Fast and Deterministic Constant Factor Approximation Algorithms for LCS Imply New Circuit Lower Bounds

Amir Abboud[1] **and Aviad Rubinstein**[*2]

1 **IBM Almaden Research Center, San Jose, CA, USA**
amir.abboud@ibm.com

2 **Department of Computer Science, Harvard University, Cambridge, MA, USA**
aviad@seas.harvard.edu

Abstract

The Longest Common Subsequence (LCS) is one of the most basic similarity measures and it captures important applications in bioinformatics and text analysis. Following the SETH-based nearly-quadratic time lower bounds for LCS from recent years [4, 22, 5, 3], it is a major open problem to understand the complexity of approximate LCS. In the last ITCS [2] drew an interesting connection between this problem and the area of circuit complexity: they proved that approximation algorithms for LCS in deterministic truly-subquadratic time imply new circuit lower bounds (E^{NP} does not have non-uniform linear-size Valiant Series Parallel circuits).

In this work, we strengthen this connection between approximate LCS and circuit complexity by applying the *Distributed PCP* framework of [6]. We obtain a reduction that holds against much larger approximation factors (super-constant, as opposed to $1 + o(1)$ in [2]), yields a lower bound for a larger class of circuits (linear-size NC^1), and is also easier to analyze.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Distributed PCP, Longest Common Subsequence, Fine-Grained Complexity, Circuit Lower Bounds, Strong Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.35

1 Introduction

The Longest Common Substring (LCS) is a fundamental similarity measure between two strings with many important applications to data comparison. It is an elegant abstraction of the core task in sequence alignment tasks in bioinformatics. Given two strings of length N, there is a classical dynamic programming algorithm that computes the LCS in time $O(N^2)$. The quadratic time requirement is prohibitive for very long strings (e.g. genomic sequences), and obtaining a substantially faster algorithm is a longstanding and central open question. In practice, biologists use heuristics such as BLAST to solve it in near-linear time but without any guarantees on the optimality of the solution [8]. Interesting results from recent years [10, 4, 22, 5] showed that under certain complexity assumptions such as "SETH"[1], there are no truly subquadratic algorithms for this problem.

* Research supported by a Rabin Postdoctoral Fellowship.

[1] The Strong Exponential Time Hypothesis (SETH) [35] postulates that we cannot solve k-SAT in $O((2-\varepsilon)^n)$ time, for some $\varepsilon > 0$ independent of and for all constant k.

For many applications, we would be happy to settle for an approximate solution if it could be found in truly subquadratic (ideally near-linear) time. In contrast to the exact variant, the complexity of approximating the LCS is poorly understood. On the algorithmic side, for strings over alphabet Σ there is a trivial $1/|\Sigma|$-approximation algorithm (using only the symbol that appears the most in both strings). On the complexity sides, the known reductions from SETH tell us very little about hardness of approximation. There are two main obstacles to obtaining SETH-based hardness of approximation:

The PCP blowup. The PCP Theorem, which is a crucial step in most NP-hardness of approximation results, can be seen as reducing the satisfiability of a CNF φ to approximate satisfiability of a new CNF φ'. If φ has n variables, the most efficient PCP constructions construct φ' with $n' = n \cdot \mathsf{polylog}(n)$ variables [28]. Obtaining a linear dependence is a major open problem (e.g. [17, 29, 41, 40, 24]).

In contrast, the known reductions from SETH to LCS begin with a CNF over n variables, and transform it to a hard instance of LCS of string length $N \approx 2^{n/2}$. Now solving LCS faster than $N^2 \approx 2^n$ time implies new algorithms for SAT. Even if we had a fantastic PCP of blowup $n' = 10n$, if we begin the reduction with the hard-to-approximate CNF φ', we would get strings of length $N' \approx 2^{n'/2} \approx 2^{5n}$.

The PCP blowup obstacle is common to almost all known reductions from SETH. For several other problems, this obstacle was addressed by the *distributed PCP*[2] framework in [6]. However, this technique does not yet suffice for hardness of approximation of LCS, in part due to the "contribution of unsatisfying assignments" obstacle described next.

Contribution of unsatisfying assignments. The second obstacle is more specific to LCS (and a few other string similarity measures like Edit Distance and Dynamic Time Warping Distance). The reduction from SETH proceeds by concatenating approximately $N \approx 2^{n/2}$ gadgets, one for each partial assignment to half of the variables. On a yes instance, matching the two gadgets that correspond to the satisfying assignment gives a higher contribution to the LCS than matching any two non-satisfying partial assignments. How much more can the satisfying pair contribute to the LCS compared to a non-satisfying pair? The largest gap we can hope to construct is a $|\Sigma|$-factor, because of the trivial $1/|\Sigma|$-approximation algorithm. If we want to keep the alphabet size small ($|\Sigma| = N^{o(1)}$), this is still negligible compared to the contribution from approximately N (disjoint) pairs of non-satisfying partial assignments.

In short, an inherent limitation of all known reduction techniques is that the multiplicative approximation of the LCS in the resulting instances can be obtained from an additive approximation of the fraction of satisfying assignments. The latter can be computed easily by sampling a small number of uniformly random assignments.

In the last ITCS, [2] drew an interesting connection between this problem and classical questions on circuit lower bounds. The authors observed that designing a *deterministic* algorithm that can approximate the number of satisfying assignment to formulas (that are slightly more complex than CNFs) is a challenging task with connections to pseudorandomness, and so this barrier can be circumvented, in a certain sense, if we restrict the attention to deterministic approximation algorithms. In particular, they show that if LCS admits a *deterministic* truly-subquadratic approximation algorithm, then certain long sought-after circuit lower bounds would be implied. However, to obtain such consequences, one would need to design a very good $(1 + o(1))$ approximation algorithms for LCS.

[2] The term *distributed PCP* was first used by Drucker [30], but in a completely different context.

Given the above two, it is natural to ask whether we can combine the techniques of [2] and [6] to obtain stronger inapproximability for LCS in deterministic truly subquadratic time. In this paper we show that this is indeed the case, and in a strong sense: using the distributed PCP framework from [6], we can replace the not-even-1.001 hardness of approximation factor from [2] with superconstant factors.

Additionally, using distributed PCPs also allows us to obtain stronger circuit lower bounds than [2]. The circuit complexity consequence of deterministic and fast approximate LCS algorithms established by [2] was that E^{NP} does not have non-uniform linear-size Valiant Series Parallel (VSP) circuits. In the 1970's Valiant defined the VSP property and argued that it is common in algorithmic circuits. This consequence is still out of reach of current techniques and is typically reported as a circuit lower bound from refuting SETH [37]. Here, we show that the VSP restriction can be replaced by the more natural restriction that the circuits have logarithmic depth (NC^1). Intuitively, the latter are more expressive. Proving the following statement would be a major breakthrough in complexity theory.

▶ **Consequence 1.1.** *The class E^{NP} does not have non-uniform circuits of size $O(n)$ and depth $O(\log n)$, nor VSP circuits of size $O(n)$.*

We are now ready to present our main theorem:

▶ **Theorem 1.2** (Main Theorem)**.** *If there is an algorithm that, given two length-N strings x, y over alphabet Σ, where $|\Sigma| = N^{o(1)}$, approximates the LCS of x and y to within any constant factor in deterministic, $O(N^{2-\varepsilon})$ time, then Consequence 1.1 follows.*

1.1 A succinct discussion of techniques

A sequence of previous works [34, 53, 50, 18, 2] reduces the task of proving Consequence 1.1 to designing deterministic algorithms for the following problem: given an OR with fan-in $2^{o(n)}$ over k-CNFs, for $k = O(n^{0.1})$, approximate the fraction of satisfying assignments (Lemma 2.4) in $\mathsf{DTIME}(2^n/n^{\omega(1)})$. The outer OR is easy to implement, and so we focus on any given CNF.

For CNFs with constant clause width, a common first step is to use the Sparsification Lemma of [36] which reduces the number of clauses to $m = O(n)$. However, the O-notation in this lemma hides a blowup which is doubly exponential in the width, so in our case ($k \approx n^{0.1}$) we are better off sticking with the trivial bound on the number of clauses: $m \lessapprox \binom{n}{n^{0.1}} \approx 2^{n^{0.1}}$.

As we mentioned earlier, the reduction constructs a gadget for each possible assignment to the first (resp. last) half of the variables. We want the LCS of the two gadgets to implement a verifier that receives the two assignments and verifies that they indeed satisfy the CNF. A key observation in [6] is that this task reduces to solving a Set Disjointness problem over the universe of clauses $[m]$: Given partial assignment $\alpha \in \{0,1\}^{n/2}$ to the first half of the variables, Alice locally constructs the set $S_\alpha \subseteq [m]$ of clauses that are not satisfied by α (but she still hopes those clauses are satisfied by the assignment to the remaining variables). Similarly, Bob locally constructs a set $T_\beta \subseteq [m]$ of clauses that he cannot guarantee are satisfied by his partial assignment, β. Now the joint assignment (α, β) satisfies the CNF if and only if S_α, T_β are disjoint.

Observe that two sets are disjoint iff their representation as binary vectors (in $\{0,1\}^m$) are orthogonal (over the reals). Indeed, so far our reduction looks like the classical reduction to the Orthogonal Vectors problem [52]: given two sets $A, B \in \{0,1\}^m$, is there a pair $a \in A, b \in B$ that is orthogonal?

Set Disjointness is a rather difficult problem in Communication Complexity: the randomized and even non-deterministic complexities of Set Disjointness are linear. Fortunatley,

there is an $\tilde{O}(\sqrt{m})$ MA-communication protocol due to [1]. In [6] m was linear, so we could enumerate over all protocols in subexponential time. Here, since $m \approx 2^{n^{0.1}}$, the quadratic saving of the MA-protocol does not help. Instead, we use an IP-protocol for Set Disjointness (also due to [1]) which uses only $\tilde{O}(\log m) \approx n^{0.1}$ communication.

We abstract the first part of the reduction (up to and including the IP-protocol) via a new problem a-la Orthogonal Vectors, that we call Tropical Tensors. Given two lists of tensors $A, B \in \{0,1\}^{[d_1]\times[d_2]\times\cdots\times[d_t]}$, we want to find a pair $a \in A, b \in B$ that maximizes a similarity measure $s(a,b)$ which is defined via a chain of alternating $+$ and max operations, where at the base we take the product of a_i and b_i; we call this the *Tropical Similarity* [3] of a and b.

Similar to Orthogonal Vectors, our new problem allows us to abstract out the PCP-like construction on one hand, and the LCS-specific gadgets on the other hand. While its definition is somewhat more involved than the original Orthogonal Vectors, the extra expressive power allows us to prove a stronger hardness of approximation result: Consequence 1.1 is implied by any truly subquadratic deterministic algorithm that can distinguish between the case where almost all pairs have almost maximum Tropical Similarity, and the case where the Tropical Similarity of every pair is tiny (see Theorem 3.2 for details). We hope that the Tropical Tensors problem will find further applications; see Remark 3.4 for some suggestions.

Once we establish the hardness of Tropical Tensors, we reduce it (Section 4) to LCS using gadgets that implement $+$ and max operations. Our reduction to LCS is particularly simple because the gap we obtain from Tropical Tensors is so large, that we do not need to pad our gadgets to enforce well-behaved solutions.

1.2 Related work

Algorithms for LCS and related problems

Even though many ideas and heuristics for LCS were designed [25, 19, 27, 26] (see also [43, 20] for surveys), none has proven sufficient to compute a better than $|\Sigma|$ approximation in strongly subquadratic time.

Many ingenious approximation algorithms were discovered for the related Edit Distance problem. A linear time $\sqrt{n}$-approximation follows from the exact algorithm that computes the Edit Distance in time $O(n+d^2)$ where $d = ED(S,T)$ [39]. Subsequently, this approximation factor has been improved to $n^{3/7}$ by Bar-Yossef et al. [12], then to $n^{1/3+o(1)}$ by Batu et al. [13]. Building on the breakthrough embedding of Edit Distance by Ostrovsky and Rabani [44], Andoni and Onak obtained the first near-linear time algorithm with a *subpolynomial* approximation factor of $2^{\tilde{O}(\sqrt{\log n})}$. Most recently, Andoni, Krauthgamer, and Onak [9] significantly improved the approximation to polylogarithmic obtaining an algorithm that runs in time $n^{1+\varepsilon}$ and gives $(\log n)^{O(1/\varepsilon)}$ approximation for every fixed $\varepsilon > 0$. There are many works on approximate Edit Distance in various computational models, see e.g. [43, 9, 23] and the references therein. It remains a huge open question whether Edit Distance can be approximated to within a constant factor in near-linear time.

A general tool for speeding up dynamic programming algorithms through a controlled relaxation of the optimality constraint is highly desirable. Encouraging positive results along these lines were recently obtained by Saha [47, 48, 49] for problems related to parsing

[3] The name is inspired by Tropical Algebras, which support $+$ and min operations.

context-free languages. However, we are still far from understanding, more generally, when and how such algorithms are possible.

Fine-grained complexity of LCS

Many hardness results have been recently shown for LCS. Shortly after the $N^{2-o(1)}$ SETH-based lower bound of Backurs and Indyk [10] for the related problem of Edit Distance, it was proven that LCS has a similar lower bound [4, 22]. Bringmann and Kunnemann [22] proved that the SETH lower bound holds even when the strings are binary, and [4] prove that LCS on k strings has an $N^{k-o(1)}$ lower bound. Very recently, [3] prove that the time complexity of computing the LCS between strings of length N that are compressed down to size n (using any of the standard grammar compressions such as Lempel-Ziv) is lower bounded by $(Nn)^{1-o(1)}$ under SETH, and a matching upper bound is known [31].

[5] proved quadratic lower bounds for LCS under safer versions of SETH where CNF is replaced with NC circuits, and connected LCS to circuit lower bounds for the first time. [5] also showed that even mildly subquadratic algorithms for LCS, e.g. $O(N^2/\log^{50} N)$ would imply breakthrough circuit lower bounds similar to Consequence 1.1. This connection to circuit lower bounds was exploited in the work of [2] who showed that such consequences can follow even from approximation algorithms.

The only SETH-based hardness of approximation results for LCS are for variants of the classical problem. For instance, approximate "closest pair" under the LCS similarity requires nearly quadratic time even for $2^{(\log N)^{1-o(1)}}$ approximation factors, under SETH [6].

Distributed PCP

As mentioned above, when viewing the PCP Theorem as a reduction from CNF to hard-to-approximate CNF, all known constructions suffer from blowup in the number of variables, which is prohibitive for fine-grained reductions. Another (in fact, the original) way to view the PCP Theorem is as a *probabilistically checkable proof*: given an assignment $x \in \{0,1\}^n$, we want to write a proof $\pi(x)$ asserting that x satisfies a known CNF φ. *Probabilistically checkable* means that the verifier should be able to query $\pi(x)$ at a small number of random locations to be convinced that φ has a satisfying assignment. Recall that most reductions from SETH to quadratic time problems construct a gadget for each partial assignment to φ. A key observation in [6] is that if we could construct a "partial PCP" for each partial assignment, the total number of gadgets remains approximately $2^{n/2}$, even if each gadget is now a little bit larger.

Thus, in the *Distributed PCP* challenge, we have two parties (Alice and Bob) who hold partial assignments $\alpha, \beta \in \{0,1\}^{n/2}$ to disjoint subsets of the variables, and want to prove to a verifier that their joint assignment satisfies the public CNF φ. A second key observation in [6] is that this challenge is equivalent to computing Set Disjointness over subsets of the clauses of φ. [6] solved this Set Disjointness problem using (a variant of) the MA-communication protocol of [1]. Here, we need the more efficient IP-communication protocol.

Other PCPs in non-standard models

Different models of "non-traditional" PCPs, such as interactive PCPs [38] and interactive oracle proofs (IOP) [16, 46] have been considered and found "positive" applications in cryptography (e.g. [32, 33, 16]). In particular, [15] obtain a linear-size IOP. It is an open question whether these interactive variants can imply interesting hardness of approximation results [15].

SETH and communication complexity

The connection between the SETH and communication complexity goes back at least to [45] who proved that a computationally efficient sublinear protocol for the 3-party Number-on-Forehead Set Disjointness problem would refute SETH.

2 Preliminaries

2.1 Derandomization and Circuit Lower Bounds

This result will utilize the connection between derandomization and circuit lower bounds which originates in the works of Impagliazzo, Kabanets, and Wigderson [34] and has been optimized significantly by the work of Williams [53], Santhanam and Williams [50], and more recently by Ben-Sasson and Viola [18]. These connections rely on "Succinct PCP" theorems [42, 18], and the recent optimized construction of Ben-Sasson and Viola [18] is essential for our results. Our starting point is the following theorem.

▶ **Theorem 2.1** (Theorem 1.4 in [18]). *Let F_n be a set of function from $\{0,1\}^n$ to $\{0,1\}$ that are efficiently closed under projections (see [18] or Definition 10 in [2]).*

If the acceptance probability of a function of the form

- *AND of fan-in $n^{O(1)}$ of*
- *OR's of fan-in 3 of*
- *functions from $F_{n+O(\log n)}$*

can be distinguished from being $= 1$ or $\leq 1/n^{10}$ in $DTIME(2^n/n^{\omega(1)})$, then there is a function f in E^{NP} on n variables such that $f \notin F_n$.

We apply this theorem where the class F_n is the class of circuits on n variables of size cn, for an arbitrarily large constant $c > 0$, and depth upper bounded by $c \log n$. By applying the deMorgan rule, and then noticing that linear-size log-depth circuits are closed under negations and OR's, we can restate the above theorem as follows (see [2] for a more detailed argument).

▶ **Lemma 2.2.** *To prove that E^{NP} does not have non-uniform circuits of size cn and depth $c \log n$ on n input variables, it is enough to show a deterministic algorithm for the following problem that runs in $2^n/n^{\omega(1)}$ time. Given a circuit over n input variables of the form:*

- *OR of fan-in $n^{O(1)}$ of*
- *circuits of size $3cn$ and depth $c \log n + 2$,*

distinguish between the case where no assignments satisfy it, versus the case in which at least $a \geq 1 - 1/n^{10}$ fraction of the assignments satisfy it.

2.2 Valiant's depth reduction

We will use the classical depth-reduction theorem of Valiant [51] to convert linear-size NC^1 circuits into an OR of CNF's, on which we will apply our distributed PCP techniques. The elegant proof is often given in courses, see e.g. [14].

▶ **Theorem 2.3** (Depth reduction [51]). *For all $\varepsilon > 0$ and $c \geq 1$, we can convert any circuit on n variables of size cn and depth $c \log n$, into an equivalent formula which is OR of $2^{f(c,\varepsilon)\cdot(n/\log\log n)}$ k-CNF's on the same n variables, where $k = O(n^{\varepsilon})$. The reduction runs in $2^{O(n/\log\log n)}$ time.*

We remark that if the circuit is assumed to have the additional "series parallel" property [51], then we can get a stronger depth reduction result where the clause size in the CNF's is constant. This was crucial to the results of [2], but our techniques here allow us to handle much larger CNF's.

Combining Valiant's depth reduction with Lemma 2.2 we conclude that to prove our complexity consequence, it is enough to distinguish unsatisfiable from $> 99\%$ satisfiable on circuits of the form: OR of CNF's with clause size n^{ε}.

▶ **Lemma 2.4.** *To prove Consequence 1.1, it is enough to show a deterministic algorithm for the following problem that runs in $2^n/n^{\omega(1)}$ time. Given a circuit over n input variables of the form:*

- *OR of fan-in $2^{O(n/\log\log n)}$ of*
- *k-CNF's where $k = O(n^{0.1})$,*

distinguish between the case where no assignments satisfy it, versus the case in which at least $a \geq 1 - 1/n^{10}$ fraction of the assignments satisfy it.

2.3 Communication complexity

We use the following IP-communication protocol due to Aaronson and Wigderson [1].

▶ **Theorem 2.5** (Essentially [1, Section 7])**.** *There exists a computationally efficient* [4] IP*-protocol for Set Disjointness over domain $[m]$ in which:*

1. *Merlin and Alice exchange $O(\log m \log\log m)$ bits;*
2. *Bob learns the outcome of $O(\log m \log\log m)$ coins tossed by Alice during the protocol;*
3. *Bob sends Alice $O(\log m)$ bits.*
4. *Alice returns* Accept *or* Reject.

If the sets are disjoint, Alice always accepts; otherwise, Alice rejects with probability at least $1/2$.

3 A surrogate problem: Tropical Tensors

In this section we introduce a new problem a-la Orthogonal Vectors, and show that approximating it with a truly subquadratic determinstic algorithm would be enough to prove the breakthrough Consequence 1.1.

▶ **Definition 3.1** (Tropical Tensors)**.** Our similarity measure s is defined with respect to parameters t and $\ell_1, \ldots, \ell_t$. For two tensors $u, v \in \{0,1\}^{d_1 \times \cdots \times d_t}$, we define their *Tropical Similarity* score with an alternating sequence of E (expecation) and max operators:

$$s(u,v) \triangleq \mathsf{E}_{i_1 \in d_1}\Big[\max_{i_2 \in d_2}\Big\{\mathsf{E}_{i_3 \in d_3}\Big[\cdots \max_{i_t \in d_t}\{u_i \cdot v_i\}\cdots\Big]\Big\}\Big].$$

Given two sets of tensors $A, B \in \{0,1\}^{d_1 \times \cdots \times d_t}$, the Tropical Tensors problem asks to find a pair $a \in A, b \in B$ that maximizes the Tropical Similarity $s(a,b)$.

▶ **Theorem 3.2.** *Let $d_1, \ldots, d_t$ be such that $d_1 d_2 \cdots d_t = N^{o(1)}$. To prove Consequence 1.1 it is enough to design a deterministic $O(N^{2-\varepsilon})$-time algorithm that, given two sets of tensors $A, B \in \{0,1\}^{d_1 \times \cdots \times d_t}$, distinguishes between the following:*

[4] Although [1] do not explicitly consider computational efficiency, it is not hard to make their protocol computationally efficient.

Completeness: A $(1 - 1/\log^{10} N)$*-fraction of the pairs* a, b *have a perfect Tropical Similarity score,* $s(a, b) = 1$.
Soundness: Every pair has low Tropical Similarity score, $s(a, b) = o(1)$.

▶ **Remark 3.3.** Our hardness of approximation for Tropical Tensors continues to hold even in the special case where we only take max's with respect to coordinates of A-tensors. In other words, we could redefine

$$s'(a, b) \triangleq \mathsf{E}_{i_1 \in d_1}\Big[\max_{j_2 \in d_2}\Big\{\mathsf{E}_{i_3 \in d_3}\Big[\cdots \max_{j_t \in d_t}\{a_{i,j} \cdot b_i\} \cdots\Big]\Big\}\Big].$$

(Note that now b is of smaller dimension.)

▶ **Remark 3.4.** Another interesting approximation variant of Tropical Tensors is the challenge of distinguishing between the sets A, B containing at least one pair with perfect Tropical Similarity ($s(a, b) = 1$) versus every pair having subconstant Tropical Similarity ($s(a, b) = o(1)$). Following the same proof outline, one could prove an analog of Theorem 3.2, whereby no $O(N^{2-\varepsilon})$-time algorithms (deterministic or randomized) solve the above problem, assuming SETH for circuits of linear size and logarithmic depth.

Thus we can obtain variants of hardness of approximation results from [6] for LCS Closest Pair, Approximate Regular Expression Matching, and Diameter in Product Metric, based on the latter assumption, which is safer than the standard SETH (i.e. SETH of k-CNF for every constant k).

Proof. Lemma 2.4 tells us that in order to prove Consequence 1.1, it is enough solve the derandomization problem on circuits of the form: an OR over $2^{O(n/\log\log n)}$ CNFs, with clause width $O(n^{0.1})$. In particular, each CNF has at most $m \triangleq 2^{\tilde{O}(n^{0.1})}$ clauses. The focus of this proof will be on reducing a single such CNF to Tropical Tensors. To reduce the OR over $2^{O(n/\log\log n)}$ CNFs to Tropical Tensors, we simply take the max over the Tropical Similarity scores constructed for each CNF.

We do the following for each CNF in the OR. The set A will contain a tensor a^α for each half-assignment $\alpha \in \{0,1\}^{n/2}$ to the CNF. Given half assignment α, let $S_\alpha \subset [m]$ be the set of clauses that it does *not* satisfy, i.e. all the literals determined by α are false. Define B, β, and T_β analogously. Observe that the CNF is satisfied by a pair (α, β) iff the sets S_α, T_β are disjoint.

Recall the IP-communication protocol for Set Disjointness (Theorem 2.5). To obtain subconstant soundness, amplify the soundness of the protocol by repeating a small superconstant number (e.g. $\log\log m$) of times.

We construct the tensors a^α and b^β recursively, using the IP protocol. Each dimension of the tensors corresponds to a message from one of the parties or a coin toss. Each entry corresponds to an entire transcript. Notice that since the total communication complexity is $\tilde{O}(\log m) = \tilde{O}(n^\varepsilon)$, the total number of possible transcripts is at most $d_1 d_2 \cdots d_t = 2^{\tilde{O}(n^\varepsilon)} = N^{o(1)}$.

At the end of the protocol, Bob sends a message. Let $[d_t]$ enumerate over all of Bob's potential messages. For each $i_{-t} \in d_1 \times d_2 \times \cdots \times d_{t-1}$, we set ${a^\alpha}_i \triangleq 1$ iff Alice accepts on message i_t from Bob at the end of the protocol (otherwise, ${a^\alpha}_i \triangleq 0$). Similarly, we set ${b^\beta}_i \triangleq 1$ iff i_t is the message that Bob sends. Hence the contribution to the Tropical Similarity is one iff Alice accepts Bob's message at the end of the protocol.

For the rest of the coordinates we take E over random coin tosses, and max over Merlin's potential messages. Hence the Tropical Similarity $s(a^\alpha, b^\beta)$ is exactly equal to the probability that Alice accepts at the end of the IP protocol given for input sets S_α, T_β.

Hence there is a one-to-one correspondence between satisfying assignments and pairs with perfect Tropical Similarity score; similarly there is a one-to-one correspondence between unsatisfying assignments and pairs with subconstant Tropical Similarity score.

Finally, we add another outside max to account for the large OR over $2^{O(n/\log\log n)}$ CNFs. ◀

4 LCS

In this section we provide a gap-preserving reduction from Tropical Tensors to Longest Common Subsequence. Together with Theorem 3.2 this completes our proof of Theorem 1.2.

Proof of Theorem 1.2. We begin with the hard instance of Tropical Tensors from Theorem 3.2. We encode each of the tensors as a string-gadget over alphabet Σ, and then concatenate all the gadgets (in arbitrary order). Unlike previous fine-grained reductions for LCS and related problems (e.g. [7, 10, 4, 22, 5, 11, 2, 21]), we do not need any padding between gadgets, since the gap we obtained for Tropical Tensors is so large.

Bit gadgets

We construct the gadget for each tensor recursively. At the base of our recursion, we use the following encoding for each bit. For each coordinate $i \in [d_1] \times \cdots \times [d_t]$, we reserve a special symbol $i \in \Sigma$. We will also have two special symbols $\perp^A, \perp^B \in \Sigma$. Thus in total $|\Sigma| = d_1 d_2 \cdots d_t + 2 = N^{o(1)}$. Finally, we are ready to define the bit-gadgets:

$$x_i(a) \triangleq \begin{cases} i & a_i = 1 \\ \perp^A & a_i = 0 \end{cases},$$

and

$$y_i(b^\beta) \triangleq \begin{cases} i & b_i = 1 \\ \perp^B & b_i = 0 \end{cases}.$$

Observe that now $LCS\left(x_i(a), y_i(b)\right) = a_i \cdot b_i$.

Tensor gadgets

We now recursively combine gadgets to implement the max and E operators. In order to implement max operators, we concatenate the corresponding x-gadgets, and concatenate the respective y-gadgets in reverse order. For example, for any fixed choice of $i_{-t} = (i_1, \ldots, i_{t-1})$, we combine bit-gadgets across the last dimension as follows:

$$x_{i_{-t}}(a) \triangleq x_{i_{-t},1}(a) \circ x_{i_{-t},2}(a) \circ \cdots \circ x_{i_{-t},d_t}(a)$$
$$y_{i_{-t}}(b) \triangleq y_{i_{-t},d_t}(b) \circ \cdots \circ y_{i_{-t},2}(b) \circ y_{i_{-t},1}(b).$$

Notice that we now have that $LCS\left(x_{i_{-t}}(a), y_{i_{-t}}(b)\right) = \max_{i_t \in [d_t]} LCS\left(x_{i_{-t},i_t}(a), y_{i_{-t},i_t}(b)\right)$.

We implement summations (E), we concatenate both the x and the y gadgets in the same order. For example, for the first dimension, we define:

$$x(a) \triangleq x_1(a) \circ x_2(a) \circ \cdots \circ x_{d_1}(a)$$
$$y(b) \triangleq y_1(b) \circ y_2(b) \circ \cdots \circ y_{d_1}(b).$$

Notice that we now have that $LCS\left(x(a), b(a)\right) = d_1 \cdot \mathsf{E}_{i_1 \in [d_1]}\left[LCS\left(x_{i_1}(a), y_{i_1}(b)\right)\right]$.

Therefore, by induction, we have that

$$\begin{aligned}\frac{1}{D} LCS\left(x(a), y(b)\right) &= \mathsf{E}_{i_1 \in d_1}\left[\max_{i_2 \in d_2}\left\{\mathsf{E}_{i_3 \in d_3}\left[\cdots \max_{i_t \in d_t}\{LCS\left(x_i(a), y_i(b)\right)\} \cdots\right]\right\}\right] \\ &= \mathsf{E}_{i_1 \in d_1}\left[\max_{i_2 \in d_2}\left\{\mathsf{E}_{i_3 \in d_3}\left[\cdots \max_{i_t \in d_t}\{a_i \cdot b_i\} \cdots\right]\right\}\right] \\ &= s(a, b),\end{aligned}$$

where $D \triangleq d_1 d_3 d_5 \cdots d_{t-1}$ is the normalization factor.

The final strings

Finally, we construct the strings x, y by concatenating the $2^{n/2}$ tensor gadgets. We call a pair of tensors a, b "good" if $s(a, b) = 1$ and "bad" if $s(a, b) = o(1)$.

Completeness

Assume that there are at least $(1 - 1/\log^{10} n) \cdot n^2$ good pairs of tensors. We consider a set of $2n - 1$ alignments between x and y: For each shift $k \in [2n - 1]$ define the alignment $\mathcal{A}_k$ that matches the tensor gadget of tensor $a_i \in A$ to the tensor gadget of $b_j \in B$ *optimally* where $j = i + k - n$, and if $j \notin [n]$ then we do not match the gadget of a_i at all. Since the alignments of gadgets to each other are made optimally, their contribution is exactly $LCS(x(a_i), y(b_j)) = D \cdot s(a_i, b_j)$. Observe that for all $i, j \in [n]$ there is exactly one k such that the gadgets of a_i and b_j are matched in $\mathcal{A}_k$, and so the total LCS score of all of these $2n - 1$ alignments is at least

$$(1 - 1/\log^{10} n) \cdot n^2 \cdot D \cdot 1.$$

Therefore at least one of these alignments has score more than $D \cdot n/2$.

Soundness

Assume that all pairs of tensors are bad. In this case, any alignment between two tensor gadgets has score at most $LCS(x(a_i), y(b_j)) = o(1) \cdot D$. We can upper bound the score of any alignment between x and y by upper bounding the number of tensor gadgets participating in the alignment. We say that a pair is participating in the alignment if any of their letters are matched to each other. Due to the non-crossing nature of alignments, we can model all pairs participating in an alignment as the edges in a bipartite *planar* graph, and it follows that there can be at most $2n$ such edges. Therefore, the score of any alignment is upper bounded by $o(1) \cdot D \cdot 2n$. ◀

Acknowledgements. We thank Scott Aaronson, Mika Goos, Elad Haramaty, and Ryan Williams for helpful discussions and suggestions.

References

1 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1):2:1–2:54, 2009. doi:10.1145/1490270.1490272.

2 Amir Abboud and Arturs Backurs. Towards hardness of approximation for polynomial time problems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 11:1–11:26. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.11.

3 Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. Fine-grained complexity of analyzing compressed data: Quantifying improvements over decompress-and-solve. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 192–203. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.26.

4 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *Proc. of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 59–78, 2015.

5 Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In *Proc. of the 48th STOC*, pages 375–388, 2016.

6 Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.12.

7 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *Proc. of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 39–51, 2014.

8 Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

9 Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *FOCS*, pages 377–386, 2010.

10 Arturs Backurs and Piotr Indyk. Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false). In *Proc. of the 47th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 51–58, 2015.

11 Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? In *Proc. of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 457–466, 2016.

12 Ziv Bar-Yossef, TS Jayram, Robert Krauthgamer, and Ravi Kumar. Approximating edit distance efficiently. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 550–559. IEEE, 2004.

13 Tuğkan Batu, Funda Ergun, and Cenk Sahinalp. Oblivious string embeddings and edit distance approximations. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 792–801. Society for Industrial and Applied Mathematics, 2006.

14 Paul Beame. Lecture notes on circuit reducibility, depth reduction, and parallel arithmetic, April 2008.

15 Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Short interactive oracle proofs with constant query complexity, via composition and sumcheck. *IACR Cryptology ePrint Archive*, 2016:324, 2016. URL: http://eprint.iacr.org/2016/324.

16 Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International*

Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II, volume 9986 of *Lecture Notes in Computer Science*, pages 31–60, 2016. doi:10.1007/978-3-662-53644-5_2.

17 Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. Constant rate pcps for circuit-sat with sublinear query complexity. *J. ACM*, 63(4):32:1–32:57, 2016. doi:10.1145/2901294.

18 Eli Ben-Sasson and Emanuele Viola. Short pcps with projection queries. In *ICALP, Part I*, pages 163–173, 2014.

19 Lasse Bergroth, Harri Hakonen, and Timo Raita. New approximation algorithms for longest common subsequences. In *String Processing and Information Retrieval: A South American Symposium, 1998. Proceedings*, pages 32–40. IEEE, 1998.

20 Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

21 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. A dichotomy for regular expression membership testing. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 307–318. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.36.

22 Karl Bringmann and Marvin Kunnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proc. of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 79–97, 2015.

23 Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 712–725. ACM, 2016. doi:10.1145/2897518.2897577.

24 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.74.

25 F Chin and Chung Keung Poon. Performance analysis of some simple heuristics for computing longest common subsequences. *Algorithmica*, 12(4-5):293–311, 1994.

26 Maxime Crochemore, Costas S Iliopoulos, Yoan J Pinzon, and James F Reid. A fast and practical bit-vector algorithm for the longest common subsequence problem. *Information Processing Letters*, 80(6):279–285, 2001.

27 J Boutet de Monvel. Extensive simulations for longest common subsequences. *The European Physical Journal B-Condensed Matter and Complex Systems*, 7(2):293–308, 1999.

28 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

29 Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016. URL: http://eccc.hpi-web.de/report/2016/128.

30 Andrew Drucker. PCPs for Arthur-Merlin Games and Communication Protocols. Master's thesis, Massachusetts Institute of Technology, 2010. URL: http://people.csail.mit.edu/andyd/Drucker_SM_thesis.pdf.

31 Paweł Gawrychowski. Faster algorithm for computing the edit distance between slp-compressed strings. In *International Symposium on String Processing and Information Retrieval*, pages 229–236. Springer, 2012.

32 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27:1–27:64, 2015. doi:10.1145/2699436.

33 Vipul Goyal, Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. Interactive locking, zero-knowledge pcps, and unconditional cryptography. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2010. doi:10.1007/978-3-642-14623-7_10.

34 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

35 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

36 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

37 Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *International Colloquium on Automata, Languages, and Programming*, pages 749–760. Springer, 2015.

38 Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008. doi:10.1007/978-3-540-70583-3_44.

39 Gad M Landau, Eugene W Myers, and Jeanette P Schmidt. Incremental string comparison. *SIAM Journal on Computing*, 27(2):557–582, 1998.

40 Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 954–961. ACM, 2017. doi:10.1145/3055399.3055412.

41 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 78:1–78:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.

42 Thilo Mie. Short pcpps verifiable in polylogarithmic time with o (1) queries. *Annals of Mathematics and Artificial Intelligence*, 56(3-4):313–338, 2009.

43 Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

44 Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *Journal of the ACM (JACM)*, 54(5):23, 2007.

45 Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.

46 Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Comput-*

ing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016, pages 49–62. ACM, 2016. doi:10.1145/2897518.2897652.

47 Balaram Saha. The dyck language edit distance problem in near-linear time. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 611–620. IEEE, 2014.

48 Barna Saha. Language edit distance and maximum likelihood parsing of stochastic grammars: Faster algorithms and connection to fundamental graph problems. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 118–135. IEEE, 2015.

49 Barna Saha. Fast & space-efficient approximations of language edit distance and RNA folding: An amnesic dynamic programming approach. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 295–306. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.35.

50 Rajesh Santhanam and Ross Williams. On medium-uniformity and circuit lower bounds. In *Computational Complexity (CCC), 2013 IEEE Conference on*, pages 15–23. IEEE, 2013.

51 Leslie G Valiant. *Graph-theoretic arguments in low-level complexity*. Springer, 1977.

52 R. Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2–3):357–365, 2005.

53 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.