1 SAT algorithms, ETH and SETH

The brute-force algorithm – checking all possible assignments – solves CNF-SAT in $\mathcal{O}(2^n \cdot \text{poly}(n, m))) = \mathcal{O}^*(2^n)$ time, where n and m denote the numbers of variables and clauses, respectively. Specifically for 3-SAT, there is a long list of improved algorithm. The current best (Hertli, FOCS'11) runs in $\mathcal{O}(1.3071^n) = \mathcal{O}(2^{0.3863n})$ time. For $k > 3$, we also know better-than-brute-force algorithms for k-SAT, but the base of the exponential function grows with k. In particular, none of these algorithms is faster than $\mathcal{O}(2^{1-c/n})$ for a constant c. This means that, with k growing to infinity, their running times converge to the naive bound $\mathcal{O}^*(2^n)$. This motivates the following definition and hypotheses.

Definition 1. $s_k = \inf\{\delta \mid k\text{-SAT can be solved in } \mathcal{O}(2^{\delta n}) \text{ time}\}$.

Exercise 1. Show that $s_2 = 0$.

Hypothesis 1 (Exponential Time Hypothesis (ETH))**.** $s_3 > 0$.

Hypothesis 2 (Strong Exponential Time Hypothesis (SETH))**.** $\lim_{k\to\infty} s_k = 1$.

As we will see later in the course, SETH implies ETH, hence the name is justified.

2 Sparsification Lemma

Let us focus on our favorite NP-complete problem – Vertex Cover – and recall its NP-hardness proof.

Exercise 2. Show a reduction from 3-SAT to Vertex Cover that produces instances with $2n + 3m$ vertices, $n + 6m$ edges, and $k = n + 2m$.

A 3-SAT instance can have at most $m = O(n^3)$ clauses. An immediate consequence of the above reduction is that, assuming ETH, there is no $2^{o(|V|^{1/3})}$ nor $\mathcal{O}^*(2^{o(k^{1/3})})$ time algorithm for Vertex Cover. Given that the best FPT algorithms for Vertex Cover run in time $\mathcal{O}^*(2^{ck})$, for a constant c, this is not a tight lower bound. Now we will learn a tool that will let us close that gap.

Theorem 1 (Sparsification Lemma (Impagliazo, Paturi, Zane))**.** *For $\epsilon > 0$, given a k-CNF formula ψ on n variables, one can compute $2^{\epsilon n}$ k-CNF formulas on n-variables $\psi_1, \ldots, \psi_{2^{\epsilon n}}$ such that: (1) ψ is satisfiable if and only if at least one of ψ_i's is satisfiable (in other words, $\psi \equiv \psi_1 \vee \cdots \vee \psi_{2^{\epsilon n}}$), and (2) each ψ_i has at most $c_{k,\epsilon} n$ clauses, where $c_{k,\epsilon}$ is a constant that depends only on k and ϵ. The formulas cane be found in $\mathcal{O}^*(2^{\epsilon n})$ time.*

Proof sketch. Think of a bounded-search-tree algorithm for SAT. As long as the formula is not sparse enough, we can find a *small* subset of literals that appears in *many* clauses. We branch on whether at least one of these literals is satisfied. In the yes-branch, we remove all the clauses containing the subset, and add a single clause composed of only that subset. In the no-branch, we remove the subset from each of the containing clauses. It remains to carefully choose the parameters for *small* and for *many*, and analyse the depth of the tree, which is highly nontrivial. An accessible analysis is available in Mohan Paturi's slides from ADFOCS'18: `https://conferences.mpi-inf.mpg.de/adfocs-18/mohan/2018_adfocs_2.pdf#page=26`. □

Now we can easily prove that an $\mathcal{O}^*(2^{o(k)})$ time algorithm for Vertex Cover would break ETH. Indeed, for any $\epsilon > 0$, we could first run the Sparsification Lemma, then apply to each formula the 3-SAT-to-VC reduction, and finally run the hypothesized VC algorithm. That would give an $\mathcal{O}^*(2^{\epsilon n+o(n)})$ time algorithm for 3-SAT, for any $\epsilon > 0$, implying that $s_3 = 0$.

Similar arguments apply to many classical NP-complete problems – e.g., Feedback Vertex Set, 3-Coloring, Dominating Set – ruling out $\mathcal{O}^*(2^{o(k)})$ time algorithms, unless ETH fails. Moreover, since 3-SAT can be reduced to Planar-3-SAT with only a quadratic increase in the formula size, most of these problems restricted to planar graphs cannot have $\mathcal{O}^*(2^{o(\sqrt{k})})$ time algorithms (assuming ETH). That matches the upper bounds following from the grid theorem and dynamic programming on tree decomposition.

ETH can be used to lower bound asymptotics of a function in running time's exponent. In order to bound a specific constant, we would need SETH. Unfortunately, so far we do not know how to prove SETH-based lower bounds against FPT algorithms parameterized by solution size (can you see why? try to go through the VC example). However, SETH is very useful for proving tight bounds for problems parameterized by treewidth. E.g., under SETH, there is no $\mathcal{O}^*((2-\epsilon)^2)$ time algorithm for Vertex Cover and no $\mathcal{O}^*((3-\epsilon)^2)$ time algorithm for Dominating Set on graphs of treewidth (or even pathwidth) bounded by w. If you are interested, see Chapter 14.5.2 in the FPT book (pages 508–514).

3 Orthogonal Vectors

Most SETH-based lower bounds for polynomial time problems go through an intermediate problem called Orthogonal Vectors.

Definition 2 (Orthogonal Vectors (OV) problem)**.** Given two sets of d-dimensional 0-1 vectors $\mathcal{U}, \mathcal{V} \subseteq \{0,1\}^d$, both of the same size $|\mathcal{U}| = |\mathcal{V}| = n$, decide if there exists a pair of vectors $u \in \mathcal{U}$ and $v \in \mathcal{V}$ that are orthogonal, i.e., their inner product $u \cdot v := \sum_{i=1}^{d}(u_i \cdot v_i)$ equals 0.

The OV problem is equally often defined with one set of vectors instead of two. As we will see later, the two problems have the same running time, up to constant factors.

The following hypothesis is implied by SETH.

Hypothesis 3 (OV Hypothesis)**.** *There is no $\mathcal{O}(n^{2-\epsilon}\mathrm{poly}(d))$ time algorithm for OV, for any constant $\epsilon > 0$.*

Theorem 2 (Williams (ICALP'04))**.** *SETH implies OV Hypothesis.*

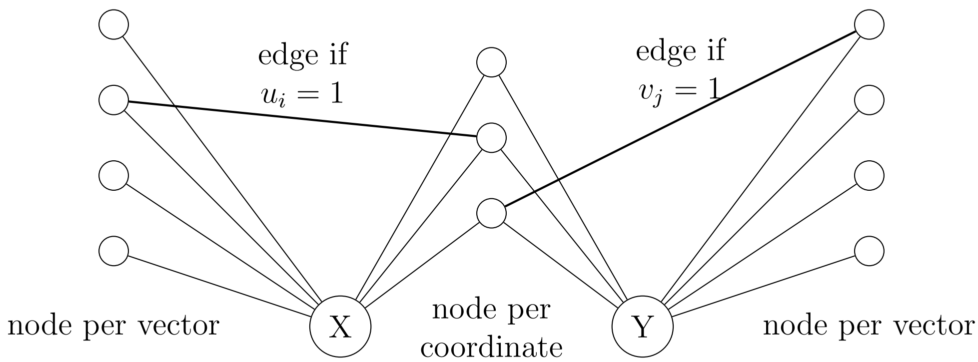
Proof. Take any CNF-SAT instance with n variables and m clauses, and split the variables into two sets, each of size $n/2$. Consider every possible assignment to the variables from the first set, and for each such assignment create an m-dimensional $(0,1)$-vector whose i-th coordinate equals 0 if the assignment satisfies the i-th clause. Let $\mathcal{U}$ denote the just constructed set of $2^{n/2}$ vectors. Repeat the same procedure for the second half of variables to obtain set $\mathcal{V}$. Now, observe that a satisfying assignment to all variables can be composed out of two partial assignments, one to the first half and one to the second half, such that any clause is satisfied by at least one of them. This corresponds to the condition that for each $i \in [m]$ at least one of the two vectors, representing the two partial assignments, has 0 as its i-th coordinate. That is, an assignment is satisfying if and only if the two vectors are orthogonal. Therefore, if there is an $O(n^{2-\epsilon}\text{poly}(d))$ time algorithm for OV, applying it to the sets $\mathcal{U}, \mathcal{V}$ yields an $O(2^{(n/2)\cdot(2-\epsilon)}\text{poly}(m))$ algorithm for CNF-SAT, which refutes SETH. □

Note that lower bounds proved by a reduction from OV should be more believable than those that follow directly from SETH – it is possible that SETH is false while OV Hypothesis remains true.

Now we will see our first lower bound for a polynomial time problem. Recall the diameter approximation problem from the last problem set.

Theorem 3 (Roditty, Vassilevska Williams (STOC'13)). *Assuming OV Hypothesis, diameter cannot be approximated within a factor of* $(3/2)-\epsilon$ *in* $\mathcal{O}(n^{2-\epsilon})$ *time, for any* $\epsilon > 0$*, even in graphs with* $m = n^{1+o(1)}$ *edges.*

Proof. (From Virginia Vassilevska Williams's slides[1]) Consider the following graph:



Observe that:

- any two vector nodes from the same side are at distance 2;
- any coordinate node is at distance 2 from everyone;
- X and Y are at distance 2 from everyone;
- two vector nodes u and v from different sides are at distance
  - 2 if there exists i with $u_i = v_i = 1$,
  - 3 otherwise.

Hence, the diameter is 3 if there is exists an orthogonal pair, and 2 otherwise.

With Sparsification Lemma we can assume $d = \mathcal{O}(\log n)$. The graph has $\mathcal{O}(n)$ nodes and $m = \mathcal{O}(n \log n)$ edges. An $\mathcal{O}(m^{2-\varepsilon})$ time algorithm distinguishing between diameter 2 and 3 would falsify OV Hypothesis. □

The same paper[2] provides a matching upper bound: a (3/2)-approximation algorithm running in time $\widetilde{\mathcal{O}}(m\sqrt{n})$. (The $\widetilde{\mathcal{O}}$ notation is a polynomial-time analogue of $\mathcal{O}^*$, i.e., $\widetilde{\mathcal{O}}(f(n)) = \mathcal{O}(f(n) \cdot \text{polylog}(n))$.)

[1] `https://people.csail.mit.edu/virgi/Graph%20problems.pdf#page=27`
[2] `https://people.csail.mit.edu/virgi/diam.pdf`