Fine-Grained Hardness

Styopa Zharkov

1 Introduction

From our very first complexity theory course, we are trained to say that polynomial time algorithms are fast and superpolynomial time algorithms are slow. This is nice because polynomials are a very rigid class—they're closed under addition, multiplication, and composition. This let's us combine poly-time algorithms in various ways and easily keep the result in poly-time as well. Outside of the complexity theorist's pink cloud of simplicity and ease, however, $O(n^{100})$ algorithms look a little more grim. Even quadratic time algorithms are frowned upon by most data workers and programmers.

In this introduction, let's take a step towards reality and examine a poly-time problem with a finer microscope. Unconditionally proving lower bounds on algorithms turns out to be incredibly difficult. In fact, there are no known techniques beyond the standard $O(n \log n)$ comparison bounds. Instead, we will look at a technique to prove stronger bounds conditionally. We'll assume a statement that is likely to be true, and we'll use it to show the lower bound that we want.

The assumption we make here is the Strong Exponential Time Hypothesis, or SETH for short. This is just a stronger version of P $\neq$ NP. Let's examine it more closely.

2 Who Are You, SETH?

Before we approach SETH himself, let's recall his little brother, ETH.

Definition 1: Let s_k be the infimum of all real numbers δ such that $k-$SAT is solvable in $O(2^{\delta n})$ time. The *Exponential Time Hypothesis* (ETH) is the statement that there exists a k such that $s_k > 0$. ◁

We saw this earlier as the statement that $k-$SAT requires exponential time. The s_k parameter here is intuitively "how brute force" the algorithm to solve $k-$SAT must be, with 0 meaning $k-SAT$ is subexponential and 1 meaning that brute force is the best algorithm there is.

Now that we got a nibble of the idea, let's bite into the definition we are after.

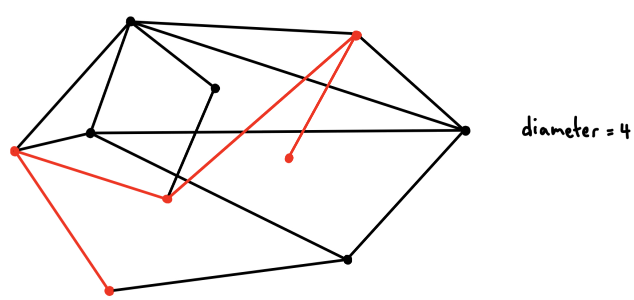
Definition 2: Let s_k be as in the previous definition. The *Strong Exponential Time Hypothesis* (SETH) is the statement that $\lim_{k\to\infty} s_k = 1$. We assume that the number of clauses is polynomial in the number of variables. ◁

Intuitively, this means that brute force is essentially as good as SAT-solving algorithms get. It's easy to see that SETH implies ETH which implies P $\neq$ NP. Since we are proving that SETH implies a lower bound, disproving the lower bound would mean disproving SETH, which would be nice. The truthfulness of SETH is debated, but it seems that most complexity theorists believe it is true.

3 Diameter of a Graph

Now, let's look at the problem that we'll be proving a lower bound for.

Definition 3: The *diameter* of a graph G is the maximum shortest path between two vertices in the graph. If G isn't connected, we say the diameter is infinity. The problem DIAMETER is the problem of finding the diameter of a graph. ◁



More specifically, we will work with trying to determine if a graph has a diameter of 2 or 3.

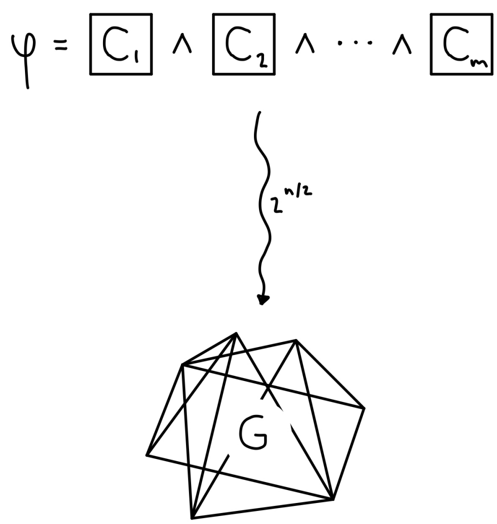
4 Hardness of Calculating the Diameter

Let's dive into the main theorem itself.

Theorem 1: SETH implies that for any $\varepsilon > 0$, computing the diameter of an N-vertex, M-edge graph takes $\Omega(MN^{1-\varepsilon})$ time [RW12]. $\triangleleft$

Our strategy for proving this will be to create a reduction from SAT to DIAMETER. The reduction will take exponential time, and the resulting graph will have exponential size. A fast algorithm to find the graph diameter would mean a faster-than-brute-force algorithm for solving SAT. This idea of an exponential time reduction is quite different from the basic poly-time reductions that we have been making thus far.

More precisely, our reduction will take in a Boolean formula $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ and output a a graph G such that the diameter of G is 3 if $\varphi \in$ SAT and 2 if $\varphi \notin$ SAT. Also, G will have $N = O(2^{n/2})$ vertices and $M = O(2^{n/2})$ edges. The reduction will take $O(2^{n/2})$ time.



Suppose that we can actually build this reduction. Then, if there exists an $\varepsilon > 0$ such that computing the diameter takes $o(MN^{1-\varepsilon})$ time, then we can solve SAT in $o(2^{n/2} + 2^{n/2} \cdot 2^{n/2-\varepsilon n/2}) = o(2^{n(1-\varepsilon/2)})$, which falsifies SETH with $\delta = \varepsilon/2$. The contrapositive of the previous statement is exactly the theorem we want to prove. So, all we need to do is create the reduction and show that it works.

Interestingly, in 1999, Aingworth, Chekuri, Indyk, and Motwani presented a randomized algorithm [ACM96]that finds the diameter of a graph with a 2/3 accuracy[1]. Their algorithm runs in $O(MN^{1/2})$ time. This is just barely not strong enough to tell the difference between a diameter 2 and diameter 3 graph. If their algorithm were any more accurate than 2/3, then we would be able to disprove SETH[2]. So, in a sense, this lower bound is tight.

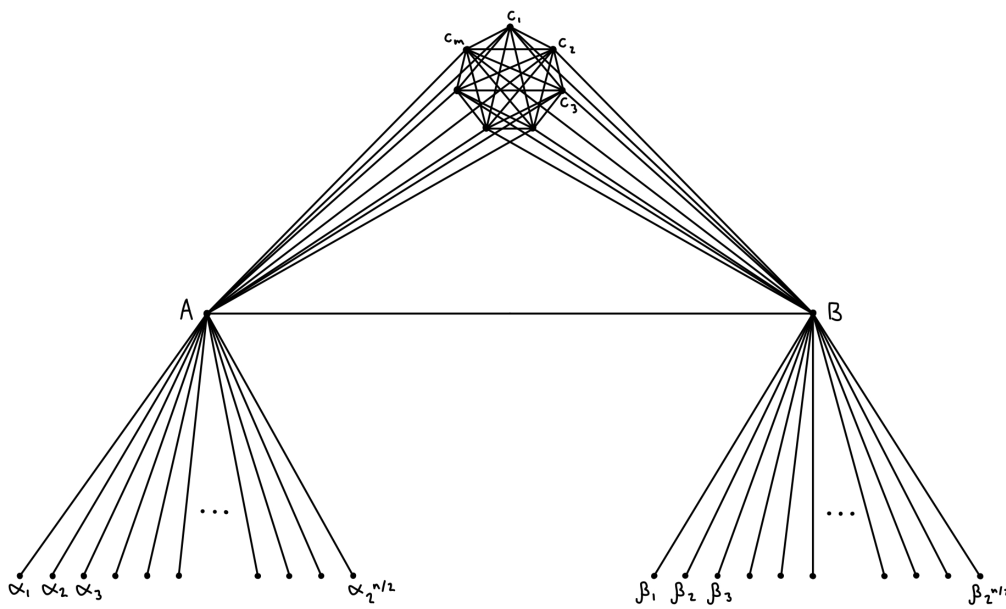
5 Construction of the Reduction

On input formula $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ with n variables, construct the graph G as follows:

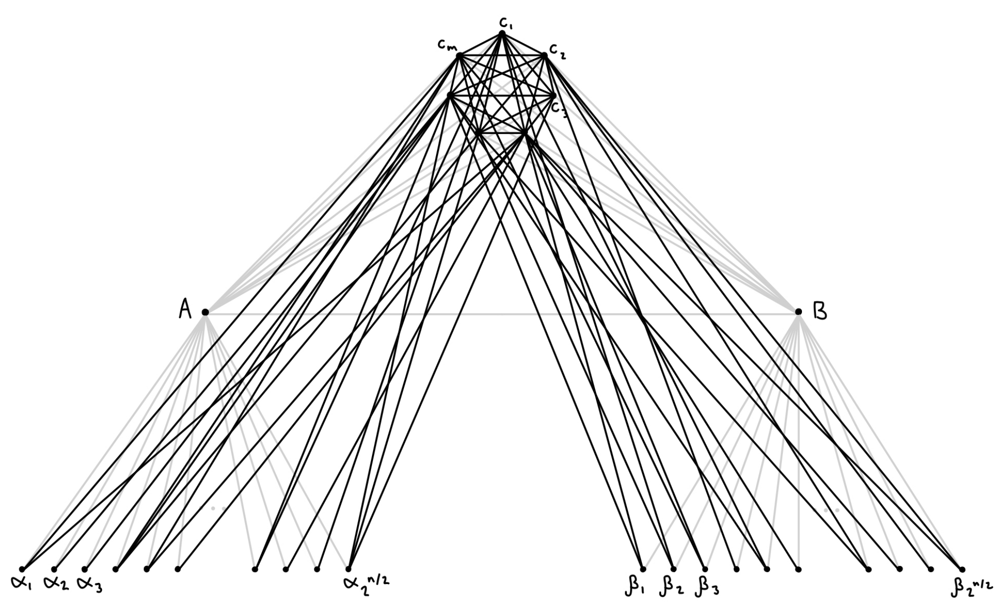
1. For each clause C_i, create a vertex c_i.
2. Create two vertices A and B.
3. Connect A to B, connect every pair of vertices c_i, c_j and connect both A and B to all vertices c_i (i.e. create a clique on all of the vertices we have so far).

[1] This means that the true diameter is at most a 2/3 factor away from the output.
[2] Although the algorithm is randomized, disproving SETH with a randomized algorithm would also be an incredible breakthrough.

4. Enumerate all possible assignments to the first $n/2$ of the n variables in φ. There are $2^{n/2}$ such assignments. Create $2^{n/2}$ vertices $\alpha_1, \ldots, \alpha_{2^{n/2}}$, one for each assignment.
5. Connect A to all α_i.
6. Create vertices $\beta_1, \ldots, \beta_{2^{n/2}}$, one for each assignment of the second half of the variables of φ.
7. Connect B to all β_i. The graph should currently look something like this.



8. Now, let's add the most interesting part. We'll connect some α_i's and some β_i's to the c_i's. Namely, let's connect α_i to c_j if the assignment of the first $n/2$ variables corresponding to α_i does not satisfy C_j. Similarly connect β_i to c_j if the assignment of the second $n/2$ variables does not satisfy C_j. Note that this is fast to check. Now, our finished graph G looks something like this:



With this, we have completed the construction.

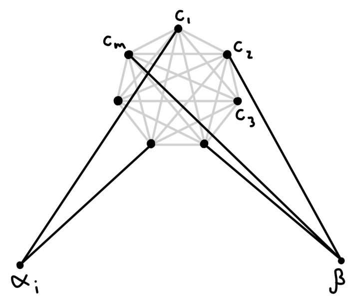
6 Proof of the Theorem

Let's analyze the construction and convince ourselves that it works.

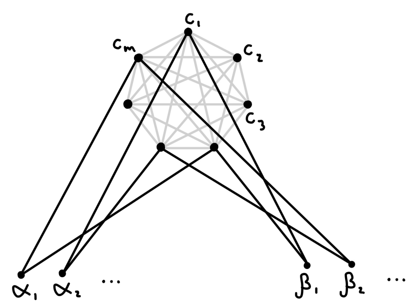
Proof of Theorem 1: First, we can see that building all of the vertices and connecting them takes $O(2^{n/2})$ time, so our construction is good on time.

Now let's analyze the correctness. We see that before we added the edges in step 8, the diameter of the graph is 3 because that's the distance from any α vertex to any β vertex. The diameter only becomes 2 if we connect every α to every β with a path of length 2 through some c_i.

Looking more closely, if $\varphi \in \text{SAT}$, then there exists some satisfying assignment. We can look at the vertices that correspond to the two halves of this assignment α_i and β_j. Every clause is satisfied either by a variable in the first half or a variable in the second half, so no c_i can be connected to both α_i and β_j. So, the distance from α_i to β_j is 3. Thus, the diameter of G is 3.



If $\varphi \notin \text{SAT}$, then every assignment isn't satisfying. So, for any α_i and β_j, there exists some clause C_ℓ that isn't satisfied. So, both c_ℓ is connected to both α_i and β_j. This means the distance between all pairs of an α and a β is 2. So, the diameter of G is 2.



So, our reduction works! By the discussion that we had in section 1.3, this means that SETH implies that computing the diameter of G takes $MN^{1-\varepsilon}$ time for any $\varepsilon > 0$, which is what we wanted to show. $\square$

To review, we created a reduction that showed that the Strong Exponential Time Hypothesis implies that the algorithms we have for DIAMETER are pretty much the best possible. We used a statement about superpolynomial time complexity to prove a statement about fine-grained complexity!

References

[ACM96] D. Aingworth, C. Chekuri, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '96, page 547–553, USA, 1996. Society for Industrial and Applied Mathematics.

[RW12] Liam Roditty and Virginia Vassilevska Williams. Approximating the diameter of a graph, 2012.