Fine-Grained Reductions and Quantum Speedups for Dynamic Programming

Amir Abboud
IBM Almaden Research Center, San Jose, California, USA
amir.abboud@ibm.com

Abstract

This paper points at a connection between certain (classical) fine-grained reductions and the question: Do quantum algorithms offer an advantage for problems whose (classical) best solution is via dynamic programming?

A remarkable recent result of Ambainis et al. [SODA 2019] indicates that the answer is positive for some fundamental problems such as Set-Cover and Travelling Salesman. They design a quantum $O^*(1.728^n)$ time algorithm whereas the dynamic programming $O^*(2^n)$ time algorithms are conjectured to be classically optimal. In this paper, fine-grained reductions are extracted from their algorithms giving the first lower bounds for problems in P that are based on the intriguing Set-Cover Conjecture (SeCoCo) of Cygan et al. [CCC 2010].

In particular, the SeCoCo implies:

- a super-linear $\Omega(n^{1.08})$ lower bound for 3-SUM on n integers,
- an $\Omega(n^{\frac{k}{c_k}-\varepsilon})$ lower bound for k-SUM on n integers and k-Clique on n-node graphs, for *any* integer $k \geq 3$, where $c_k \leq \log_2 k + 1.4427$.

While far from being tight, these lower bounds are significantly stronger than what is known to follow from the Strong Exponential Time Hypothesis (SETH); the well-known $n^{\Omega(k)}$ ETH-based lower bounds for k-Clique and k-SUM are vacuous when k is constant.

Going in the opposite direction, this paper observes that some "sequential" problems with previously known fine-grained reductions to a "parallelizable" core also enjoy quantum speedups over their classical dynamic programming solutions. Examples include RNA Folding and Least-Weight Subsequence.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Fine-Grained Complexity, Set-Cover, 3-SUM, k-Clique, k-SUM, Dynamic Programming, Quantum Algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.8

Category Track A: Algorithms, Complexity and Games

Funding We acknowledge the support of the Quantum Computing Sciences program of the U.S. Air Force, Office of Scientific Research, administered through Air Force Research Laboratory contract FA8750-18-C-0098.

Acknowledgements We thank Karl Bringmann and the anonymous reviewers for helpful feedback.

1 Introduction

An increasing amount of effort is being dedicated to the question: When and by how much can quantum algorithms beat classical ones? Perhaps the most successful approach for getting quantum speedups is using Grover's search [22], which offers a quadratic improvement over classical exhaustive search, which is the bottleneck in the best-known algorithms for a long list of problems. This list includes nearly all problems studied in fine-grained complexity such as SAT, Orthogonal Vectors, 3-SUM, All-Pairs Shortest Paths, and so on. In fact, none

of the popular conjectures in fine-grained complexity (e.g. SETH) remain plausible when Grover's quantum search is allowed (with one exception, to be discussed shortly). It gives magical capabilities such as the following theorem.

▶ **Theorem 1** (Quantum Minimum Finding [17]). *Let $a_1, \ldots, a_n$ be integers accessed by a procedure $\mathcal{P}$. There exists a quantum algorithm that finds $\min_{i=1}^{n}\{a_i\}$ with success probability at least $2/3$ using $O(\sqrt{n})$ applications of $\mathcal{P}$.*

But what about problems whose best known algorithm is via dynamic programming, not exhaustive search? Could those problems be resilient to quantum speedups? A famous open question in this context is whether we can solve sequence alignment problems such Edit Distance and Longest Common Subsequence in truly-subquadratic quantum time. Such an algorithm could lead to important progress in bioinformatics in the future. The main challenge is the sequential nature of dynamic programming, which prevents us from using a Grover-like approach; one first solves the problem on small instances and then combines them to solve larger and larger instances. None of the steps involve an expensive exhaustive search. Another example is the Set-Cover problem, which can be solved in $O^*(2^n)$ time[1] with classical dynamic programming [19], and is conjectured by Cygan et al. [15] to have an $\Omega((2-\delta)^n)$ lower bound, for all $\delta > 0$. This is the so-called Set-Cover Conjecture (SeCoCo). Would this popular conjecture remain plausible in a quantum world?

A recent breakthrough of Ambainis et al. [5] shows otherwise. The authors give quantum $O^*(1.728^n)$ time algorithms for Set-Cover and Traveling Salesman, as well as other exponential speedups for problems such as Graph Bandwidth and Feedback Arc Set. These are problems where the best known classical algorithm is via dynamic programming. (It is unclear if their techniques will lead to solving Edit Distance in truly subquadratic quantum time.)

Ambainis et al. find a way to use the quantum minimum finding theorem above to solve Set-Cover. This can be viewed as a *reduction* from Set-Cover to a "parallelizable" core, and for the final algorithm to be fast the reduction should be efficient, qualifying it as a *fine-grained reduction* from Set-Cover to some minimum-finding problem. The main observation here is that the parallelizable-nature of the latter problem allows for further reductions to the popular problems in fine-grained complexity such as 3-SUM and k-Clique. Previously, no interesting reductions from Set-Cover to natural problems in P were known, and this gives a partial resolution to an open question in fine-grained complexity (see Section 5.1 in [39] and the discussion at the end of this paper). The sequential nature of Set-Cover had been a barrier for reductions as well, and the ideas of Ambainis et al. overcome it. The next subsection states the results and elaborates on their importance to the landscape of fine-grained complexity.

Section 3 suggests that this connection between quantum speedups for sequential problems and fine-grained reductions could be interesting also in the other direction.

1.1 The Consequences to Fine-Grained Complexity

The goal of fine-grained complexity and algorithms is to achieve upper and lower bounds that are as tight as possible for the computational problems of interest. The lower bounds are obtained via reductions and are based on a small set of popular conjectures, such as the Strong Exponential Time Hypothesis (SETH)[2] [23, 24], regarding the hardness of certain

[1] The notation $O^*(\cdot)$ hides polynomial factors.

[2] ETH states that 3-SAT cannot be solved in $2^{o(n)}$ time. The stronger version, SETH, states that we cannot solve k-SAT in $O((2-\varepsilon)^n)$ time for all k.

core problems, e.g. CNF-SAT. By now, there is a very long and evergrowing list of lower bounds that are tight up to $n^{o(1)}$ factors. See the recent surveys [40, 37]. One of the main gaps in knowledge, when it comes to this framework, is regarding the connections between the conjectures. For example, the k-SUM and k-Clique conjectures were used to prove many lower bound results that SETH (or any other conjectures) seem incapable of proving. Many conjectures have their own "hardness-class" and the whole theory would be much better if these conjectures can be supported by SETH or other (but different) conjectures.

k-SUM and k-Clique

The k-SUM problem asks to find k among n given numbers that sum to zero. It can be solved in $O(n^{\lceil k/2 \rceil})$ time which is conjectured to be optimal up to $n^{o(1)}$ factors [3]. The $k = 3$ case is the 3-SUM conjecture which is an old and famous conjecture in computational geometry [20]. This conjecture, for an even k, is supported by the hypothesis that Subset-Sum cannot be solved in $O(2^{(1/2-\varepsilon)n})$ for some $\varepsilon > 0$. A reduction of Patrascu and Williams [33] gives an $n^{\Omega(k)}$ lower bound for k-SUM assuming the ETH, when k is super-constant.

The k-Clique problem asks to find k among n nodes that form a clique. It can be solved in $O(n^{\frac{\omega}{3} \cdot k})$ time[3] [32, 18], where ω is the matrix multiplication exponent [38, 21], and this is conjectured to be optimal (and that $\omega = 2 + o(1)$) [1]. This conjecture, for k divisible by 3, is supported by the hypothesis that Max-Cut cannot be solved exponentially faster than its current runtime, since the current-best algorithm by Williams reduces Max-Cut to k-Clique [36]. The k-SUM conjecture implies an $n^{\lceil k/2 \rceil - o(1)}$ lower bound for k-Clique [4], and the ETH implies an $n^{\Omega(k)}$ lower bound for super-constant k [12, 13].

The aforementioned ETH lower bounds do not imply anything for these two problems when k is constant. No nontrivial SETH-based lower bounds are known, for any constant k.

SeCoCo vs. SETH

In a seminal paper on fine-grained reductions among NP-complete problems, Cygan et al. [15] introduced SeCoCo and used it to prove tight lower bounds for problems with dynamic programming solutions. It states that Set-Cover on n elements cannot be solved in $(2 - \delta)^n$ time, even when the sets have constant size (more formally defined in Section 2). Since then SeCoCo has been utilized for lower bounds for other NP-hard problems [6, 9, 26, 28].

Currently, SETH and SeCoCo seem incomparable and each of them has some advantages over the other as a hardness assumption. However, it is likely that SeCoCo will turn out to be a strictly safer (weaker, more believable) conjecture; there is no known barrier for (tightly) reducing SAT to Set-Cover and, at least back in 2010, Cygan et al. conjectured that such a reduction exists. On the other hand, a (tight) reduction in the other way, from Set-Cover to SAT, would be much more surprising. Unless the reduction is unusual, it would allow one to use the trivial 2^n algorithm for SAT to solve Set-Cover in 2^n time in a way that is simpler (no dynamic programming) and better in some ways (e.g. parallelizable). Even with today's knowledge, the latter point demonstrates an advantage of SeCoCo as a basis for reductions: it gives barriers for (e.g.) solving the end problem without dynamic programming. Meanwhile, SETH enjoys other benefits: First, it is much more popular. Second, SAT has been more extensively studied than Set-Cover in many areas such as complexity theory and verification. And third, refuting it implies some (weak) circuit lower bounds; this does not make it more believable, but it does "raise the stakes" in a way that is not known for SeCoCo. The reader is referred to [15, 31, 2, 27] for other discussions on the matter.

[3] The bound is slightly more complicated bound when k is not divisible by 3.

New Results

The main result of this paper is an extraction of fine-grained reductions from the quantum algorithms of Ambainis et al. leading to the first nontrivial lower bounds in P that are based on the Set-Cover Conjecture. The lower bounds can be based on a weaker form of the conjecture (discussed in Section 2) where sets are allowed to have size up to $n^{o(1)}$. The full version of the paper shows the same lower bound for the k-Orthogonal Vectors problem, via a minor modification to the reductions.

▶ **Theorem 2.** *Let $k \geq 3$ and[4] $c_k = k \cdot H(k^{-1}) = \log_2(k-1) + k \cdot \log \frac{k}{k-1}$. If for some $\varepsilon > 0$, either*

- *the k-Sum problem on n integers, or*
- *the k-Clique problem on n node graphs and $n^{2-\varepsilon}$ edges,*

can be solved in $O(n^{\frac{k}{c_k}-\varepsilon})$ time, then the (weak) SeCoCo is false.

Recall that the conjectured lower bounds are $1/c_k = 1/2$ for k-SUM and $1/c_k = 2/3$ for k-Clique. Whereas here, as k grows, the function c_k approaches $\log_2 k + 1.4427$, which means that the lower bound is approximately $\Omega(n^{\frac{k}{\log k}})$. It is non-trivial for k-Clique for all $k \geq 9$, while for k-SUM (where the input size is smaller) it is meaningful already in the most famous case of $k = 3$.

▶ **Corollary 3.** *If 3-SUM on n integers can be solved in $O(n^{1/H(1/3)-\varepsilon}) = O(n^{1.089-\varepsilon})$ time, for some $\varepsilon > 0$, then the (weak) SeCoCo is false.*

Perhaps the most exciting aspect of these results is that one can finally have a concrete lower bound for k-Clique and k-SUM for constant k. For instance, it implies an n^{10} lower bound for 2^{11}-Clique and 2^{11}-Sum. These two problems are canonical in parameterized complexity as they can be reduced to many other problems while preserving the boundedness of the natural parameter. Thus, the SeCoCo conjecture implies concrete lower bounds for all those other problems as well, when the parameter is a fixed constant. It is still an important open question to prove such lower bounds with a fixed $c_k > 0$ that is independent of k.

Replacing SeCoCo with SETH is a big open question. It is likely to be possible; in particular, it would follow if a reduction from SAT to Set-Cover is found. Such a reduction was conjectured to exist by Cygan et al. [15]. Notably, this suggests a barrier for proving much higher lower bounds for 3-SUM. It is known that the nondeterministic version of SETH (NSETH) rules out a SETH lower bound for 3-SUM of $\Omega(n^{1.5+\varepsilon})$ [11], and by the conjecture of Cygan et al. this is also a barrier for SeCoCo-based lower bounds.

Further open questions are discussed at the end of the paper. Since the algorithms of Ambainis et al. [5] for Set-Cover and for TSP are very similar, the exact same lower bounds above can be based on a corresponding assumption about the hardness of TSP. The modified proofs are deferred to the full version of the paper.

Less-Fine-Grained Complexity: Linear vs. Super-Linear

Traditional complexity theory classifies problems into polynomial (efficient) vs. super-polynomial (inefficient). Due, in part, to the increase in the data sizes this classification is now viewed as often being too coarse. Fine-Grained complexity's goal is to get a more exact classification that will be more relevant in many scenarios. It is natural to view the

[4] Let $H(p)$ be the binary entropy function on $p \in [0,1]$. See Section 2.

first-order goal of such a theory as: classifying problems into near-linear time solvable ones, those that can be solved "efficiently" even in "Big-Data" settings, vs. the ones that require super-linear time to solve. The latter set of problems will have to be relaxed before they can be solved at large scales. The more valuable and pleasing exact classification can be viewed as the next-step after this *less-fine-grained* classification is achieved.

This less-fine-grained classification has been conditionally achieved for a long list of problems, and the results of this paper strengthen the foundations for some of these results. For example, many problems in computational geometry are 3-SUM-hard in the sense that there is a linear-time reduction from 3-SUM to them. If these problems can be solved in near-linear time, then 3-SUM can, which is conjectured to be impossible (by a much weaker version of the 3-SUM conjecture). But what are other justifications for these "lower bounds"? Corollary3 gives a new one: such near-linear time algorithms would refute SeCoCo.

1.2 Other Related Work

Another connection between fine-grained complexity and quantum computing was recently demonstrated by Chen and Wang [14]. The authors show that a fast and communication-efficient *quantum* protocol for a function f implies a fast classical approximate counting algorithm for a related pair counting problem. They instantiate this connection in order to show a new approximate counting algorithm for the #-Orthogonal-Vectors problem.

A very recent paper by Khadiev [25] suggests a new quantum dynamic programming approach for problems on DAGs. It will be interesting to see if fine-grained reductions can be extracted.

2 Fine-Grained Reductions

Preliminaries

The standard notation $[n] = \{1, \ldots, n\}$ will be used throughout the paper. The starting point of the reductions in this paper is Set-Cover.

▶ **Definition 4** (The Set-Cover Problem). *Given m sets over the universe $U = [n]$, return the minimum number of sets required to cover U.*

The Set-Cover Conjecture (SeCoCo) of Cygan et al. states that the problem requires $2^{n-o(n)}$ time, even when all sets have constant size. Krauthgamer and Trabelsi [27] demonstrate that the Log-Set-Cover Conjecture, where the sets can have size up to $O(\log n)$, is equally useful but potentially more believable. They show that refuting it implies a breakthrough algorithm for Directed Hamiltonicity and Directed n-Tree. For the lower bounds in this paper, an even more relaxed conjecture is sufficient[5], where the sets are allowed to be of any size $n^{o(1)}$. Note that this automatically bounds the number of sets by $m = 2^{o(n)}$ which will be negligible.

The Weak Set-Cover Conjecture. No algorithm can solve Set-Cover over the universe $[n]$ with sets of size $n^{o(1)}$ in time $O((2-\delta)^n)$ where $\delta > 0$.

The main idea in the reductions, which is the crux of the quantum algorithms in [5], is an unusual split-and-list where all subsets of $[n]$ of size up to $d = n/k$ are enumerated. The analysis will rely on the following approximation of binomial coefficients, bounding the number of such sets:

[5] Although the stronger assumption would allow to extend the lower bound for larger values of k beyond $n^{1-\Omega(1)}$.

▶ **Lemma 5** (Entropy Approximation). *For all* $1 \leq d \leq n/2$,

$$\binom{n}{\leq d} = \sum_{i=1}^{d} \binom{n}{d} \leq 2^{H(d/n)\cdot n},$$

where $H(\epsilon) = -(\epsilon \log_2 \epsilon + (1-\epsilon)\log_2(1-\epsilon))$ *is the binary entropy of* $0 \leq \epsilon \leq 1$.

Observe that $H(1/k) = \frac{\log_2 k + O(1)}{k}$. In the proofs, the following bounds will be used for any $1 \leq k \leq n^{\varepsilon}$ where $\varepsilon < 1$: $\binom{n}{\leq n/k} \leq 2^{H(1/k)\cdot n}$ and when $t = n^{o(1)}$ then $\binom{n}{\leq n/k+t} \leq 2^{(H(1/k)+o(1))\cdot n}$. The last inequality follows because $\binom{n}{\leq n/k+t} \leq n^{o(1)} \cdot \binom{n}{n/k+t} \leq n^{o(1)} \cdot \binom{n}{(1+o(1))\cdot n/k} \leq 2^{H((1+o(1))/k)\cdot n + o(n)} \leq 2^{(1+o(1))\cdot H(1/k)\cdot n + o(n)} \leq 2^{(H(1/k)+o(1))\cdot n}$.

Key Observation

Following the notation in Ambainis et al., for a set system $\mathcal{S}$ over a universe U we denote the size of the minimum set-cover by $f(U, \mathcal{S})$. By definition, it follows that for any k sets $U_1, \ldots, U_k$ with union equal to U and any $\mathcal{S}$:

$$\sum_{i=1} f(U_i, \mathcal{S}) \geq f(U, \mathcal{S})$$

The central claim for all the reductions in this paper (and the quantum algorithm of Ambainis et al.) states that the above will be an equality in some cases.

▷ Claim 6. If all sets in a set system $\mathcal{S}$ over a universe U have size at most t then for all integers $k \geq 2$ such that $t < n/k$ there exist k disjoint subsets $U_1, \ldots, U_k \subseteq U$ each of size between $n/k - t$ and $n/k + t$ such that their union is equal to U and:

$$\sum_{i=1}^{k} f(U_i, \mathcal{S}) \leq f(U, \mathcal{S}).$$

Proof. Let $\mathcal{C} \subseteq \mathcal{S}$ be a minimum set-cover of U of size $\ell = f(U, \mathcal{S})$. To define a partition of U we consider this process: Start from an empty set X and add to it elements from U in the following way. Pick an arbitrary ordering of the sets in $\mathcal{C} = \{S_1, \ldots, S_\ell\}$ and go through them in order, at the i^{th} step we add all elements of S_i to X, unless they were already present.

Two observations about this process: (i) Since $\mathcal{C}$ is a Set-Cover we will end up with $X = U$. And (ii) all sets in $\mathcal{S}$ have size at most t and therefore each step can add up to t elements to X.

Define the sets $U_1, \ldots, U_k$ as follows. U_1 contains all elements added to X up until the earliest step in which $|X|$ became n/k or greater. Due to observation (ii) this guarantees that U_1 contains at most $n/k + t - 1$ elements. U_2 contains "the next" about n/k elements, and so on. More formally, U_i contains all elements added to X after the step in which its size became $\geq i \cdot n/k$ and up until the step in which its size became $\geq (i+1) \cdot n/k$. Again, due to observation (ii) the size of each U_i is upper bounded by $n/k + t$ and lower bounded by $n/k - t$. The sets are disjoint by definition and their union is equal to U.

Finally, we prove the inequality by arguing that each U_i can be covered separately using only the sets from $\mathcal{C}$. This is because one can choose exactly the sets that are responsible for adding the elements to U_i in the process in order to cover U_i. This is a cover by construction and thus $f(U_i, \mathcal{S})$ is upper bounded by this number of sets. And since the sets responsible for each U_i are different, the total number we choose for all the U_i's is exactly $|\mathcal{C}|$. Thus, $\sum_{i=1}^{k} f(U_i, \mathcal{S}) \leq |\mathcal{C}| = f(U, \mathcal{S})$. ◁

Intermediate Problems

This now allows for two preliminary reductions from Set-Cover to a parameterizd version of Exact Cover and to a simpler version of the k-Orthogonal-Vectors problem.

▶ **Definition 7** (The k-Exact-Cover Problem). *Given k lists of sets, each containing N subsets of the same universe U, decide if there are k subsets, one from each list, that are an exact cover of U, that is, they are disjoint and their union is equal to U.*

The following reduction from Set-Cover to this problem is a combination of existing tricks with the core idea of the algorithm by Ambainis et al. which is the enumeration of all subsets of $[n]$ of size equal to n/k up to plus-or-minus the size of a set in the instance, and the preprocessing of their solutions. The correctness will follow from the above key claim.

▶ **Lemma 8.** *For all $2 \leq k \leq n^{\varepsilon}$ where $\varepsilon < 1$, the Set-Cover problem on n elements and m sets of size $n^{o(1)}$ can be reduced to $2^{o(n)}$ instances of the k-Exact-Cover problem on k lists of $N \leq 2^{H(1/k)\cdot n+o(n)}$ subsets over a universe U of size n. The reduction runs in time $2^{H(1/k)\cdot n+o(n)}$.*

Proof. First, reduce Set-Cover to the decision version: is there a set-cover of size exactly ℓ? This can be done simply with an overhead of n in the number of instances by trying all possible values for $1 \leq \ell \leq n$, and returning the smallest one with a positive answer. Thus, it is enough to only consider the decision version.

The goal is to look for a partition of U into k sets that can be covered separately at no extra cost, as promised by Claim 6. Let $U_1, \ldots, U_k$ be such sets, and let $\alpha' = (f(U_1, \mathcal{S}), \ldots, f(U_k, \mathcal{S}))$ be the k-tuple of the sizes of their minimum covers. Note that for all i: $f(U_i, \mathcal{S}) \leq |U_i| \leq n/k + t$ where $t = n^{o(1)}$ is an upper bound on the size of the given sets. As a second preliminary step, we enumerate all tuples α of k numbers in $[n/k + t]$ that sum to ℓ, in an attempt to guess α', and for each tuple there will be an instance of k-Exact-Cover. The total number of tuples is at most $(n/k + t)^k = 2^{o(n)}$. From now on fix a tuple $\alpha = (\alpha_1, \ldots, \alpha_k)$.

Enumerate all subsets of U that are of size up to $n/k + t$, call this collection $\mathcal{P}$. In a preprocessing step, compute the minimum set-cover $f(P, \mathcal{S})$ for all sets $P \in \mathcal{P}$. This can be done in $O(|\mathcal{P}| \cdot m)$ time: In the classical dynamic programming algorithm we compute $f(X, \mathcal{S})$ for increasingly larger sets via the formula $f(X, \mathcal{S}) = \min_{S \in \mathcal{S}}\{f(X \setminus S, \mathcal{S}) + 1\}$. Do the same, but do not process sets such that $|X| > n/k + t$. Thus, the total running time for this step, which is the most expensive in the reduction, is $O(\binom{n}{\leq n/k+t} \cdot m) \leq 2^{H(1/k)\cdot n+o(n)}$.

For all $i \in [k]$, the i^{th} list $\mathcal{L}_i$ in the k-Exact-Cover instance contains all sets $P \in \mathcal{P}$ whose minimum set-cover size matches the guess in α, that is,

$$\mathcal{L}_i = \{P \in \mathcal{P} \mid f(P, \mathcal{S}) = \alpha_i\}.$$

This completes the reduction: at least one of the k-Exact-Cover instances (for some tuple α) is a yes-instance, if and only if there is a set-cover of size ℓ.

For the correctness, observe that for any α and any k sets $X_1 \in \mathcal{L}_1, \ldots, X_k \in \mathcal{L}_k$ whose union is equal to U we have that $f(U, \mathcal{S}) \leq \sum_{i=1}^{k} f(X_i, \mathcal{S}) = \ell$. Thus, if one of the k-Exact-Cover instances is a yes, then $f(U, \mathcal{S}) \leq \ell$. For the other direction, assume that there is a set-cover of size ℓ and consider the partition $U_1, \ldots, U_k$ promised by Claim 6 and let $\alpha = \alpha'$ above be the tuple of minimum set-cover sizes for the k sets in that partition. The instance corresponding to this α is a yes-instance because each U_i will exist in $\mathcal{L}_i$ and their union is equal to U and, moreover, they are disjoint. ◀

The second intermediate problem superficially looks like the k-Orthogonal-Vectors problem (which is the canonical problem for SETH-based lower bounds) but it is in fact easier and is more closely related to k-Clique.

▶ **Definition 9** (The k-Pairwise-Disjoint-Sets Problem, or k-Pairwise-Orthogonal-Vectors)**.** *Given k lists of sets, each containing N subsets of the same universe U, decide if there are k subsets, one from each list, that are pairwise disjoint.*

The reduction is also by a combination of existing tricks with the key claim.

▶ **Lemma 10.** *For all $2 \leq k \leq n^{\varepsilon}$ where $\varepsilon < 1$, the Set-Cover problem on n elements and m sets of size $n^{o(1)}$ can be reduced to $2^{o(n)}$ instances of the k-Pairwise-Disjoint-Sets problem on k lists of $N \leq 2^{H(1/k)\cdot n+o(n)}$ subsets over a universe U of size n. The reduction runs in time $2^{(1/k+H(1/k))\cdot n+o(n)}$.*

Moreover, in all instances, each subset can be disjoint to at most $2^{(1-1/k)\cdot H(\frac{1}{k-1})\cdot n+o(n)}$ others.

The last sentence of the lemma will be helpful in reducing the sparsity of the k-Clique instances later on.

Proof. The proof is similar to the one of Lemma 8 with a bit more. The additional challenge here is that the end problem does not check that the union of the sets is equal to U, only that they are disjoint. The trick to handle this is yet another guessing step making sure that the sizes of the sets we are choosing sums up to n (before, we only checked that the sum of their *min covers* is equal to ℓ).

First, as before, we reduce to the decision version, and also enumerate k-tuples $\alpha \in [n/k + t]^k$ with sum ℓ, that try to guess the values $f(U_i, \mathcal{S})$ in the partition $U_1, \ldots, U_k$ promised by Claim 6. The additional step here is to also enumerate all k-tuples $\beta \in [n/k+t]^k$ with sum equal to $|U| = n$, that try to guess the values $|U_i|$. The number of β's is also upper bounded by $2^{o(n)}$. From now on fix both an $\alpha = (\alpha_1, \ldots, \alpha_k)$ and a $\beta = (\beta_1, \ldots, \beta_k)$.

Then, as before, we enumerate the collection $\mathcal{P}$ of all subsets of U of size up to $n/k + t$, and compute $f(P, \mathcal{S})$ for each $P \in \mathcal{P}$. These computations can be done with dynamic programming in a total of $2^{H(1/k)\cdot n+o(n)}$ time.

For all $i \in [k]$, the i^{th} list $\mathcal{L}_i$ in the k-Pairwise-Disjoint-Sets instance contains all sets $P \in \mathcal{P}$ whose minimum set-cover size matches the guess in α *and* their size matches the guess in β, that is,

$$\mathcal{L}_i = \{P \in \mathcal{P} \mid f(P, \mathcal{S}) = \alpha_i \text{ and } |U_i| = \beta_i\}.$$

This completes the reduction: at least one of the k-Pairwise-Disjoint-Sets instances (for some tuples α, β) is a yes-instance, if and only if there is a Set-Cover of size ℓ.

For the correctness, observe that for any α, β and any k sets $X_1 \in \mathcal{L}_1, \ldots, X_k \in \mathcal{L}_k$ that are disjoint, we have that $f(U, \mathcal{S}) \leq \sum_{i=1}^{k} f(X_i, \mathcal{S}) = \ell$ and $|X_1 \cup \cdots \cup X_k| = \sum_{i=1}^{k} |X_i| = \sum_i \beta_i = n$. The latter implies that the sets cover the entire U. Thus, if one of the k-Exact-Cover instances is a yes, then $f(U, \mathcal{S}) \leq \ell$. For the other direction, assume that there is a set-cover of size ℓ and consider the partition $U_1, \ldots, U_k$ promised by Claim 6 and let α be the tuple of minimum set-cover sizes for the k sets in that partition, and β be the tuple of their sizes. The instance corresponding to this α, β pair is a yes-instance because each U_i will exist in $\mathcal{L}_i$ and they are disjoint.

Finally, observe that each subset in our instances has size at least n/k, and therefore it can be disjoint to at most

$$\binom{n - n/k}{n/k + t} = 2^{(1-1/k)\cdot H(\frac{1}{k-1})\cdot n+o(n)}$$

other sets in every instance. ◀

2.1 The Reduction to k-SUM

The following version of k-SUM is convenient for reductions.

▶ **Definition 11** (The k-SUM Problem). *Given k lists of N numbers in $[M]$ and a target number $t \in [M]$, decide if there are k numbers, one from each list, that sum to t.*

This "list" or "colored" version is equivalent to the more natural version (where we only have one list and/or where the target is fixed to $t = 0$) up to a k factor in the number of numbers and a k factor in their size.

The reduction to k-SUM is by a simple encoding of the k-Exact-Cover problem.

▶ **Theorem 12.** *For all $2 \leq k \leq n^{\varepsilon}$ where $\varepsilon < 1$, the Set-Cover problem on n elements and m sets of size $n^{o(1)}$ can be reduced to $2^{o(n)}$ instances of the k-SUM problem on k lists of $N = 2^{H(1/k)\cdot n+o(n)}$ integers and a target, where the numbers are $n \cdot \lceil \log_2 k \rceil = O(k \log N)$ bits long. The reduction runs in time $2^{H(1/k)\cdot n+o(n)}$.*

Proof. First, reduce Set-Cover instance to $2^{o(n)}$ instance of k-Exact-Cover, as in Lemma 8, and then reduce each instance to an equivalent k-SUM instance. The k-Exact-Cover instances have k lists of $N \leq 2^{H(1/k)\cdot n+o(n)}$ subsets over the universe $U = [n]$.

The following is a natural mapping g from sets $X \subseteq [n]$ to $(n \cdot \log_2 k)$ bit integers: for all $i \in [n]$, the bit number $(i-1) \cdot \lceil \log_2 k \rceil + 1$ in $g(X)$ is set to 1 if $i \in X$ and to 0 otherwise. All other bits are set to 0, and these are just "buffers" of length $\log_2 k$ that prevent the sum of k numbers to carry over from one "interesting" location to another. Another way to think of the mapping is as writing a number in base k where the i^{th} digit is 1 if $i \in X$ and 0 otherwise.

Let $\mathcal{L}_i$ be the i-th list in the k-Exact-Cover, and we will map it into a list of integers L_i by encoding each set $X \in \mathcal{L}_i$ with the integer $g(X)$. The target sum t for the k-SUM instance is the number $g(U)$, or in other words, the number that is all 1 in base k. This completes the reduction.

For the correctness, one can check that for any k sets $X_1, \ldots, X_k$: $\sum_{i=1}^{k} g(X_i) = g(U)$ if and only if the X_i's are disjoint and $X_1 \cup \cdots \cup X_k = U$. ◀

This proves the k-SUM part of the main theorem, and the lower bound for 3-SUM.

2.2 The Reduction to k-Clique

A reduction from k-SUM on n numbers to $n^{o(1)}$ instances of k-Clique on n nodes is known [4], and it can be used directly to get the desired result. However, the instances generated by that reduction could be dense ($m = n^2$ edges), whereas the following direct reduction from Set-Cover gives the same result but on sparser graphs.

The following version of k-Clique is most convenient for reductions and is equivalent up to a factor k.

▶ **Definition 13** (The k-Clique Problem). *Given a k-partite graph with N nodes in each part and M edges, decide if there is a k-clique (with one node in each part).*

The reduction to k-Clique is almost immediate after one goes though the k-Pairwise-Disjoint-Sets problem.

▶ **Theorem 14.** *For all $2 \leq k \leq n^{\varepsilon}$ where $\varepsilon < 1$, the Set-Cover problem on n elements and m sets of size $n^{o(1)}$ can be reduced to $2^{o(n)}$ instances of the k-Clique problem on k partite graphs of $N = 2^{H(1/k)\cdot n+o(n)}$ nodes and $M = 2^{(H(1/k)+(1-1/k)\cdot H(\frac{1}{k-1}))\cdot n+o(n)}$ edges. The reduction runs in time $2^{H(1/k)\cdot n+o(n)}$.*

Proof. First, reduce Set-Cover instance to $2^{o(n)}$ instance of k-Pairwise-Disjoint-Sets, as in Lemma 10, and then reduce each instance to an equivalent k-Clique instance. The nodes correspond to the sets and two nodes are connected by an edge iff the corresponding sets are disjoint. A k-clique corresponds directly to k pairwise-disjoint sets, and the correctness of the reduction is immediate. The parameters in the statement follow from the parameters in Lemma 10, including the bound on the number of edges which is:

$$M = N \cdot 2^{(1-1/k)\cdot H(\frac{1}{k-1})\cdot n + o(n)}.$$ ◀

This proves the k-Clique part of the main theorem. A corollary of this reduction is an $\Omega(M^{1.01-\varepsilon})$ lower bound for 9-Clique on M edge graphs under the Set-Cover Conjecture.

3 Quantum Algorithms

This section goes in the other direction and discusses a few examples where previously known fine-grained reductions from a "sequential problem" to a "parallelizable core" can be used to obtain quantum speedups. While these arguments are not the simplest way to obtain such quantum speedup, they may still be interesting conceptually.

The first example is from the work of Künnemann et al. [29] who investigated the fine-grained complexity of *one-dimensional dynamic programming*. Their results center around instantiations of a generic Least-Weight Subsequence (LWS) problem where: Given an ordered sequence of n data items, with a succinctly represented function that assigns a weight to each pair, find a (non-contiguous) subsequence of the data points that minimizes the total weight of pairs adjacent in the subsequence. The problem can be instantiated by fixing a weight function, and it can model basic questions such as the coin change problem and finding the longest chain of nested boxes. The authors prove subquadratic-equivalences between these problems and a parallelizable core such as the $(\min, +)$-Convolution problem or vector domination. Thus, classically these problems are unlikely to be solvable in subquadratic time. However, since the parallelizable core problems are easy to solve in $O(n^{2-\varepsilon})$ time for some $\varepsilon > 0$ (and even linear time) for Grover's search, their reductions lead to $O(n^{2-\varepsilon})$ quantum time algorithms for coin change, finding the longest chain of nested boxes, and many other problems as well.

Another example is the RNA Folding problem from bioinformatics and the related problems of Language Edit Distance and Stochastic Context-Free Grammar Parsing. These problems can be solved in cubic time with dynamic programming, and a reduction *a la* Valiant's Parser [35] shows that each of these problems can be solved in the same time as the $(\min, +)$-matrix multiplication problem, or distance product, which is equivalent to All Pairs Shortest Paths and many other problems [41]. In fact, this reduction was recently used by Bringmann et al. [10] to improve the upper bound to $O(n^{2.8244})$ by utilizing a special structure in the $(\min, +)$-matrix multiplication instances produced by the reduction. This is the fastest classical algorithm to date for each of these problems. Can it be improved with a quantum algorithm? Yes, even without any special structure, the $(\min, +)$-matrix multiplication problem is parallelizable and can be solved in $O(n^{2.5})$ time, and via Valiant's reduction, so can these problems. A detailed exposition of these reductions can be found in [10], and related examples on how to incorporate such quantum matrix multiplication algorithm to solve combinatorial problems can be found in [30].

4 Open Questions

The results in this paper lead to the following thoughts.

- From a technical perspective, the reductions in this paper and the quantum algorithms of Ambainis et al. are via an unusual split-and-list approach: it is less efficient, increasing the search space size from 2^n to $2^{\frac{k}{H(1/k)}\cdot n}$, but it can simplify the structure of the space. Is this approach fruitful in other settings, not only when the problems are hard to parallelize? Can it be applied to SAT to get SETH-based lower bounds, that may not be tight, but could have a whole new flavor?
- The efficiency of the reductions does not exactly match that of the quantum algorithm for Set-Cover, since the algorithm benefits from more asymmetry in the splitting[6]. However, it is plausible that better quantum algorithms will lead to better reductions (and lower bounds), and vice versa. Intuitively, the best result one can hope for, without refuting conjectures, is to solve Set-Cover in $\sqrt{2^n}$ quantum time and extract an $n^{k/2}$ (classical) lower bound for k-SUM (which would be tight, for an even k) and for k-Clique (which would be tight if proven for sparse enough graphs). Perhaps this can be achieved by incorporating some of the ideas from the more technically involved works on Set-Cover, e.g. [8, 31, 7, 34], into the currently simple reduction/algorithm. This would be an exciting finding.
- Proving an $\Omega(n^{\varepsilon k})$ lower bound for k-Clique for a fixed $\varepsilon > 0$ (ideally $\varepsilon = 2/3$) under SETH (or even under SeCoCo) is still an important open question. The results here give a qualitatively similar results for k that is a small constant, but one would hope for more. This issue is often considered related to the frequently raised question: Does SETH (or SeCoCo) imply an $(1+\varepsilon)^n$ lower bound for 3-SAT or 100-SAT for any fixed $\varepsilon > 0$?
- The open question of whether SeCoCo implies lower bounds for problems in P was only answered in a limited sense here. Yes, interesting lower bounds for k-Clique and k-SUM can be derived, but the real intent behind the question is: Can SeCoCo give interesting lower bounds in P that other "established" or popular conjectures are incapable of proving? The results here do not qualify, but perhaps they will lead to such results. Can the structure of Set-Cover be utilized further in order to prove similar lower bounds for easier problems? A natural candidate is the $(\min,+)$-Convolution problem which is considered to be the easiest problem without a truly subquadratic time algorithm [16].
- In [5], the quantum speedup over classical dynamic programming for the Graph Bandwidth problem is more substantial than it is for Set-Cover and TSP. The best classical algorithm has $O^*(5^n)$ complexity while the quantum one runs in time $O^*(2.945^n)$, which is quite close to $\sqrt{5^n}$. This suggests that if one reduces the Graph Bandwidth problem to problems in P, e.g. 3-SUM, a tighter relationship could be established, leading to higher "conditional lower bounds". This is more challenging than reducing from Set-Cover due to the more complicated nature of their quantum algorithm.

References

1 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.

[6] By following their algorithm more closely one can get a slightly higher lower bound for the following asymmetric version of k-SUM: Given $k/2$ lists of n_1 numbers and $k/2$ lists of n_2 numbers, where $n_1 = O(n_2^{\varepsilon})$ decide if there is a k-SUM.

2 Amir Abboud, Karl Bringmann, Danny Hermelin, and Dvir Shabtay. SETH-Based Lower Bounds for Subset Sum and Bicriteria Path. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 41–57, 2019.

3 Amir Abboud and Kevin Lewi. Exact Weight Subgraphs and the k-sum Conjecture. In *Proc. of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 1–12, 2013.

4 Amir Abboud, Kevin Lewi, and R. Ryan Williams. Losing Weight by Gaining Edges. In *Proc. of the 22th annual European Symposium on Algorithms (ESA)*, pages 1–12, 2014.

5 Andris Ambainis, Kaspars Balodis, Janis Iraids, Martins Kokainis, Krisjanis Prusis, and Jevgenijs Vihrovs. Quantum Speedups for Exponential-Time Dynamic Programming Algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1783–1793, 2019.

6 Andreas Björklund, Holger Dell, and Thore Husfeldt. The Parity of Set Systems Under Random Restrictions with Applications to Exponential Time Problems. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 231–242, 2015.

7 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *J. Comput. Syst. Sci.*, 87:119–139, 2017. doi:10.1016/j.jcss.2017.03.003.

8 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set Partitioning via Inclusion-Exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009. doi:10.1137/070683933.

9 Andreas Björklund, Petteri Kaski, and Lukasz Kowalik. Constrained Multilinear Detection and Generalized Graph Motifs. *Algorithmica*, 74(2):947–967, 2016. doi:10.1007/s00453-015-9981-1.

10 Karl Bringmann, Fabrizio Grandoni, Barna Saha, and Virginia Vassilevska Williams. Truly Subcubic Algorithms for Language Edit Distance and RNA-Folding via Fast Bounded-Difference Min-Plus Product. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 375–384, 2016.

11 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proc. of the 7th ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 261–270, 2016.

12 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005.

13 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.

14 Lijie Chen and Ruosong Wang. Classical Algorithms from Quantum and Arthur-Merlin Communication Protocols. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 23:1–23:20, 2019.

15 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Transactions on Algorithms*, 12(3):41, 2016.

16 Marek Cygan, Marcin Mucha, Karol Węgrzycki, and Michał Włodarczyk. On problems equivalent to (min,+)-convolution. *ACM Transactions on Algorithms (TALG)*, 15(1):14, 2019.

17 Christoph Dürr and Peter Høyer. A Quantum Algorithm for Finding the Minimum. *CoRR*, quant-ph/9607014, 1996. arXiv:quant-ph/9607014.

18 Friedrich Eisenbrand and Fabrizio Grandoni. On the complexity of fixed parameter clique and dominating set. *Theoretical Computer Science*, 326(1):57–67, 2004.

19 Fedor V Fomin, Dieter Kratsch, and Gerhard J Woeginger. Exact (exponential) algorithms for the dominating set problem. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 245–256. Springer, 2004.

20 Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry*, 5(3):165–185, 1995.

21 François Le Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*, pages 296–303, 2014.

22 Lov K Grover. A fast quantum mechanical algorithm for database search. *arXiv preprint*, 1996. arXiv:quant-ph/9605043.

23 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

24 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

25 Kamil Khadiev. Quantum Dynamic Programming Algorithm for DAGs. Applications for AND-OR DAG Evaluation and DAG's Diameter Search. *CoRR*, abs/1804.09950, 2018. arXiv:1804.09950.

26 Lukasz Kowalik and Juho Lauri. On finding rainbow and colorful paths. *Theor. Comput. Sci.*, 628:110–114, 2016. doi:10.1016/j.tcs.2016.03.017.

27 Robert Krauthgamer and Ohad Trabelsi. The Set Cover Conjecture and Subgraph Isomorphism with a Tree Pattern. *arXiv preprint*, 2017. arXiv:1711.08041.

28 R Krithika, Abhishek Sahu, and Prafullkumar Tale. Dynamic parameterized problems. *Algorithmica*, 80(9):2637–2655, 2018.

29 Marvin Künnemann, Ramamohan Paturi, and Stefan Schneider. On the Fine-Grained Complexity of One-Dimensional Dynamic Programming. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 21:1–21:15, 2017.

30 Aran Nayebi and Virginia Vassilevska Williams. Quantum algorithms for shortest paths problems in structured instances. *CoRR*, abs/1410.6220, 2014. arXiv:1410.6220.

31 Jesper Nederlof. Finding Large Set Covers Faster via the Representation Method. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, pages 69:1–69:15, 2016.

32 J. Nešetřil and S. Poljak. On the complexity of the subgraph problem. *Commentationes Math. Universitatis Carolinae*, 26(2):415–419, 1985.

33 Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1065–1075. SIAM, 2010.

34 Ohad Trabelsi. Nearly Optimal Time Bounds for kPath in Hypergraphs. *CoRR*, abs/1803.04940, 2018. arXiv:1803.04940.

35 Leslie G. Valiant. General Context-Free Recognition in Less than Cubic Time. *J. Comput. Syst. Sci.*, 10(2):308–315, 1975. doi:10.1016/S0022-0000(75)80046-8.

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

37 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity.

38 Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th symposium on Theory of Computing*, pages 887–898. ACM, 2012.

39 Virginia Vassilevska Williams. Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis (Invited Talk). In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 17–29, 2015.

40 Virginia Vassilevska Williams. Some Open Problems in Fine-Grained Complexity. *SIGACT News*, 49(4):29–35, 2018. doi:10.1145/3300150.3300158.

41 Virginia Vassilevska Williams and R. Ryan Williams. Subcubic Equivalences Between Path, Matrix, and Triangle Problems. *J. ACM*, 65(5):27:1–27:38, 2018. doi:10.1145/3186893.