Lecture 14 (25.01.2013) Author: Marcin Wrochna

Strong Exponential Time Hypothesis

1 Motivation

The Exponential Time Hypothesis gives us lower bounds of the form "the problem cannot be solved in time $2^{o(n)}$", or " $f(k)n^{o(k)}$ ", or a little stronger, that "there is an $\varepsilon > 0$ such that the problem cannot be solved in $2^{\varepsilon n}$ ". Still, non-trivial algorithms with time complexity of the form $\mathcal{O}(c^n)$ with $c < 2$ are known, and it would be interesting to know whether the constant can be made lower, or whether other problems have such non-trivial algorithms. With ETH alone, we don't even know how to show any particular ε, like a lower bound of 1.01^n for Independent Set. We need a much stronger assumption, which will allow us to argue that known algorithms (often the brute-force algorithms) are probably optimal.

2 Strong Exponential Time Hypothesis

Conjecture 1 (Strong Exponential Time Hypothesis [3, 1])**.**
For all $\delta < 1$ there is a k such that k-CNF-SAT cannot be solved in $\mathcal{O}(2^{\delta n})$ time.

Let $s_k = \inf\{\delta :$ there is an $\mathcal{O}(2^{\delta n})$ algorithm for k-CNF-SAT$\}$ and let $s_\infty = \lim_{k\to\infty} s_k$ (s_k is an increasing sequence bounded by 1). Then the conjecture can be restated as $s_\infty = 1$.

SETH implies that CNF-SAT cannot be solved in time $\mathcal{O}(2^{\delta n})$, not even in time $\mathcal{O}(n^{f(k)}2^{\delta n})$, for any function f and $\delta < 1$ (k being the maximum clause length). Otherwise for some $\delta' < 1$ (actually any $\delta' > \delta$) and any fixed k, the algorithm would run in time $\mathcal{O}(n^{f(k)}2^{\delta n}) \leq \mathcal{O}(2^{\delta' n})$, contradicting SETH. The conjecture is stronger, as hypothetically there could be a sequence of algorithms $\langle A_k\rangle_{k\in\mathbb{N}}$ such that each A_k solves k-CNF-SAT in $c_k 2^{\delta n}$ time, but calculating the k-th algorithm is too hard.

In particular, SETH implies that CNF-SAT cannot be solved in time $\mathcal{O}^\star(2^{\delta n}) = \mathcal{O}(poly(m)2^{\delta n})$ – the number of clauses m (equivalently, the input size) can be exponentially larger than the number of variables n, but is bounded by n^k, k being the maximum clause length.

The following fact is known, but we are not going to prove it during the course.

Fact 1 ([3])**.** *The Exponential Time Hypothesis implies that, for some constant d, $s_k \leq s_\infty(1 - \frac{d}{k})$. In particular, the sequence s_k increases infinitely often.*

No upper bound on s_k better than $1 - 1/\mathcal{O}(k)$ is known, which makes SETH plausible. Still, it is not as widely believed as ETH is. Nonetheless, reductions using SETH show a strong bound on what currently known techniques can achieve, and it can be argued that designing faster algorithms for the problem is just as hard as for the more canonical CNF-SAT.

3 Dominating Set

A brute-force algorithm for Dominating Set uses $\mathcal{O}(n^{k+1})$ time. Exercises 121,122 show how this can be brought down to $\mathcal{O}(n^{k+o(1)})$.

Theorem 2 ([4])**.** *Assuming* SETH*, for all $k \geq 3$ there is no $\mathcal{O}(n^{k-\varepsilon})$ algorithm for k-*Dominating Set *(with $\varepsilon > 0$).*

Proof. Fix $k \geq 3$. Let ϕ be a CNF formula with n variables (clauses are unbounded in size). Divide the variables into k blocks, $\frac{n}{k}$ variables each (we assume $k|n$ for simplicity). Build:

- for each block, a clique $K_{2^{n/k}}$ representing all possible valuations of variables in that block,
- for each block, an additional vertex connected to all vertices of the block's clique,
- for each clause, a vertex connected to all valuations that make it true (e.g. if the clause C contains the literal $\neg x_1$, then it is connected to all vertices of the first clique that represent a valuation of the first block having $x_1 = \texttt{false}$).

If ϕ is satisfiable, then the k-element set of valuations of the blocks is a dominating set: every clique (and the additional vertex of its block) is dominated by the valuation selected, and every clause is dominated by block valuations that make it true. If the graph has a k-element dominating set, then (because of the additional vertices) it must select at least one valuation from each clique (not the additional vertex, without loss of generality), thus exactly one (as there are k cliques), and the corresponding valuation of all variables satisfies all clauses.

The size of the graph is $n' = k(2^{n/k} + 1) + m$. Suppose there was (for any $k \geq 3$) an algorithm solving such instances of k-Dominating Set in time $\mathcal{O}((n')^{k-\varepsilon})$. Then it would solve CNF-SAT in time $\mathcal{O}(m^k \cdot k^k \cdot (2^{n/k})^{k-\varepsilon}) = \mathcal{O}(2^{n(1-\frac{\varepsilon}{k})} poly(m)) = \mathcal{O}(2^{\delta n} poly(m))$ for some $\delta < 1$, and l-CNF-SAT in time $\mathcal{O}(2^{\delta n} poly(n^l)) \leq \mathcal{O}(2^{\delta' n})$ for any $\delta' > \delta$, contradicting SETH. □

4 Hitting Set

Recall the definition:

d-Hitting Set
Input: A family $\mathcal{F} \subseteq 2^U$ ($\lvert U\rvert = n$) of sets of cardinality at most d, and an integer s
Question: Does there exist $X \subseteq U$, such that $\lvert X\rvert \leq s$ and X intersects with each $F \in \mathcal{F}$?

Theorem 3 ([2])**.** *Assume* SETH*. For all $\delta < 1$ there is a d such that d-*Hitting Set *cannot be solved in $\mathcal{O}(2^{\delta n})$ time. In particular,* Hitting Set *cannot be solved in $\mathcal{O}^\star(2^{\delta n})$ time (because it can't be solved in $\mathcal{O}(poly(n^d)2^{\delta n})$, where d is the maximum cardinality of sets in the input).*

Proof. We would like to reduce k-CNF-SAT to k'-Hitting Set, turning n variables of a fromula ϕ into a slightly larger number of elements in the universe. The simple reduction from the previous lecture created two elements for every variable – this would only give a lower bound of the form $O(\sqrt{2}^{\delta n})$.

Instead of grouping elements into blocks of 2 (representing a `true` and `false` valuation of one variable), we can group more, allowing a selection of any subset of a fixed size, to encode more information.

Fix any integer p, assume $2 \nmid p$ for simplicity and $p|n'$ (n' will be fixed later).
Let $U = \{\underbrace{e_1, \ldots, e_p}_{\text{block } 1}, \ldots, \underbrace{e_{n'-p+1}, \ldots, e_{n'-p+p}}_{\text{block } \frac{n'}{p}}\}$.
We will make the hitting set select a constant number of elements from each block – choosing $\lfloor p/2 \rfloor$ then gives the maximum number of $\binom{p}{\lfloor p/2 \rfloor}$ possibilities. These possibilities can encode all valuations of $\alpha_p := \lfloor \log_2 \binom{p}{\lfloor p/2 \rfloor} \rfloor$ variables. This allows us to encode the values of asymptotically

$$\frac{\alpha_p}{p} = \frac{\lfloor \log_2 \binom{p}{\lfloor p/2 \rfloor} \rfloor}{p} \sim \frac{\log_2(\frac{2^p}{\sqrt{p}})}{p} \xrightarrow[p\to\infty]{} 1$$

variables per element (instead of $\frac{1}{2}$ in the easier reduction).

Group variables of ϕ into blocks of size α_p. We now define the instance of k'-hitting set:

- U is a set of $n' = \frac{n}{\alpha_p} \cdot p$ elements (enough to encode all variables).
- For each block in U, add all subsets of size $\lceil p/2 \rceil + 1$ to $\mathcal{F}$. This guarantees any hitting set selects at least $\lfloor p/2 \rfloor$ elements from this block (otherwise the complement of the selected set is unhit and at least $\lceil p/2 \rceil + 1$).
- By requiring the hitting set to have at most $\frac{n'}{p}\lfloor p/2 \rfloor$ elements, we assert that it selects exactly $\lfloor p/2 \rfloor$ from each block.
- We fix the encoding of valuations of α_p variables in $\lfloor p/2 \rfloor$-element subsets of each block. Some possible subsets may be left unused (they encode nothing).
- Each clause C in ϕ contains $\leq k$ variables. They belong to $\leq k$ blocks of α_p variables each, which correspond to $\leq k$ blocks of p elements each. Let X' be the set of all variables in these variable-blocks and $U' \subseteq U$ be the set of all elements in these element-blocks. For each valuation $f : X' \to \{\texttt{true}, \texttt{false}\}$ that doesn't satisfy C, add to $\mathcal{F}$ the set $U' \setminus A_f$, where A_f is the (disjoint sum of) $\lfloor p/2 \rfloor$-subsets that encode this valuation. $|U' \setminus A_f| \leq |U'| \leq k \cdot p$.

  This asserts that any hitting set differs by some element from all encodings of valuations that falsify C. It thus selects only valuations that satisfy C. (We also add $U' \setminus A$ for sets A which encode nothing in some block, to assert that a valuation is always selected).

  (Notice that the number of solutions is preserved).

The maximum cardinality of $F \in \mathcal{F}$ is $k' \leq \max(\lceil p/2 \rceil + 1, k \cdot p) = k \cdot p$.

Suppose now that there is some $\delta < 1$ such that for each d, there is an algorithm for d-Hitting Set running in time $c_d 2^{\delta n}$. The reduction will replace n by $n' = \frac{p}{\alpha_p} n$, but those can be brought arbitrarily close, because $\frac{p}{\alpha_p} \searrow 1$. Namely, for any $\delta' > \delta$, let p be such that $\frac{p}{\alpha_p}\delta < \delta'$. Then, given any k-CNF-SAT instance of size n, we can reduce it to an instance of k'-Hitting Set with $k' = k \cdot p$ and size $n' = \frac{p}{\alpha_p} n$ and solve it in time $c_{k'} 2^{\delta n'} = c_{k'} 2^{\delta \frac{p}{\alpha_p} n} < c_{k'} 2^{\delta' n}$. This gives algorithms with a coefficient of δ' for any k-CNF-SAT (in other words, $s_k \leq \delta$ for all k), which contradicts SETH. □

5 Set Splitting

Set Splitting
Input: A family $\mathcal{F} \subseteq 2^U$ ($\lvert U\rvert = n$) of sets (of any cardinality)
Question: Does there exist $X \subseteq U$, such that for all $F \in \mathcal{F}$ both $F \cap X$ and $F \cap (U \setminus X)$ are nonempty?

Theorem 4. *Assuming* SETH, Set Splitting *cannot be solved in time* $\mathcal{O}^\star(2^{\delta n})$ *for any* $\delta < 1$.

For an instance $(U, \mathcal{F}, t)$ of Hitting Set we define an instance $(U', \mathcal{F}'_{t_1,\ldots,t_{n/p}})$ of Splitting Set for each sequence (t_i) with $0 \leq t_i \leq p$ and $\sum t_i \leq t$. We group U into blocks $U_1, \ldots, U_{n/p}$ of p elements each – we aim at hitting sets that select t_i elements from block i. We want the original instance to have a solution iff one of the new instances has a solution.

- Use the same universe with two new elements: $U' = U \cup \{a, b\}$. Add the set $\{a, b\}$ to $\mathcal{F}'_{\ldots}$. Any splitting set will choose exactly one of those elements, and a solution exists iff there is a solution X with $a \in X, b \notin X$ (because the complement of a solution is also a solution).
- For each $F \subseteq U_i$ of size $\lvert F\rvert = p - t_i + 1$, add $F \cup \{b\}$ to $\mathcal{F}'$. This asserts that any solution not containing b must select at least t_i elements from U_i (otherwise some $F \cup \{b\}$ would be all contained in the complement of the solution).
- For each $F \subseteq U_i$ of size $\lvert F\rvert = t_i+1$, add $F \cup \{a\}$ to $\mathcal{F}'$. This asserts that any solution containing a must select at most t_i elements from U_i (otherwise $F \cup \{a\}$ would be all contained in the solution).
- For all $S \in \mathcal{F}$, add $S \cup \{b\}$ to $\mathcal{F}'$. This asserts any solution not containing b must intersect every S.

The construction shows that if $(U', \mathcal{F}'_{t_1,\ldots,t_{n/p}})$ has a solution, then it has a solution containing a and not b, thus it has a solution that intersects every $S \in \mathcal{F}$ and selects exactly t_i elements from block U_i. Conversely, if there is a hitting set X selecting t_i elements from block U_i, then $X \cup \{a\}$ splits:

- every $S \cup \{b\}$ (because it hits S and doesn't hit $\{b\}$),
- every $F \cup \{a\}$ with $\lvert F\rvert = t_i + 1$ (because it hits $\{a\}$ and not all of F),
- every $F \cup \{b\}$ with $\lvert F\rvert = p - t_i + 1$ (because it hits F and doesn't hit $\{b\}$),
- and of course $\{a, b\}$.

We produce at most $(p+1)^{n/p}$ instances – one for each sequence $(t_i)_{i=1\ldots n/p}$ with $0 \leq t_i \leq p$ (and $\sum t_i = t$). This is exponential in n, but we can lower the base arbitrarily close to 1. Namely, suppose Set Splitting can be solved in time $\mathcal{O}^\star(2^{\delta n})$. Then by the above reduction we can solve Hitting Set in time $\mathcal{O}^\star(2^{\delta(n'+2)}(p+1)^{n'/p})$, where $n' = \lceil \frac{n}{p} \rceil p < n+p$, for any p. This gives at most $\mathcal{O}^\star(2^{\delta n'} 2^{n' \frac{log_2 p}{p}}) \leq \mathcal{O}^\star(2^{(\delta + \frac{log_2(p+1)}{p})(n+p)})$, but $\lim_{p\to\infty}(\delta + \frac{log_2(p+1)}{p}) = \delta$, therefore for any $\delta' > \delta$ we can fix a p that gives an algorithm for Hitting Set running in time $\mathcal{O}^\star(2^{\delta'(n+p)}) = \mathcal{O}^\star(2^{\delta' n})$, which contradicts the previous theorem.

Exercises 119 and 120 show that, through simple reductions from Splitting Set to NAE-SAT and from that back to CNF-SAT, the Strong Exponential Time Hypothesis is equivalent to similar statements for Hitting Set, Splitting Set and NAE-SAT.

References

[1] C. Calabro, R. Impagliazzo, and R. Paturi. The complexity of satisfiability of small depth circuits. *Parameterized and Exact Computation*, pages 75–85, 2009.

[2] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlstrom. On problems as hard as cnf-sat. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 74–84. IEEE, 2012.

[3] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62:367–375, 2001.

[4] M. Pătraşcu and R. Williams. On the possibility of faster sat algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1065–1075. Society for Industrial and Applied Mathematics, 2010.