \rceil})$. The k-Sum conjecture then says that this runtime is essentially optimal.

Conjecture 1 (k-Sum conjecture)**.** *k-Sum requires time $O(n^{\lceil k/2 \rceil - \varepsilon})$ for all $\varepsilon > 0$.*

As we will see next lecture, the k-Sum conjecture implies interesting lower bounds for geometric problems.

We have previously seen that Subset-Sum is hard assuming the Exponential Time Hypothesis (ETH). In particular, we have seen a reduction from 3-SAT on n variables and m clauses to Subset-Sum of size $n' = O(n + m)$ and $l = O(n + m)$. Hence, if ETH is true, there is a $c > 0$ such that Subset-Sum is not in time $\min\{2^{cn}, 2^{cl}\}$.

We want to prove a similar statement for k-Sum. Let $c_k = \inf_c k\text{-Sum} \in \text{TIME}(n^c)$ be the best exponent in the runtime for k-Sum. Note that we take an infimum to avoid problems if there is an infinite sequence of better and better algorithms rather than a single optimal algorithm.

Theorem 1 ([1]). *If ETH is true, then $c_k = \Theta(k)$.*

Proof. The idea is to reduce Subset-Sum to k-Sum. Let $A = \{a_1, \ldots, a_n\}, T$ be the input to the Subset-Sum problem. Divide A into k sets $\{(i-1)\frac{n}{k}+1, \ldots, i\frac{n}{k}\}$ of size $\frac{n}{k}$ each and let A_i be all sums of subsets of this sets. Note that $N = |A_i| = 2^{n/k}$. We then ask if the k-Sum instance $A_1, \ldots, A_k, T$ has a solution.

Using the fact that we have an $O(N^{c_k+\delta})$ algorithm for k-Sum for any $\delta > 0$ we get a total time complexity of

$$O\left(k2^{n/k}\right) + O\left(N^{c_k+\delta}\right) = O\left(2^{n\frac{c_k+\delta}{k}}\right)$$

Assuming ETH the time complexity of Subset-Sum is lower bounded by 2^{cn} for some constant c. Therefore we have $\frac{c_k}{k} \geq c$ and therefore $c_k = \Omega(k)$. □

2 Independent Set and k-Independent Set

In this section we consider the k-independent set problem we have seen in earlier lectures

Definition 3. *The* k-independent set *problem is given a graph G, decide whether there is a set of k vertices such that they are independent, i.e. there is no edge between them.*

Note the the independent set problem on the complement graph is the k-clique problem. For the purpose of this section there is no difference between the k-independent set problem and the k-clique problem.

As seen earlier, this problem can be solved in time $n^{\omega k/3}$ where ω is the matrix multiplication exponent by reduction to the problem of finding a triangle in a graph. Triangle detection of a graph G with adjacency matrix M can be solved in time $O(n^\omega)$ by checking if M^3 contains a positive entry in the diagonal.

We have also seen that the maximum independent set problem requires time 2^{cn} for some constant c assuming ETH.

Definition 4. *The* maximum independent set *problem is given a graph G and $t \in \mathbb{N}$, decide if there is an independent set of size at least t.*

Let c_k be the best exponent for k-independent set. We want to show that $c_k = \Theta(k)$ with a reduction from independent set to k-independent set.

Consider an input $G = (V, E), t$ for the independent set problem. Partition V into k sets $V_1, \ldots, V_k$ with $|V_i| = \frac{n}{k}$ arbitrarily.

Consider a tuple $t_1, \ldots, t_k$ with $t_1 + \cdots + t_k = t$. Note that there are $O(n^k)$ such tuples. We build a graph G' consisiting of k cliques $U_1, \ldots, U_k$ where the vertex set of U_i consists of all independents sets of V_i of size exactly t_i. For two vertices $S \in U_i$ and $T \in U_j$ we further add an edge (S, T) if $S \cup T$ is *not* and independent set.

In G' any k-independent set must be exactly one vertex per clique. Furthermore, since a k-independent set cannot contain any edges, such a set must necessarily correspond to an independent set of size t in the original graph G.

To execute this reduction and then solve the k-independent set problem we need to construct this graph and solf the k-independent set problem n^k times. The time complexity is therefore given by

$$n^k\left(k2^{2n/k} + \left(2^{n/k}\right)^{c_k}\right)$$

Assuming ETH this has to be lower bounded by 2^{cn} for some constant c. We can conclude $c \leq \max\{\frac{2}{k}, \frac{c_k}{k}\}$ and therefore $c_k = \Theta(k)$.

3 Tight Lower Bounds for Orthogonal Vectors

So far we used ETH to argue about the asymptotic growth of the the exponent. To get a result on a more specific exponent we need to use a strong hypothesis.

Definition 5. *The* Strong Exponential Time Hypothesis *(SETH) is that for all $\varepsilon > 0$ there is a k such that k-SAT requires time $\Omega(2^{n-\varepsilon n})$.*

We want to apply SETH to the 2-Orthogonal Vectors problem.

Definition 6. *The* 2-Orthogonal Vectors *problem is given two sets of boolean vectors $\mathcal{S}$ and $\mathcal{T}$ with $|\mathcal{S}| = |\mathcal{T}| = n$ and every $A \in \mathcal{S}$ and $B \in \mathcal{T}$ are d-dimensional boolean vectors, i.e. $A = A_1 A_2 \dots A_d$ with $A_i \in \{0,1\}$ and $B = B_1 \dots B_d$ with $B_i \in \{0,1\}$.*

The question is if there is $A \in \mathcal{S}$ and $B \in \mathcal{T}$ such that A and B are orthogonal, i.e.

$$A_1 B_1 + \cdots + A_d B_d = 0$$

We will consider the problem where $d = \mathsf{polylog} n$. The obvious algorithm for 2-orthogonal vectors tests all pairs of vectors and runs in time $O(n^2 d)$.

The *Orthogonal Vectors Conjecture* (OVC) is that this is essentially optimal. Specifically, if $d = \omega(\log n)$, then there is no $O(n^{2-\varepsilon})$ algorithm for any $\varepsilon > 0$.

Theorem 2 ([2])**.** *SETH implies OVC*

Proof. We use the split and list technique which we already used in the previous examples. Consider a k-CNF with variable set $x_1, \dots, x_n$ and let $C_1, \dots, C_m$ be its clauses. By the Sparsification Lemma we can assume $m = O(n)$.

We will construct two sets of d-dimensional boolean vectors $\mathcal{S}$ and $\mathcal{T}$ with $N = |\mathcal{S}| = |\mathcal{T}| = 2^{n/2}$ and $d = m = cn = c' \log |\mathcal{S}|$.

Split the variables into two sets and let α be an assigment to the first set of variables. Define the vector A_α as

$$(A_\alpha)_j = \begin{cases} 1 & \text{if } \alpha \text{ does not satisfy } C_i \\ 0 & \text{otherwise} \end{cases}$$

Symmetrically, for β an assignment to the second set of variables define B_β as

$$(B_\beta)_j = \begin{cases} 1 & \text{if } \beta \text{ does not satisfy } C_i \\ 0 & \text{otherwise} \end{cases}$$

We define $\mathcal{S}$ as the set of all A_α obtained that way and $\mathcal{T}$ as the set of all B_β.

We have that α, β satisfies the formula if and only if for all j either α or β satisfies C_j. Hence either $(A_\alpha)_j = 0$ or $(B_\beta)_j = 0$, which is the case exactly if A_α and B_β are orthogonal.

If we now have an algorithm for orthogonal vectors that runs in time $O(n^{2-\varepsilon})$ for some $\varepsilon > 0$ if $d = \omega(\log N)$ tk hen we have such an algorithm for $d = C \log N$ for all constants C. Hence for any k we get a total time to solve k-SAT of

$$O\left(2^{n/2} + \left(2^{n/2}\right)^{2-\varepsilon}\right) = O\left(2^{n - n\varepsilon/2}\right)$$

which contradicts the Strong Exponential Time Hypothesis. □

4 Graph Diameter

In this section we show that it is hard under SETH to approximate the diameter of a graph within a factor of 3/2.

Definition 7. *For an unweighted, undirected graph G, the diameter of G is the maximum (shortest path) distance between any two vertices.*

The obvious algorithm for graph diameter does a breadth first search from every starting point. The total time complexity of this algorithm is $O(nm)$.

We reduce the orthogonal vectors problem to the graph diameter problem.

Given an instance of the orthogonal vectors problem, we construct a graph $G = (V, E)$ with

$$V = \mathcal{S} \cup \mathcal{T} \cup \{1, \ldots, d\} \cup \{s, t\}$$

We have to following edges:

- (s, A) for all $A \in \mathcal{S}$
- (s, i) for all $i \in \{1, \ldots, d\}$
- (t, B) for all $B \in \mathcal{T}$
- (t, i) for all $i \in \{1, \ldots, d\}$
- (s, t)
- (A, i) if $(A)_i = 1$
- (B, i) if $(B)_i = 1$

We can observe the following distances:

- $\text{dist}(A_1, A_2) = 2$
- $\text{dist}(B_1, B_2) = 2$
- $\text{dist}(A, i) \leq 2$
- $\text{dist}(B, i) \leq 2$
- $\text{dist}(A, s) = 1$
- $\text{dist}(A, t) = 2$
- $\text{dist}(B, s) = 2$
- $\text{dist}(B, t) = 1$
- $\text{dist}(i, j) = 2$

Furthermore we have

$$\text{dist}(A, B) = \begin{cases} 2 & \text{if they are not orthogonal} \\ 3 & \text{if they are orthogonal} \end{cases}$$

Hence we have a diameter of 3 if there is an orthogonal pair and a diameter of 2 otherwise. Using the orthogonal vectors conjecture (or SETH) we can therefore conclude that distinguishing between graphs of diameters 2 and 3 requires time $\Omega(n^{2-\varepsilon})$ for all ϵ. In particular this also implies that approximating the diameter within a factor of 3/2 also requires quadratic time.

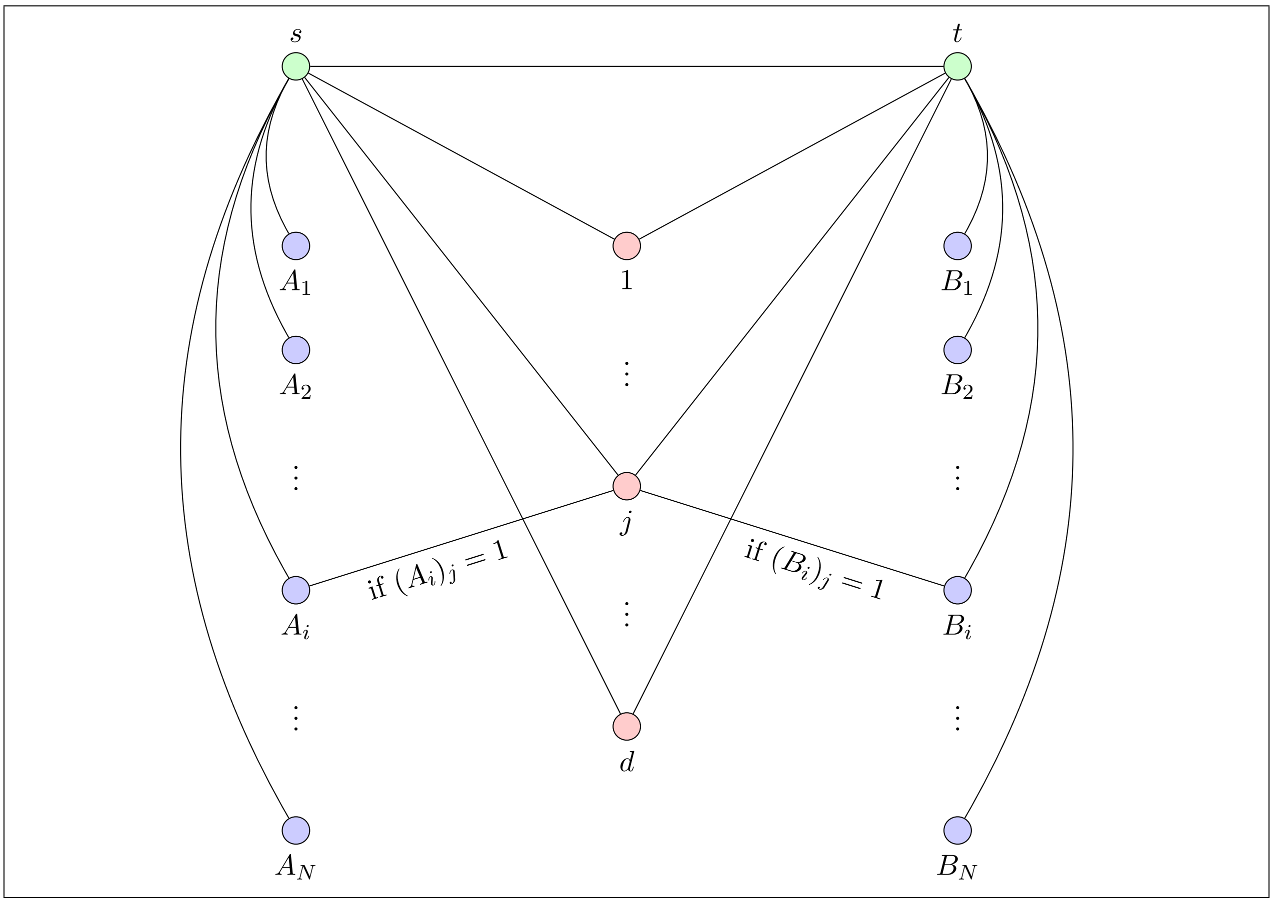

Figure 1: The reduction from orthogonal vectors to graph diameter

References

[1] Mihai Patrascu and Ryan Williams. On the possibility of faster sat algorithms. In *SODA*, volume 10, pages 1065–1075. SIAM, 2010.

[2] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2):357–365, 2005.