58th Annual IEEE Symposium on Foundations of Computer Science

Distributed PCP Theorems for Hardness of Approximation in P

(Extended Abstract)

Amir Abboud
Stanford University
Computer Science Department
Palo Alto, CA, USA
abboud@cs.stanford.edu

Aviad Rubinstein
UC Berkeley
EECS
Berkeley, CA, USA
aviad@eecs.berkeley.edu

Ryan Williams
MIT
CSAIL and EECS
Cambridge, MA, USA
rrw@mit.edu

***Abstract*—We present a new *distributed* model of probabilistically checkable proofs (PCP). A satisfying assignment $x \in \{0,1\}^n$ to a CNF formula ϕ is shared between two parties, where Alice knows $x_1, \dots, x_{n/2}$, Bob knows $x_{n/2+1}, \dots, x_n$, and both parties know ϕ. The goal is to have Alice and Bob jointly write a PCP that x satisfies ϕ, while exchanging little or no information. Unfortunately, this model as-is does not allow for nontrivial query complexity. Instead, we focus on a *non-deterministic* variant, where the players are helped by Merlin, a third party who knows all of x.**

Using our framework, we obtain, for the first time, PCP-like reductions from the Strong Exponential Time Hypothesis (SETH) to approximation problems in P. In particular, under SETH we show that there are no truly-subquadratic approximation algorithms for Maximum Inner Product over $\{0,1\}$-vectors, LCS Closest Pair over permutations, Approximate Partial Match, Approximate Regular Expression Matching, and Diameter in Product Metric. All our inapproximability factors are nearly-tight. In particular, for the first three problems we obtain nearly-polynomial factors of $2^{(\log n)^{1-o(1)}}$; only $(1+o(1))$-factor lower bounds (under SETH) were known before.

As an additional feature of our reduction, we obtain new SETH lower bounds for the *exact* "monochromatic" Closest Pair problem in the Euclidean, Manhattan, and Hamming metrics.

***Index Terms*—fine-grained complexity; similarity search; strong exponential-time hypothesis; closest pair; longest common subsequence; inapproximability**

I. Introduction

Fine-Grained Complexity classifies the time complexity of fundamental problems under popular conjectures, the most productive of which has been the Strong Exponential Time Hypothesis[1] (SETH). The list of "SETH-Hard" problems is long, including central problems in pattern matching and bioinformatics [1], [2], [3], graph algorithms [4], [5], dynamic data structures [6], parameterized complexity and exact algorithms [7], [8], [9], computational geometry [10], time-series analysis [11], [12], and even economics [13] (a longer list can be found in [14]).

For most problems in the above references, there are natural and meaningful approximate versions, and for most of them the time complexity is wide open (a notable exception is [4]). Perhaps the most important and challenging open question in the field of Fine-Grained Complexity is whether a framework for *hardness of approximation in* P is possible. To appreciate the gaps in our knowledge regarding inapproximability, consider the following fundamental problem from the realms of similarity search and statistics, of finding the most *correlated* pair in a dataset.

Definition I.1 (The Max Inner Product Problem (Max-IP)). Given a set of N binary vectors in $\{0,1\}^d$, return a pair that maximizes the inner product.

Thinking of the vectors as subsets of $[d]$, this Max-IP problem asks to find the pair with largest overlap, a natural similarity measure. A naïve algorithm solves the problem in $O(N^2 d)$ time, and one of the most-cited fine-grained results is a SETH lower bound for this problem.[2] Assuming SETH, we cannot solve Max-IP (exactly) in $N^{2-\varepsilon} \cdot 2^{o(d)}$ time, for any $\varepsilon > 0$ [15].

This lower bound is hardly pleasing when one

[1] SETH is a pessimistic version of P$\neq$ NP, stating that for every $\varepsilon > 0$ there is a k such that k-SAT cannot be solved in $O((2-\varepsilon)^n)$ time.

[2] As a matter of fact, the lower bound is only for the bichromatic version of the problem, where we are given two sets of vector and want to find the best pair, one from each list. This distinction between monochromatic and bichromatic not so important for now, and we will only address it in Section I-B1.

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.12

of the most vibrant areas of Algorithms[3] is concerned with designing *approximate* but *near-linear* time solutions for such similarity search problems. For example, the original motivation of the celebrated MinHash algorithm was to solve the indexing version of this problem [16], [17], and one of the first implementations was at the core of the AltaVista search engine. The problem has important applications all across Computer Science, most notably in Machine Learning, databases, and information retrieval, e.g. [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34].

MAX-IP seems to be more challenging than closely related problems where similarity is defined as small Euclidean distance rather than large inner product. For the latter, we can get near-linear $O(N^{1+\varepsilon})$ time algorithms, for all $\varepsilon > 0$, at the cost of some constant $f(\varepsilon)$ error that depends on ε [18], [19], [23], [25]. In contrast, for MAX-IP, even for a moderately subquadratic running time of $O(N^{2-\varepsilon})$, all known algorithms suffer from *polynomial* $N^{g(\varepsilon)}$ approximation factors.

Meanwhile, the SETH lower bound for MAX-IP was only slightly improved by Ahle, Pagh, Razenshteyn, and Silvestri [31] to rule out $1 + o(1)$ approximations, leaving a huge gap between the not-even-1.001 lower bound and the polynomial upper bound.

Open Question 1. *Is there an $O(N^{1+\varepsilon})$-time algorithm for computing an $f(\varepsilon)$-approximation to* MAX INNER PRODUCT *over binary vectors?*

This is just one of the many significant open questions that highlight our inability to prove hardness of approximation in P, and pour cold water on the excitement from the successes of Fine-Grained Complexity. It is natural to try to adapt tools from the NP-Hardness-of-approximation framework (namely, the celebrated *PCP Theorem*) to P. Unfortunately, when starting from SETH, almost everything in the existing theory of PCPs breaks down. Whether PCP-like theorems for Fine-Grained Complexity are possible, and what they could look like, are fascinating open questions.

Our main result is the first SETH-based PCP-like theorem, from which several strong hardness of approximation in P results follow. We identify a canonical problem that is hard to approximate, and further gadget-reductions allow us to prove SETH-based inapproximability results for basic problems such as Subset Queries, Closest Pair under the Longest Common Subsequence similarity measure, and Furthest Pair (Diameter) in product metrics. In particular, assuming SETH, we negatively resolve Open Question 1 in a very strong way, proving an almost tight lower bound for MAX-IP.

A. PCP-like Theorems for Fine-Grained Complexity

The following meta-structure is common to most SETH-based reductions: given a CNF φ, construct $N = O\left(2^{\frac{n}{2}}\right)$ gadgets, one for each assignment to the first/last $n/2$ variables, and embed those gadgets into some problem A. The embedding is designed so that if A can be solved in $O\left(N^{2-\varepsilon}\right) = O\left(2^{\left(1-\frac{\varepsilon}{2}\right)n}\right)$ time, a satisfying assignment for φ can be efficiently recovered from the solution, contradicting SETH.

The most obvious barrier to proving fine-grained hardness of approximation is the lack of an appropriate PCP theorem. Given a 3-SAT formula φ, testing that an assignment $x \in \{0,1\}^n$ satisfies φ requires reading all n bits of x. The PCP Theorem [35], [36], shows how to transform $x \in \{0,1\}^n$ into a PCP (*probabilistically checkable proof*) $\pi = \pi(\varphi, x)$, which can be tested by a probabilistic verifier who only reads a few bits from π. This is the starting point for almost all proofs of NP-hardness of approximation. The main obstacle in using PCPs for fine-grained hardness of approximation is that all known PCPs incur a blowup in the size proof: $\pi(\varphi, x)$ requires $n' \gg n$ bits. The most efficient known PCP, due to Dinur [37], incurs a polylogarithmic blowup ($n' = n \cdot \mathsf{polylog}(n)$), and obtaining a PCP with a constant blowup is a major open problem (e.g. [38], [39]). However, note that even if we had a fantastic PCP with only $n' = 10n$, a reduction of size $N' = 2^{\frac{n'}{2}} = 2^{5n}$ does not imply any hardness at all. Our goal is to overcome this barrier:

Open Question 2. *Is there a PCP-like theorem for fine-grained complexity?*

Distributed PCPs

Our starting point is that of error-correcting codes, a fundamental building block of PCPs. Suppose that Alice and Bob want to encode a message $m = (\alpha; \beta) \in \{0,1\}^n$ in a distributed fashion. Neither Alice nor Bob knows the entire message: Alice knows the first half ($\alpha \in \{0,1\}^{\frac{n}{2}}$), and Bob knows the second half ($\beta \in \{0,1\}^{\frac{n}{2}}$). Alice can locally compute an encoding $E'(\alpha)$ of her half, and Bob locally computes an encoding $E'(\beta)$ of his. Then the concatenation of the Alice's and Bob's strings, $E(m) = (E'(\alpha); E'(\beta))$, is an error-correcting encoding of m.

[3] In SODA'17, two entire sessions were dedicated to algorithms for similarity search.

Now let us return to *distributed PCPs*. Alice and Bob share a k-SAT[4] formula φ. Alice has an assignment $\alpha \in \{0,1\}^{\frac{n}{2}}$ to the first half of the variables, and Bob has an assignment $\beta \in \{0,1\}^{\frac{n}{2}}$ to the second half. We want a protocol where Alice locally computes a string $\pi'(\alpha) \in \{0,1\}^{n'}$, Bob locally computes $\pi'(\beta) \in \{0,1\}^{n'}$, and together $\pi(\alpha;\beta) = (\pi'(\alpha), \pi'(\beta))$ is a valid probabilistically checkable proof that $x = (\alpha, \beta)$ satisfies φ. That is, a probabilistic verifier can read a constant number of bits from $(\pi'(\alpha), \pi'(\beta))$ and decide (with success probability at least $2/3$) whether (α, β) satisfies φ.

It is significant to note that **if distributed PCPs can be constructed, then very strong reductions for fine-grained hardness of approximation follow**, completely overcoming the barrier for fine-grained PCPs outlined above. The reason is that we can still construct $N = O\left(2^{\frac{n}{2}}\right)$ gadgets, one for each half assignment $\alpha, \beta \in \{0,1\}^{\frac{n}{2}}$, where the gadget for α also encodes $\pi'(\alpha)$. The blowup of the PCP only affects the size of each gadget, which is negligible compared to the number of gadgets. In fact, this technique would be so powerful, that we could reduce SETH to problems like approximate ℓ_2-Nearest Neighbor, where the existing sub-quadratic approximation algorithms (e.g. [25]) would falsify SETH!

Alas, distributed PCPs are *unconditionally impossible* (even for 2-SAT) by a simple reduction from Set Disjointness:

Theorem I.2 (Reingold [40]; informal)**.** *Distributed PCPs are impossible.*

Proof (sketch). Consider the 2-SAT formula $\varphi \triangleq \bigwedge_{i=1}^{n/2} (\neg\alpha_i \vee \neg\beta_i)$. This φ is satisfied by assignment $(\alpha;\beta)$ iff the vectors $\alpha, \beta \in \{0,1\}^{\frac{n}{2}}$ are disjoint. If a PCP verifier can decide whether $(\alpha;\beta)$ satisfies φ by a constant number of queries to $(\pi'(\alpha), \pi'(\beta))$, then Alice and Bob can simulate the PCP verifier to decide whether their vectors are disjoint, while communicating only a constant number of bits (the values read by the PCP verifier). This contradicts the randomized communication complexity lower bounds of $\Omega(n)$ for set disjointness [41], [42], [43]. □

Note that the proof shows that even distributed PCPs with $o(n)$ queries are impossible.

Distributed and non-deterministic PCPs: As noted above, set disjointness is very hard for randomized communication, and hard even for non-deterministic communication [44]. But Aaronson and Wigderson [45] showed that set disjointness does have $\tilde{O}(\sqrt{n})$ Merlin-Arthur (MA) communication complexity. In particular, they construct a simple protocol where the standard Bob and an untrusted Merlin (who can see both sets of Alice and Bob) each send Alice a message of length $\tilde{O}(\sqrt{n})$. If the sets are disjoint, Merlin can convince Alice to accept; if they are not, Alice will reject with high probability regardless of Merlin's message.

Our second main insight in this paper is this: for problems where the reduction from SETH allows for an efficient OR gadget, we can enumerate over all possible messages from Merlin and Bob[5]. Thus we incur only a subexponential blowup[6] in the reduction size,while overcoming the communication barrier. Indeed, the construction in our PCP-like theorem can be interpreted as implementing a variant of Aaronson and Wigderson's MA communication protocol. The resulting PCP construction is *distributed* (in the sense described above) and *non-deterministic* (in the sense that Alice receives sublinear advice from Merlin).

It can be instructive to view our distributed PCP model as a 4-party (computationally-efficient) communication problem. Merlin wants to convince Alice, Bob, and Veronica (the verifier) that Alice and Bob jointly hold a satisfying assignment to a publicly-known formula. Merlin sees everything except the outcome of random coin tosses, but he can only send $o(n)$ bits to only Alice. Alice and Bob each know half of the (allegedly) satisfying assignment, and each of them must (deterministically) send a (possibly longer) message to Veronica. Finally, Veronica tosses coins and is restricted to reading only $o(n)$ bits from Alice's and Bob's messages, after which she must output Accept/Reject.

Patrascu and Williams [7] asked whether it is possible to use Aaronson and Wigderson's MA protocol for Set Disjointness to obtain better algorithms for satisfiability. Taking an optimistic twist, our results in this paper may suggest this is indeed possible: if any of several important and simple problems admit efficient approximation algorithms, then faster algorithms for (exact) satisfiability may be obtained via Aaronson and Wigderson's MA protocol.

B. *Our results*

Our distributed and non-deterministic PCP theorem is formalized and proved in the full version. Since our

[4] In the formulation of SETH, k is a "sufficiently large constant". However, for the purposes of our discussion here it suffices to think of $k = 3$.

[5] In fact, enumerating over Merlin's possible messages turns out to be easy to implement in the reductions; the main bottleneck is the communication with Bob.

[6] Subexponential in n (the number of k-SAT variables), which implies sub*polynomial* in $N \approx 2^{n/2}$.

main interest is proving hardness-of-approximation results, we abstract the prover-verifier formulation by reducing our PCP to an Orthogonal-Vectors-like problem which we call PCP-VECTORS (see below). PCP-VECTORS turns out to be an excellent starting point for many results, yielding easy reductions for fundamental problems and giving essentially tight inapproximability bounds. We begin with the description of PCP-VECTORS, and then exhibit what we think are the most interesting applications.

a) PCP-Vectors: We introduce an intermediate problem which we call PCP VECTORS. The purpose of introducing this problem is to abstract out the prover-verifier formulation before proving hardness of approximation in P, very much like NP-hardness of approximation reductions start from gap-3-SAT or LABEL COVER.

Definition I.3 (PCP-VECTORS)**.** The input to this problem consists of two sets of vectors $A \subset \Sigma^{L \times K}$ and $B \subset \Sigma^{L}$, The goal is to find vectors $a \in A$ and $b \in B$ that maximize

$$s\left(a,b\right) \triangleq \Pr_{\ell \in L}\left[\bigvee_{k \in K}\left(a_{\ell,k} = b_{\ell}\right)\right]. \qquad (1)$$

Theorem I.4. *Let $\varepsilon > 0$ be any constant, and let (A, B) be an instance of* PCP-VECTORS *with N vectors and parameters $|L|, |K|, |\Sigma| = N^{o(1)}$.Then, assuming SETH, $O\left(N^{2-\varepsilon}\right)$-time algorithms cannot distinguish between:*

- *(Completeness) there exist a^*, b^* such that $s\left(a^*, b^*\right) = 1$; and*
- *(Soundness) for every $a \in A, b \in B$, we have $s\left(a,b\right) \leq 1/2^{(\log N)^{1-o(1)}}$.*

We also have a symmetric variant of PCP-VECTORS (which we call SYMMETRIC PCP-VECTORS), where the vectors come from one set. There is some tradeoff between the properties of the two variants: In PCP-VECTORS, we can afford to assume additional structure on the hard instances, which supports reductions to structured problems like SUBSET QUERY and REGULAR EXPRESSION. In contrast, having one set of vectors in SYMMETRIC PCP-VECTORS simplifies reductions to Closest Pair problems with one set, like MAX IP and LCS CLOSEST PAIR.

Definition I.5 (SYMMETRIC PCP-VECTORS)**.** The input to this problem consists of a single set of vectors $U \subset \Sigma^{L \times K}$.The goal is to find a pair of vectors $u^*, v^* \in U$ (where $u^* \neq v^*$) that maximize

$$s\left(u,v\right) \triangleq \Pr_{\ell \in L}\left[\bigvee_{k \in K}\left(u_{\ell,k} = v_{\ell,k}\right)\right]. \qquad (2)$$

Theorem I.6. *Let $\varepsilon > 0$ be any constant, and let U be an instance of* SYMMETRIC PCP-VECTORS *with N vectors and parameters $|L|, |K|, |\Sigma| = N^{o(1)}$. Then, assuming SETH, $O\left(N^{2-\varepsilon}\right)$-time algorithms cannot distinguish between:*

- *(Completeness) there exist $u^* \neq v^* \in U$ such that $s\left(u^*, v^*\right) = 1$; and*
- *(Soundness) for every $u, v \in U$ we have $s\left(u,v\right) \leq 1/2^{(\log N)^{1-o(1)}}$.*

Furthermore, we have the guarantee that for every $\ell \in L$, there is at most one $k \in K$ such that $u_{\ell,k} = v_{\ell,k}$[7]*.*

b) Max Inner Product: Our first application is a strong resolution of Open Question 1, under SETH. Not only is an $O(1)$-factor approximation impossible in $O(N^{1+\varepsilon})$ time, but we must pay a near-polynomial $2^{(\log N)^{1-o(1)}}$ approximation factor if we do not spend nearly-quadratic $N^{2-o(1)}$ time!

Theorem I.7. *Assuming SETH, for all $\varepsilon > 0$, every $O(N^{2-\varepsilon})$ time algorithm for* MAX INNER PRODUCT *on N vectors from $\{0,1\}^d$ with dimension $d = N^{o(1)}$ must have approximation factor at least $2^{(\log N)^{1-o(1)}}$.*

Improving our lower bound even to some N^{ε} factor would refute SETH via the known MAX-IP algorithms (see e.g. [31]). Using a standard trick, Theorem I.7 also applies to the harder (but more useful) search version widely known as MIPS.

Corollary I.8. *Assuming SETH, for all $\varepsilon > 0$, no algorithm can preprocess a set of N vectors $p_1, \ldots, p_N \in D \subseteq \{0,1\}^d$ in polynomial time, and subsequently given a query vector $q \in \{0,1\}^d$ can distinguish in $O(N^{1-\varepsilon})$ time between the cases:*

- *(Completeness) there is a $p_i \in D$ such that $\langle p_i, q \rangle \geq s$; and*
- *(Soundness) for all $p_i \in D$, $\langle p_i, q \rangle \leq s/2^{(\log N)^{1-o(1)}}$,*

even when $d = N^{o(1)}$ and the similarity threshold $s \in [d]$ is fixed for all queries q.

Except for the $(1 + o(1))$-factor lower bound [31] which transfers to MIPS as well, the only lower bounds known were either for specific techniques [46], [47], [48], [24], or were in the cell-probe model but only ruled out extremely efficient queries [49], [50], [51], [52], [53], [54].

An important version of MAX-IP is when the vectors are in $\{-1,1\}^d$ rather than $\{0,1\}^d$. This

[7] Notice that for the asymmetric variant PCP-VECTORS, this is true without loss of generalitysince all $a_{\ell,k}$ are compared to the same b_{ℓ}.

version is closely related to other famous problems such as the light bulb problem and the problem of learning parity with noise (see the reductions in [28]). Negative coordinates often imply trivial results for *multiplicative* hardness of approximation: it is possible to shift a tiny gap of k vs. $k+1$ to a large multiplicative gap of 0 vs 1 by adding k coordinates with -1 contribution. In the natural version where we search for a pair with maximum inner product *in absolute value*, this trick does not quite work. Still, Ahle et al. [31] exploit such cancellations to get a strong hardness of approximation result using an interesting application of Chebychev embeddings. The authors had expected that a different approach must be taken to prove constant factor hardness for the $\{0,1\}$ case. Interestingly, since it is easy to reduce $\{0,1\}$ to $\{-1,1\}$[8], our reduction also improves their lower bound for the $\{-1,1\}$ case from $2^{\tilde{\Omega}(\sqrt{\log N})}$ to the almost-tight $2^{(\log N)^{1-o(1)}}$. This also implies an $N^{1-o(1)}$-time lower bound for queries in the indexing version of the problem.

c) Subset Queries: A seemingly easier special case of MAX-IP which has received extensive attention is the Subset Query problem [55], [56], [57], [58] which is known to be equivalent to the classical Partial Match problem, for which the first non-trivial algorithms appeared in Ronald Rivest's PhD thesis [59], [60]. Since our goal is to prove lower bounds, we consider its offline or batch version (and the lower bound will transfer to the data structure version):

> Given a collection of (text) sets $T_1, \ldots, T_N \subseteq [d]$ and a collection of (pattern) sets $P_1, \ldots, P_N \subseteq [d]$, is there a set P_i that is contained in a set T_j?

In the c-approximate case, we want to distinguish between the case of exact containment, and the case where no T_j can cover more than a c-fraction of any P_i. We prove that even this very simple problem must pay a $2^{(\log N)^{1-o(1)}}$ approximation factor if it is to be solved in truly-subquadratic time. Again, the only previous lower bound factor was $(1+o(1))$, which follows from [31].

Theorem I.9. *Assuming SETH, for any $\varepsilon > 0$, given two sets $\mathcal{D}, \mathcal{P}$ of N subsets of a universe $[d]$ where $d = N^{o(1)}$ and all sets $P \in \mathcal{P}$ have size k, no $O(N^{2-\varepsilon})$ time algorithm can distinguish between the cases:*

- *(Completeness) there are $P \in \mathcal{P}, D \in \mathcal{D}$ such that $P \subseteq D$; and*
- *(Soundness) for all $P \in \mathcal{P}, D \in \mathcal{D}$ we have $|D \cap P| \leq k/2^{(\log N)^{1-o(1)}}$.*

d) Longest Common Subsequence Closest Pair: Efficient approximation algorithms have the potential for major impact in *sequence alignment problems*, the standard similarity measure between genomes and biological data. One of the most cited scientific papers of all time studies BLAST, a *heuristic* algorithm for sequence alignment that often returns grossly sub-optimal solutions[9] but always runs in near-linear time, in contrast to the best-known worst-case quadratic-time algorithms. For theoreticians, to get the most insight into these similarity measures, it is common to think of them as Longest Common Subsequence (LCS) or Edit Distance. The LCS CLOSEST PAIR problem is:

> Given a (data) set of N strings and a (query) set of N strings, all of which have length $m \ll N$, find a pair, one from each set, that have the maximum length common subsequence (noncontiguous).

The search version and the Edit Distance version are defined analogously. Good algorithms for these problems would be highly relevant for bioinformatics.

The known gaps between upper and lower bounds are huge. A series of breakthroughs [61], [62], [63], [64], [65], [66] led to "good" approximation algorithms for Edit Distance: the closest pair version can be solved in near-linear time with a $2^{O(\sqrt{\log m \log\log m})}$ approximation. Meanwhile, LCS resisted all these attacks, and to our knowledge, no non-trivial algorithms are known. On the complexity side, only a $(1+o(1))$-approximation factor lower bound is known for LCS [11], [12], [67], and getting a 1.001 approximation in near-linear time is not known to have any consequences. For certain algorithmic techniques like metric embeddings there are nearly logarithmic lower bounds for Edit-Distance, but even under such restrictions the gaps are large [68], [69], [70], [71], [72].

Perhaps our most surprising result is a *separation* between these two classical similarity measures. Although there is no formal equivalence between the two, they have appeared to have the same complexity no matter what the model and setting are. We prove that LCS Closest Pair is *much harder* to approximate than Edit Distance.

Theorem I.10. *Assuming SETH, there is no $\left(2^{(\log N)^{1-o(1)}}\right)$-approximation algorithm for* LCS

[8] E.g. map each 0 to a random string in $\{\pm 1\}^d$, and map each 1 to the string 1^d.

[9] Note that many of its sixty-thousand citations are by other algorithms achieving better results (on certain datasets).

CLOSEST PAIR *on N permutations of length $m = N^{o(1)}$ in time $O\left(N^{2-\varepsilon}\right)$, for all $\varepsilon > 0$.*

Notice that our theorem holds even under the restriction that the sequences are *permutations*. This is significant: in the "global" version of the problem where we want to compute the LCS between two long strings of length n, one can get the exact solution in near-linear time if the strings are permutations (the problem becomes the famous Longest Increasing Subsequence), while on arbitrary strings there is an $N^{2-o(1)}$ time lower bound from SETH. The special case of permutations has received considerable attention due to connections to preference lists, applications to biology, and also as a test-case for various techniques. In 2001, Cormode, Muthukrishnan, and Sahinalp [73] gave an $O(\log d)$-approximate nearest neighbor data structure for Edit Distance on permutations with $N^{o(1)}$ time queries (improved to $O(\log \log m)$ in [74]), and raised the question of getting similar results for LCS. Our result gives a strong negative answer under SETH, showing that LCS CLOSEST PAIR suffers from near-polynomial approximation factors when the query time is truly sublinear.

e) Regular Expression Matching: Given two sets of strings of length m, a simple hashing-based approach lets us decide in near-linear time if there is a pair of Hamming distance 0 (equal strings), or whether all pairs have distance at least 1. A harder version of this problem, which appears in many important applications, is when one of the sets of strings is described by a regular expression:

> Given a regular expression R of size N and a set S of N strings of length m, can we distinguish between the case that some string in S is matched by R, and the case that every string in S is far in Hamming distance[10] from every string in $\mathrm{L}(R)$ (the language defined by R)?

This is a basic approximate version of the classical regular expression matching problem that has been attacked from various angles throughout five decades, e.g. [75], [76], [77], [78], [79], [80], [81], [82], [83], [3], [84]. Surprisingly, we show that this problem is essentially as hard as it gets: even if there is an exact match, it is hard to find any pair with Hamming distance $(1-\varepsilon)\cdot m$, for any $\varepsilon > 0$. For the case of binary alphabets, we show that even if an exact match exists (a pair of distance 0), it is hard to find a pair of distance $(\frac{1}{2}-\varepsilon)\cdot m$, for any $\varepsilon > 0$. Our lower bounds also rule out interesting algorithms for the harder setting of Nearest-Neighbor queries: Preprocess a regular expression so that given a string, we can find a string in the language of the expression that is approximately-the-closest one to our query string. The formal statement and definitions of regular expressions are given in the full version.

Theorem I.11 (informal)**.** *Assuming SETH, no $O(N^{2-\varepsilon})$-time algorithm can, given a regular expression R of size N and a set S of N strings of length $m = N^{o(1)}$, distinguish between the two cases:*

- *(Completeness) some string in S is in $\mathrm{L}(R)$*
- *(Soundness) all strings in S have Hamming distance $(1-o(1))\cdot m$ (or, $(\frac{1}{2}-o(1))\cdot m$ if the alphabet is binary) from all strings in $\mathrm{L}(R)$.*

f) Diameter in Product Metrics: The *diameter* (or *furthest pair*) problem has been well-studied in a variety of metrics (e.g. graph metrics [85], [4], [86]). There is a trivial 2-approximation in near-linear time (return the largest distance from an arbitrary point), and for arbitrary metrics (to which we get query access) there is a lower bound stating that a quadratic number of queries is required to get a $(2-\delta)$-approximation [87]. For ℓ_2-metric, there is a sequence of improved subquadratic-time approximation algorithms [88], [89], [90], [91], [92], [93]. The natural generalization to the ℓ_p-metric for arbitrary p is, to the best of our knowledge, wide open.

While we come short of resolving the complexity of approximating the diameter for ℓ_p-metrics, we prove a tight inapproximability result for the slightly more general problem for the product of ℓ_p metrics. Given a collection of metric spaces $M_i = \langle X_i, \Delta_i \rangle$, their *$f$-product metric* is defined as

$$\Delta\Big((x_1,\ldots,x_k),(y_1,\ldots,y_k)\Big)$$
$$\triangleq f\Big(\Delta_1(x_1,y_1),\ldots,\Delta_k(x_k,y_k)\Big).$$

In particular, we are concerned with the ℓ_2-product of ℓ_∞-spaces, whose metric is defined as:

$$\Delta_{2,\infty}(x,y) \triangleq \sqrt{\sum_{i=1}^{d_2}\left(\max_{j=2}^{d_\infty}\left\{\Big|x_{i,j}-y_{i,j}\Big|\right\}\right)^2}. \quad (3)$$

(This is a special case of the more general $\Delta_{2,\infty,1}(\cdot,\cdot)$ product metric, studied by [74].)

Product metrics (or *cascaded norms*) are useful for aggregating different types of data [94], [95], [96], [97]. They also received significant attention from the algorithms community because they allow rich embeddings, yet are amenable to algorithmic techniques (e.g. [95], [93], [98], [74], [72], [66]).

[10] In our hard instances, all the strings in $\mathrm{L}(R)$ will be of length m, so Hamming distance is well defined.

Theorem I.12 (Diameter)**.** *Assuming SETH, there are no $(2-\delta)$-approximation algorithms for* Product-Metric Diameter *in time $O\left(N^{2-\varepsilon}\right)$, for any constants $\varepsilon, \delta > 0$.*

1) Closest Pair vs. "Bichromatic" Closest Pair: The main results in this paper extend known hardness-in-P results to also rule out efficient approximation algorithms. An additional feature of our reduction is that it does not suffer from the following caveat, common to almost all previous work. Going back to the Max-IP problem, for example, the known hardness results of [15], [31] hold only for the "bichromatic" variant of the problem: given sets of vectors A, B, the algorithm must find a pair $a \in A$ and $b \in B$ that maximizes $a \cdot b$. In contrast, our results hold both for the bichromatic variant and the "monochromatic" variant, where given a single set U, one must find a pair $x, y \in U$ (s.t. $x \neq y$) that maximizes $x \cdot y$.[11] For the latter variant, even in the *exact* setting (no approximation allowed), it was open whether there is a SETH-based lower bound.

As a corollary, we obtain via known reductions[12] exact hardness for monochromatic variants of other problems. For example,

Corollary I.13 (Monochromatic Euclidean and Hamming Closest Pair)**.** *Assuming SETH, there are no $O\left(N^{2-\varepsilon}\right)$-time exact algorithms for the (monochromatic)* Closest Pair *problem, with Euclidean, Manhattan, or Hamming metrics, for any constant $\varepsilon > 0$.*

Note that in low dimensions, the monochromatic version of Euclidean Closest Pair is known to admit polynomially faster algorithms than the bichromatic version [99]. Furthermore, [100] recently showed that even in higher dimensions, hardness for the monochromatic Euclidean Closest Pair cannot be proven by reducing the bichromatic to the monochromatic.

C. Related work

For all the problems we consider, SETH lower bounds for the exact (bichromatic) version are known. See [15], [29] for the Max-IP and Subset Queries problems, [1], [11], [12], [101] for LCS Closest Pair, [3], [84] for Regular Expression Matching, and [15] for Metric Diameter.

Prior to our work, some hardness of approximation results were known using more problem-specific techniques. For example, distinguishing whether the diameter of a graph on $O(n)$ edges is 2 or at least 3 in truly-subquadratic time refutes SETH [4], which implies hardness for $(3/2-\varepsilon)$ approximations. (This is somewhat analogous to the NP-hardness of distinguishing 3-colorable graphs from graphs requiring at least 4 colors, immediately giving hardness of approximation for the chromatic number.) In most cases, however, this fortunate situation does not occur. The only prior SETH-based hardness of approximation results proved with more approximation-oriented techniques are by Ahle et al. [31] for Max-IP via clever embeddings of the vectors. As discussed above, for the case of $\{0,1\}$-valued vectors, their inapproximability factor is still only $1+o(1)$.

[67] show that, under certain complexity assumptions, *deterministic* algorithms cannot approximate the Longest Common Subsequence (LCS) of two strings to within $1+o(1)$ in truly-subquadratic time. They tackle a completely orthogonal obstacle to proving SETH-based hardness of approximation: for problems like LCS with two long strings, the quality of approximation depends on the *fraction of assignments* that satisfy a SAT instance. There is a trivial algorithm for approximating this fraction: sample assignments uniformly at random. See further discussion on Open Question 4.

Recent works by Williams [102] (refuting the MA-variant of SETH) and Ball et al. [103] also utilize low-degree polynomials in the context of SETH and related conjectures. Their polynomials are quite different from ours: they sum over many possible assignments, and are hard to *evaluate* (in contrast, the polynomials used in our proof correspond to a single assignment, and they are trivial to evaluate).

The main technical barrier to hardness of approximation in P is the blowup incurred by standard PCP constructions; in particular, we overcome it with distributed constructions. There is also a known construction of PCP with linear blowup for large (but sublinear) query complexity [38] with non-uniform verifiers; note however that merely obtaining linear blowup is not small enough for our purposes. Different models of "non-traditional" PCPs, such as interactive PCPs [104] and interactive oracle proofs (IOP) [105], [106] have been considered and found "positive" applications in cryptography (e.g. [107], [108], [105]). In particular, [109] obtain a linear-size IOP. It is an open question whether these interactive variants can imply interesting hardness of approximation results [109]. (And it would be very interesting if our distributed PCPs have any cryptographic appli-

[11] In fact, our results hold for an even stronger variant: given two sets A, B, we show that it is hard to distinguish between the case where there is a pair $a \in A$ and $b \in B$ with a large inner product, and the case where every pair $x \neq y \in (A \cup B)$ has a small inner product.

[12] In fact, via the "trivial" reduction that uses the exact same instance.

cations!)

After the first version of this paper became public, it was brought to our attention that the term ”distributed PCP” has been used before in a different context by Drucker [110]. In the simplest variant of Drucker’s model, Alice and Bob want to compute $f(\alpha, \beta)$ with minimal communication. They receive a PCP that allegedly proves that $f(\alpha, \beta) = 1$; Alice and Bob each query the PCP at two random locations and independently decide whether to accept or reject the PCP. As with the interactive variants of PCP, we don’t know of any implications of Drucker’s work for hardness of approximation, but we think that this is a fascinating research direction.

D. Discussion

In addition to resolving the fine-grained approximation complexity of several fundamental problems, our work opens a hope to understanding more basic questions in this area. We list a few that seem to represent some of the most fundamental challenges, as well as exciting applications.

a) LCS CLOSEST PAIR PROBLEM *over* $\{0, 1\}$*:* The LCS CLOSEST PAIR PROBLEM is most interesting in two regimes: permutations (which, by definition, require a large alphabet); and small alphabet, most notably $\{0, 1\}$. For the regime of permutations, we obtain nearly-polynomial hardness of approximation. For small alphabet Σ, per contra, there is a trivial $1/|\Sigma|$-approximation algorithm in near-linear time: pick a random $\sigma \in \Sigma$, and restrict all strings to their σ-subset. Are there better approximation algorithms?

Our current hardness techniques are limited because this problem does not admit an approximation preserving OR-gadget for a large OR. In particular the $1/|\Sigma|$-approximation algorithm outlined above implies that we cannot combine much more than $|\Sigma|$ substrings in a clever way and expect the LCS to correspond to just one substring.

Open Question 3. *Is there a* 1.1*-approximation for the* LCS CLOSEST PAIR PROBLEM *on binary inputs running in* $O(n^{2-\varepsilon})$ *time, for some* $\varepsilon > 0$*?*

b) LCS PROBLEM *(with two strings):* Gadgets constructed in a fashion similar to our proof of Theorem I.10 can be combined together (along with some additional gadgets) into two long strings A, B of length m, in a way that yields a reduction from SETH to computing the longest common subsequence (LCS) of (A, B), ruling out *exact* algorithms in $O(m^{2-\varepsilon})$ [11], [12]. However, in the instances output by this reduction, approximating the value of the LCS reduces to approximating the *fraction* of assignments that satisfy the original formula; it is easy to obtain a good additive approximation by sampling random assignments. The recent work of [67] mentioned above, uses complexity assumptions on deterministic algorithms to tackle this issue, but their ideas do not seem to generalize to randomized algorithms.

Open Question 4. *Is there a* 1.1*-approximation for* LCS *running in* $O(n^{2-\varepsilon})$ *time, for some* $\varepsilon > 0$*? (Open for all alphabet sizes.)*

c) Dynamic Maximum Matching: A holy grail in dynamic graph algorithms is to maintain a $(1 + \varepsilon)$-approximation for the *Maximum Matching* in a dynamically changing graph, while only spending amortized $n^{o(1)}$ time on each update. Despite a lot of attention in the past few years [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], current algorithms are far from achieving this goal: one can obtain a $(1 + \varepsilon)$-approximation by spending $\Omega(\sqrt{m})$ time per update, or one can get an 2-approximation with $\tilde{O}(1)$ time updates.

For exact algorithms, we know that $n^{o(1)}$ update times are impossible under popular conjectures [121], [6], [122], [123], [124], such as 3-SUM[13], Triangle Detection[14] and the related Online Matrix Vector Multiplication[15]. From the viewpoint of PCP’s, this question is particularly intriguing since it seems to require hardness amplification for one of these other conjectures. Unlike all the previously mentioned problems, even the exact case of dynamic matching is not known to be SETH-hard.

Open Question 5. *Can one maintain an* $(1 + \varepsilon)$*-approximate maximum matching dynamically, with* $n^{o(1)}$ *amortized update time?*

New frameworks for hardness of approximation: More fundamental than resolving any particular problem, our main contribution is a conceptually new framework for proving hardness of approximation for problems in P via *distributed PCPs*. In particular, we were able to resolve several open problems while relying on simple algebrization techniques from early days of PCPs (e.g. [127] and reference therein). It is plausible that our results can be improved by importing into our framework more advanced techniques

[13] The 3-SUM Conjecture, from the pioneering work of [125], states that we cannot find three numbers that sum to zero in a list of n integers in $O(n^{2-\varepsilon})$ time, for some $\varepsilon > 0$.

[14] The conjecture that no algorithm can find a triangle in a graph on m edges in $O(m^{4/3-\varepsilon})$ time, for some $\varepsilon > 0$, or even just that $O(m^{1+o(1)})$ algorithms are impossible [6].

[15] The conjecture that given a Boolean $n \times n$ matrix M and a sequence of n vectors $v_1, \ldots, v_n \in \{0, 1\}^n$ we cannot compute the n products $M \cdot x_i$ in an online fashion (output Mx_i before seeing x_{i+1}) in a total of $O(n^{3-\varepsilon})$ time [123]. See [126] for a recent upper bound.

from decades of work on PCPs — starting with verifier composition [35], parallel repetition [128], Fourier analysis [129], etc.

d) Hardness from other sublinear communication protocols for Set Disjointness: A key to our results is an MA protocol for Set Disjointness with sublinear communication, which trades off between the size of Merlin's message and the size of Alice and Bob's messages. There are other non-standard communication models where Set Disjointness enjoys a sublinear communication protocol, for example quantum communication[16] [130].

Open Question 6. *Can other communication models inspire new reductions (or algorithms) for standard computational complexity?*

e) Hardness of approximation from new models of PCPs: This is the most open-ended question. Formulating a clean conjecture about distributed PCPs was extremely useful for understanding the limitations and possibilities of our framework — even though our original conjecture turned out to be false.

Open Question 7. *Formulate a simple and plausible PCP-like conjecture that resolves any of the open questions mentioned in this section.*

Acknowledgements

We thank Karthik C.S., Alessandro Chiesa, Søren Dahlgaard, Piotr Indyk, Rasmus Pagh, Ilya Razenshteyn, Omer Reingold, Nick Spooner, Virginia Vassilevska Williams, Ameya Velingker, and anonymous reviewers for helpful discussions and suggestions.

This work was done in part at the Simons Institute for the Theory of Computing. We are also grateful to the organizers of Dagstuhl Seminar 16451 for a special collaboration opportunity.

A.A. was supported by the grants of Virginia Vassilevska Williams: NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338. A.R. was supported by Microsoft Research PhD Fellowship, NSF grant CCF-1408635 and by Templeton Foundation grant 3966. R.W. was supported by an NSF CAREER grant (CCF-1741615).

References

[1] A. Abboud, V. V. Williams, and O. Weimann, "Consequences of faster alignment of sequences," in *Proc. of the 41st ICALP*, 2014, pp. 39–51.

[2] A. Backurs and P. 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)*, 2015, pp. 51–58.

[3] ——, "Which regular expression patterns are hard to match?" in *Proc. of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2016, pp. 457–466.

[4] L. Roditty and V. Vassilevska Williams, "Fast approximation algorithms for the diameter and radius of sparse graphs," in *Proc. of the 45th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2013, pp. 515–524.

[5] J. Gao, R. Impagliazzo, A. Kolokolova, and R. R. Williams, "Completeness for first-order properties on sparse structures with algorithmic applications," in *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017, pp. 2162–2181.

[6] A. Abboud and V. Vassilevska Williams, "Popular conjectures imply strong lower bounds for dynamic problems," in *Proc. of the 55th FOCS*, 2014, pp. 434–443.

[7] M. Patrascu and R. Williams, "On the possibility of faster SAT algorithms," in *Proc. of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2010, pp. 1065–1075. [Online]. Available: http://dx.doi.org/10.1137/1.9781611973075.86

[8] D. Lokshtanov, D. Marx, and S. Saurabh, "Lower bounds based on the exponential time hypothesis," *Bulletin of the EATCS*, vol. 105, pp. 41–72, 2011.

[9] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström, "On problems as hard as CNF-SAT," *ACM Transactions on Algorithms*, vol. 12, no. 3, p. 41, 2016.

[10] K. Bringmann, "Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails," in *Proc. of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014, pp. 661–670.

[11] A. Abboud, A. Backurs, and V. Vassilevska Williams, "Tight hardness results for LCS and other sequence similarity measures," in *Proc. of the 56th FOCS*, 2015, pp. 59–78.

[12] K. Bringmann and M. 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)*, 2015, pp. 79–97.

[13] D. Moeller, R. Paturi, and S. Schneider, "Subquadratic algorithms for succinct stable matching," in *International Computer Science Symposium in Russia*. Springer, 2016, pp. 294–308.

[14] V. V. Williams, "Hardness of easy problems: basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk)," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 43, 2015.

[15] R. R. Williams, "A new algorithm for optimal 2-constraint satisfaction and its implications," *Theoretical Computer Science*, vol. 348, no. 2–3, pp. 357–365, 2005.

[16] A. Z. Broder, "On the resemblance and containment of documents," in *Compression and Complexity of Sequences 1997. Proceedings*. IEEE, 1997, pp. 21–29.

[17] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," *Computer Networks and ISDN Systems*, vol. 29, no. 8-13, pp. 1157–1166, 1997.

[18] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proc. of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[19] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. of the 47th FOCS*. IEEE, 2006, pp. 459–468.

[20] A. Rahimi, B. Recht *et al.*, "Random features for large-scale kernel machines." in *NIPS*, vol. 3, no. 4, 2007, p. 5.

[21] P. Ram and A. G. Gray, "Maximum inner-product search using cone trees," in *Proc. of the 18th ACM SIGKDD*

[16] Note that due to Grover's algorithm, SETH is false for quantum computational complexity; but it is also false for MA [102], which doesn't prevent us from using the MA communication protocol in an interesting way.

international conference on Knowledge discovery and data mining. ACM, 2012, pp. 931–939.

[22] A. Shrivastava and P. Li, "Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips)," in *Advances in Neural Information Processing Systems*, 2014, pp. 2321–2329.

[23] A. Andoni, P. Indyk, H. L. Nguyen, and I. Razenshteyn, "Beyond locality-sensitive hashing," in *Proc. of the 25th SODA*. SIAM, 2014, pp. 1018–1028.

[24] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt, "Practical and optimal lsh for angular distance," in *Advances in Neural Information Processing Systems*, 2015, pp. 1225–1233.

[25] A. Andoni and I. Razenshteyn, "Optimal data-dependent hashing for approximate near neighbors," in *Proc. of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. ACM, 2015, pp. 793–801.

[26] B. Neyshabur and N. Srebro, "On symmetric and asymmetric lshs for inner product search," in *Proc. of the 32nd International Conference on Machine Learning, ICML*, 2015, pp. 1926–1934.

[27] A. Shrivastava and P. Li, "Asymmetric minwise hashing for indexing binary inner products and set containment," in *Proc. of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 981–991.

[28] G. Valiant, "Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem," *Journal of the ACM (JACM)*, vol. 62, no. 2, p. 13, 2015.

[29] J. Alman and R. Williams, "Probabilistic polynomials and hamming nearest neighbors," in *Proc. of the 56th FOCS*. IEEE, 2015, pp. 136–150.

[30] M. Karppa, P. Kaski, and J. Kohonen, "A faster subquadratic algorithm for finding outlier correlations," in *Proc. of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2016, pp. 1288–1305.

[31] T. D. Ahle, R. Pagh, I. Razenshteyn, and F. Silvestri, "On the complexity of inner product similarity join," in *Proc. of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. ACM, 2016, pp. 151–164.

[32] C. Teflioudi and R. Gemulla, "Exact and approximate maximum inner product search with lemp," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 1, p. 5, 2016.

[33] T. Christiani and R. Pagh, "Set similarity search beyond minhash," *arXiv preprint arXiv:1612.07710*, 2016.

[34] T. Christiani, "A framework for similarity search with space-time tradeoffs using locality-sensitive filtering," in *Proc. of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2017, pp. 31–46.

[35] S. Arora and S. Safra, "Probabilistic checking of proofs: A new characterization of NP," *J. ACM*, vol. 45, no. 1, pp. 70–122, 1998. [Online]. Available: http://doi.acm.org/10.1145/273865.273901

[36] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," *J. ACM*, vol. 45, no. 3, pp. 501–555, 1998. [Online]. Available: http://doi.acm.org/10.1145/278298.278306

[37] I. Dinur, "The PCP theorem by gap amplification," *J. ACM*, vol. 54, no. 3, p. 12, 2007. [Online]. Available: http://doi.acm.org/10.1145/1236457.1236459

[38] E. Ben-Sasson, Y. Kaplan, S. Kopparty, O. Meir, and H. Stichtenoth, "Constant rate pcps for circuit-sat with sublinear query complexity," *J. ACM*, vol. 63, no. 4, pp. 32:1–32:57, 2016. [Online]. Available: http://doi.acm.org/10.1145/2901294

[39] I. Dinur, "Mildly exponential reduction from gap 3sat to polynomial-gap label-cover," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 23, p. 128, 2016. [Online]. Available: http://eccc.hpi-web.de/report/2016/128

[40] O. Reingold, March 2017, private Communication.

[41] B. Kalyanasundaram and G. Schnitger, "The probabilistic communication complexity of set intersection," *SIAM J. Discrete Math.*, vol. 5, no. 4, pp. 545–557, 1992. [Online]. Available: http://dx.doi.org/10.1137/0405044

[42] A. A. Razborov, "On the distributional complexity of disjointness," *Theor. Comput. Sci.*, vol. 106, no. 2, pp. 385–390, 1992. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(92)90260-M

[43] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, "An information statistics approach to data stream and communication complexity," *J. Comput. Syst. Sci.*, vol. 68, no. 4, pp. 702–732, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.jcss.2003.11.006

[44] M. Karchmer, E. Kushilevitz, and N. Nisan, "Fractional covers and communication complexity," *SIAM J. Discrete Math.*, vol. 8, no. 1, pp. 76–92, 1995. [Online]. Available: http://dx.doi.org/10.1137/S0895480192238482

[45] S. Aaronson and A. Wigderson, "Algebrization: A new barrier in complexity theory," *TOCT*, vol. 1, no. 1, pp. 2:1–2:54, 2009. [Online]. Available: http://doi.acm.org/10.1145/1490270.1490272

[46] R. Motwani, A. Naor, and R. Panigrahy, "Lower bounds on locality sensitive hashing," *SIAM Journal on Discrete Mathematics*, vol. 21, no. 4, pp. 930–935, 2007.

[47] A. Andoni, D. Croitoru, and M. Patrascu, "Hardness of nearest neighbor under l-infinity," in *Proc. of the 49th FOCS*, 2008, pp. 424–433.

[48] R. O?Donnell, Y. Wu, and Y. Zhou, "Optimal lower bounds for locality-sensitive hashing (except when q is tiny)," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 1, p. 5, 2014.

[49] A. Andoni, P. Indyk, and M. Patrascu, "On the optimality of the dimensionality reduction method," in *Proc. of the 47th FOCS*. IEEE, 2006, pp. 449–458.

[50] R. Panigrahy, K. Talwar, and U. Wieder, "A geometric approach to lower bounds for approximate near-neighbor search and partial match," in *Proc. of the 49th FOCS*. IEEE, 2008, pp. 414–423.

[51] ——, "Lower bounds on near neighbor search via metric expansion," in *Proc. of the 51st FOCS*. IEEE, 2010, pp. 805–814.

[52] M. Kapralov and R. Panigrahy, "Nns lower bounds via metric expansion for l? and emd," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2012, pp. 545–556.

[53] A. Abdullah and S. Venkatasubramanian, "A directed isoperimetric inequality with application to bregman near neighbor lower bounds," in *Proc. of the 47th STOC*. ACM, 2015, pp. 509–518.

[54] A. Andoni, T. Laarhoven, I. P. Razenshteyn, and E. Waingarten, "Optimal hashing-based time-space trade-offs for approximate near neighbors," in *Proc. of the 28th SODA*, 2017, pp. 47–66.

[55] K. Ramasamy, J. M. Patel, J. F. Naughton, and R. Kaushik, "Set containment joins: The good, the bad and the ugly." in *VLDB*, 2000, pp. 351–362.

[56] S. Melnik and H. Garcia-Molina, "Adaptive algorithms for set containment joins," *ACM Transactions on Database Systems (TODS)*, vol. 28, no. 1, pp. 56–99, 2003.

[57] P. Agrawal, A. Arasu, and R. Kaushik, "On indexing error-tolerant set containment," in *Proc. of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 927–938.

[58] A. Goel and P. Gupta, "Small subset queries and bloom filters using ternary associative memories, with applications," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1, pp. 143–154, 2010.

[59] R. L. Rivest, "Analysis of associative retrieval algorithms." Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1974, aAI7420230.

[60] ——, “Partial-match retrieval algorithms,” *SIAM Journal on Computing*, vol. 5, no. 1, pp. 19–50, 1976.

[61] G. M. Landau, E. W. Myers, and J. P. Schmidt, “Incremental string comparison,” *SIAM Journal on Computing*, vol. 27, no. 2, pp. 557–582, 1998.

[62] Z. Bar-Yossef, T. Jayram, R. Krauthgamer, and R. Kumar, “Approximating edit distance efficiently,” in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*. IEEE, 2004, pp. 550–559.

[63] T. Batu, F. Ergun, and C. Sahinalp, “Oblivious string embeddings and edit distance approximations,” in *Proc. of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics, 2006, pp. 792–801.

[64] R. Ostrovsky and Y. Rabani, “Low distortion embeddings for edit distance,” *Journal of the ACM (JACM)*, vol. 54, no. 5, p. 23, 2007.

[65] A. Andoni, R. Krauthgamer, and K. Onak, “Polylogarithmic approximation for edit distance and the asymmetric query complexity,” in *FOCS*, 2010, pp. 377–386.

[66] A. Andoni and K. Onak, “Approximating edit distance in near-linear time,” *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1635–1648, 2012.

[67] A. Abboud and A. Backurs, “Towards hardness of approximation for polynomial time problems,” in *ITCS, to appear*, 2017.

[68] A. Andoni, M. Deza, A. Gupta, P. Indyk, and S. Raskhodnikova, “Lower bounds for embedding edit distance into normed spaces,” in *Proc. of the 14th SODA*, 2003, pp. 523–526.

[69] S. C. Sahinalp and A. Utis, “Hardness of string similarity search and other indexing problems,” in *International Colloquium on Automata, Languages, and Programming*. Springer, 2004, pp. 1080–1098.

[70] S. Khot and A. Naor, “Nonembeddability theorems via fourier analysis,” in *Proc. of the 46th FOCS*. IEEE, 2005, pp. 101–110.

[71] A. Andoni and R. Krauthgamer, “The computational hardness of estimating edit distance [extended abstract],” in *Proc. of the 48th FOCS*, 2007, pp. 724–734.

[72] A. Andoni, T. S. Jayram, and M. Patrascu, “Lower bounds for edit distance and product metrics via poincaré-type inequalities,” in *Proc. of the 21st SODA*, 2010, pp. 184–192.

[73] G. Cormode, S. Muthukrishnan, and S. C. Sahinalp, “Permutation editing and matching via embeddings,” in *International Colloquium on Automata, Languages, and Programming*. Springer, 2001, pp. 481–492.

[74] A. Andoni, P. Indyk, and R. Krauthgamer, “Overcoming the l_1 non-embeddability barrier: algorithms for product metrics,” in *Proc. of the 20th SODA*, 2009, pp. 865–874.

[75] K. Thompson, “Programming techniques: Regular expression search algorithm,” *Communications of the ACM*, vol. 11, no. 6, pp. 419–422, 1968.

[76] E. W. Myers and W. Miller, “Approximate matching of regular expressions,” *Bulletin of mathematical biology*, vol. 51, no. 1, pp. 5–37, 1989.

[77] G. Myers, “A four russians algorithm for regular expression pattern matching,” *Journal of the ACM (JACM)*, vol. 39, no. 2, pp. 432–448, 1992.

[78] S. Wu, U. Manber, and E. Myers, “A subquadratic algorithm for approximate regular expression matching,” *Journal of algorithms*, vol. 19, no. 3, pp. 346–360, 1995.

[79] J. R. Knight and E. W. Myers, “Approximate regular expression pattern matching with concave gap penalties,” *Algorithmica*, vol. 14, no. 1, pp. 85–121, 1995.

[80] E. Myers, P. Oliva, and K. Guimarães, “Reporting exact and approximate regular expression matches,” in *Combinatorial pattern matching*. Springer, 1998, pp. 91–103.

[81] G. Navarro, “Approximate regular expression searching with arbitrary integer weights.” *Nord. J. Comput.*, vol. 11, no. 4, pp. 356–373, 2004.

[82] D. Belazzougui and M. Raffinot, “Approximate regular expression matching with multi-strings,” *Journal of Discrete Algorithms*, vol. 18, pp. 14–21, 2013.

[83] P. Bille and M. Thorup, “Faster regular expression matching,” in *International Colloquium on Automata, Languages, and Programming*. Springer, 2009, pp. 171–182.

[84] K. Bringmann, A. Grønlund, and K. G. Larsen, “A dichotomy for regular expression membership testing,” *CoRR*, vol. abs/1611.00918, 2016. [Online]. Available: http://arxiv.org/abs/1611.00918

[85] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani, “Fast estimation of diameter and shortest paths (without matrix multiplication),” *SIAM J. Comput.*, vol. 28, no. 4, pp. 1167–1181, 1999.

[86] S. Chechik, D. H. Larkin, L. Roditty, G. Schoenebeck, R. E. Tarjan, and V. V. Williams, “Better approximation algorithms for the graph diameter,” in *Proc. of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2014, pp. 1041–1052.

[87] P. Indyk, “A sublinear time approximation scheme for clustering in metric spaces,” in *40th Annual Symposium on Foundations of Computer Science, FOCS*, 1999, pp. 154–159. [Online]. Available: http://dx.doi.org/10.1109/SFFCS.1999.814587

[88] Ö. Egecioglu and B. Kalantari, “Approximating the diameter of a set of points in the euclidean space,” *Inf. Process. Lett.*, vol. 32, no. 4, pp. 205–211, 1989. [Online]. Available: http://dx.doi.org/10.1016/0020-0190(89)90045-8

[89] D. V. Finocchiaro and M. Pellegrini, “On computing the diameter of a point set in high dimensional euclidean space,” *Theor. Comput. Sci.*, vol. 287, no. 2, pp. 501–514, 2002. [Online]. Available: http://dx.doi.org/10.1016/S0304-3975(01)00258-4

[90] A. Borodin, R. Ostrovsky, and Y. Rabani, “Subquadratic approximation algorithms for clustering problems in high dimensional spaces,” *Machine Learning*, vol. 56, no. 1-3, pp. 153–167, 2004. [Online]. Available: http://dx.doi.org/10.1023/B:MACH.0000033118.09057.80

[91] P. Indyk, “Dimensionality reduction techniques for proximity problems,” in *Proc. of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000, pp. 371–378. [Online]. Available: http://dl.acm.org/citation.cfm?id=338219.338582

[92] A. Goel, P. Indyk, and K. R. Varadarajan, “Reductions among high dimensional proximity problems,” in *Proc. of the Twelfth Annual Symposium on Discrete Algorithms*, 2001, pp. 769–778. [Online]. Available: http://dl.acm.org/citation.cfm?id=365411.365776

[93] P. Indyk, “Better algorithms for high-dimensional proximity problems via asymmetric embeddings,” in *Proc. of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003, pp. 539–545. [Online]. Available: http://dl.acm.org/citation.cfm?id=644108.644200

[94] ——, “On approximate nearest neighbors in non-euclidean spaces,” in *39th Annual Symposium on Foundations of Computer Science, FOCS*, 1998, pp. 148–155. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1998.743438

[95] ——, “Approximate nearest neighbor algorithms for frechet distance via product metrics,” in *Proc. of the 18th Annual Symposium on Computational Geometry,*, 2002, pp. 102–106. [Online]. Available: http://doi.acm.org/10.1145/513400.513414

[96] G. Cormode and S. Muthukrishnan, “Space efficient mining of multigraph streams,” in *Proc. of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 2005, pp. 271–282. [Online]. Available: http://doi.acm.org/10.1145/1065167.1065201

[97] T. S. Jayram and D. P. Woodruff, "The data stream space complexity of cascaded norms," in *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2009, pp. 765–774. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2009.82

[98] P. Indyk, "Approximate nearest neighbor under edit distance via product metrics," in *Proc. of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2004, pp. 646–650. [Online]. Available: http://dl.acm.org/citation.cfm?id=982792.982889

[99] M. I. Shamos and D. Hoey, "Closest-point problems," in *Proc. of the 16th FOCS*. IEEE, 1975, pp. 151–162.

[100] R. David, Karthik C. S., and B. Laekhanukit, "The curse of medium dimension for geometric problems in almost every norm," *CoRR*, vol. abs/1608.03245, 2016. [Online]. Available: http://arxiv.org/abs/1608.03245

[101] A. Abboud, T. D. Hansen, V. V. Williams, and R. Williams, "Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made," in *Proc. of the 48th STOC*, 2016, pp. 375–388.

[102] R. R. Williams, "Strong ETH breaks with merlin and arthur: Short non-interactive proofs of batch evaluation," in *31st Conference on Computational Complexity, CCC*, 2016, pp. 2:1–2:17. [Online]. Available: http://dx.doi.org/10.4230/LIPIcs.CCC.2016.2

[103] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Average-case fine-grained hardness," *IACR Cryptology ePrint Archive*, vol. 2017, p. 202, 2017. [Online]. Available: http://eprint.iacr.org/2017/202

[104] Y. T. Kalai and R. Raz, "Interactive PCP," in *Automata, Languages and Programming, 35th International Colloquium, ICALP*, 2008, pp. 536–547.

[105] E. Ben-Sasson, A. Chiesa, and N. Spooner, "Interactive oracle proofs," in *Theory of Cryptography - 14th International Conference, TCC*, 2016, pp. 31–60.

[106] O. Reingold, G. N. Rothblum, and R. D. Rothblum, "Constant-round interactive proofs for delegating computation," in *Proc. of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2016, pp. 49–62.

[107] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: Interactive proofs for muggles," *J. ACM*, vol. 62, no. 4, pp. 27:1–27:64, 2015. [Online]. Available: http://doi.acm.org/10.1145/2699436

[108] V. Goyal, Y. Ishai, M. Mahmoody, and A. Sahai, "Interactive locking, zero-knowledge pcps, and unconditional cryptography," in *Advances in Cryptology - CRYPTO, 30th Annual Cryptology Conference,*, 2010, pp. 173–190.

[109] E. Ben-Sasson, A. Chiesa, A. Gabizon, M. Riabzev, and N. Spooner, "Short interactive oracle proofs with constant query complexity, via composition and sumcheck," *IACR Cryptology ePrint Archive*, vol. 2016, p. 324, 2016. [Online]. Available: http://eprint.iacr.org/2016/324

[110] A. Drucker, "PCPs for Arthur-Merlin Games and Communication Protocols," Master's thesis, Massachusetts Institute of Technology, 2010.

[111] M. Gupta and R. Peng, "Fully dynamic (1+ e)-approximate matchings," in *Proc. of the 54th FOCS*. IEEE, 2013, pp. 548–557.

[112] O. Neiman and S. Solomon, "Simple deterministic algorithms for fully dynamic maximal matching," *ACM Transactions on Algorithms (TALG)*, vol. 12, no. 1, p. 7, 2016.

[113] S. Bhattacharya, M. Henzinger, and G. F. Italiano, "Deterministic fully dynamic data structures for vertex cover and matching," in *Proc. of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2014, pp. 785–804.

[114] S. Baswana, M. Gupta, and S. Sen, "Fully dynamic maximal matching in o(\logn) update time," *SIAM Journal on Computing*, vol. 44, no. 1, pp. 88–113, 2015.

[115] A. Bernstein and C. Stein, "Fully dynamic matching in bipartite graphs," *arXiv preprint arXiv:1506.07076*, 2015.

[116] S. Bhattacharya, M. Henzinger, and G. F. Italiano, "Design of dynamic algorithms via primal-dual method," *arXiv preprint arXiv:1604.05337*, 2016.

[117] A. Bernstein and C. Stein, "Faster fully dynamic matchings with small approximation ratios," in *Proc. of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2016, pp. 692–711.

[118] D. Peleg and S. Solomon, "Dynamic (1+ ε)-approximate matchings: a density-sensitive approach," in *Proc. of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2016, pp. 712–729.

[119] S. Bhattacharya, M. Henzinger, and D. Nanongkai, "New deterministic approximation algorithms for fully dynamic matching," in *Proc. of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2016, pp. 398–411.

[120] S. Solomon, "Fully dynamic maximal matching in constant update time," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*. IEEE, 2016, pp. 325–334.

[121] M. Pătraşcu, "Towards polynomial lower bounds for dynamic problems," in *Proc. of the 42nd Annual ACM Symposium on Theory Of Computing (STOC)*, 2010, pp. 603–610.

[122] T. Kopelowitz, S. Pettie, and E. Porat, "Higher lower bounds from the 3sum conjecture," in *Proc. of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2016, pp. 1272–1287.

[123] M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak, "Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture," in *Proc. of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, 2015, pp. 21–30.

[124] S. Dahlgaard, "On the hardness of partially dynamic graph problems and connections to diameter," in *43rd International Colloquium on Automata, Languages, and Programming, ICALP*, 2016, pp. 48:1–48:14.

[125] A. Gajentaan and M. H. Overmars, "On a class of o(n2) problems in computational geometry," *Comput. Geom.*, vol. 5, pp. 165–185, 1995.

[126] K. G. Larsen and R. R. Williams, "Faster online matrix-vector multiplication," in *Proc. of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2017, pp. 2182–2189.

[127] C. Lund, L. Fortnow, H. J. Karloff, and N. Nisan, "Algebraic methods for interactive proof systems," *J. ACM*, vol. 39, no. 4, pp. 859–868, 1992. [Online]. Available: http://doi.acm.org/10.1145/146585.146605

[128] R. Raz, "A parallel repetition theorem," *SIAM J. Comput.*, vol. 27, no. 3, pp. 763–803, 1998. [Online]. Available: http://dx.doi.org/10.1137/S0097539795280895

[129] J. Håstad, "Some optimal inapproximability results," *J. ACM*, vol. 48, no. 4, pp. 798–859, 2001. [Online]. Available: http://doi.acm.org/10.1145/502090.502098

[130] H. Buhrman, R. Cleve, and A. Wigderson, "Quantum vs. classical communication and computation," in *Proc. of the Thirtieth Annual ACM Symposium on the Theory of Computing*, 1998, pp. 63–68. [Online]. Available: http://doi.acm.org/10.1145/276698.276713