Syntactic Separation of Subset Satisfiability Problems

Eric Allender
Rutgers University, Piscataway, NJ 08854, USA
allender@cs.rutgers.edu

Martín Farach-Colton
Rutgers University, Piscataway, NJ 08854, USA
farach@cs.rutgers.edu

Meng-Tsung Tsai
National Chiao Tung University, Hsinchu, Taiwan
mtsai@cs.nctu.edu.tw

Abstract

Variants of the *Exponential Time Hypothesis* (ETH) have been used to derive lower bounds on the time complexity for certain problems, so that the hardness results match long-standing algorithmic results. In this paper, we consider a syntactically defined class of problems, and give conditions for when problems in this class require strongly exponential time to approximate to within a factor of $(1-\varepsilon)$ for some constant $\varepsilon > 0$, assuming the *Gap Exponential Time Hypothesis* (Gap-ETH), versus when they admit a PTAS. Our class includes a rich set of problems from additive combinatorics, computational geometry, and graph theory. Our hardness results also match the best known algorithmic results for these problems.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Syntactic Class, Exponential Time Hypothesis, APX, PTAS

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.16

Category APPROX

Funding *Eric Allender*: This research was supported in part by NSF grant CCF 1514164.
Martín Farach-Colton: This research was supported in part by NSF grants CCF 1637458, CNS 1408782, IIS 1541613, and NIH grant 1U01CA198952-01.
Meng-Tsung Tsai: This research was supported in part by the Ministry of Science and Technology of Taiwan under contract MOST grant 107-2218-E-009-026-MY3.

1 Introduction

Variants of the *Exponential Time Hypothesis* (ETH) [30, 31] have been used to derive lower bounds that match long-standing upper bounds for several important problems. In particular, the *Strong Exponential Time Hypothesis* (SETH) has been used to study the fine-grained complexity of problems in **P** [46, 47, 1, 14, 8], and the *Gap Exponential Time Hypothesis* (Gap-ETH) [21, 40] was used to study inapproximability [17, 22]. In this paper, we consider a syntactically-defined class of problems, defined below, and give conditions for when problems in this class require strongly exponential time to approximate to within a factor of $(1-\varepsilon)$ for some constant $\varepsilon > 0$, assuming Gap-ETH, versus when they admit a PTAS. Our hardness results also match the best known algorithmic results for these problems. Our class includes a rich set of problems from additive combinatorics, computational geometry, and graph theory.

Let $L = \{\ell_1(\mathbf{x}), \ell_2(\mathbf{x}), \ldots, \ell_k(\mathbf{x})\}$ be a finite set of homogeneous linear functions in $\mathbb{Z}[\mathbf{x}]$ on the same set of variables $\mathbf{x} = (x_1, x_2, \ldots, x_r)$. We define a function $\ell(\mathbf{x})$ to be TRUE at $\mathbf{a}$ if $\ell(\mathbf{a}) \neq 0$. Otherwise, it is FALSE at $\mathbf{a}$. For any set S and integer r, let

$$\mathcal{D}(S, r) := \{(x_1, x_2, \ldots, x_r) \in S^r : x_i \neq x_j \text{ if } i \neq j \text{ for all } i, j \in [1, r]\},$$

that is, the set of permutations over all subsets of S of size r.

SUBSET-CSAT($\mathbf{L}$). Define $L^*(\mathbf{x}) = \bigwedge_{\ell \in L} \ell(\mathbf{x})$. Given a set S of n integers, find a largest $T \subseteq S$ so that for each r-tuple $\mathbf{a} = (a_1, a_2, \ldots, a_r) \in \mathcal{D}(T, r)$, L^* is TRUE at $\mathbf{a}$.[1]

SUBSET-DSAT($\mathbf{L}$). Define $L^+(\mathbf{x}) = \bigvee_{\ell \in L} \ell(\mathbf{x})$. Given a set S of n integers, find a largest $T \subseteq S$ so that for each r-tuple $\mathbf{a} = (a_1, a_2, \ldots, a_r) \in \mathcal{D}(T, r)$, L^+ is TRUE at $\mathbf{a}$.[1]

Many problems can be encoded as one of these two problem types [35, 51, 23, 54, 20, 29, 42, 24, 2, 25], some of which are known to be **APX**-hard, some of which are known to be **NP**-hard, and some of which have no known hardness result. The best known exact algorithms for each of them take strongly exponential time, i.e. $2^{\Omega(n)}$ time. Our main results are Theorem 2 and Theorem 3, below, which can be used to show that all these problems are ***strongly* APX-*hard***, where we define a problem X to be strongly **APX**-hard if there exists a ***size-preserving* PTAS** (SPTAS) reduction from MAX-3SAT to X. A SPTAS reduction is a PTAS reduction whose output has "size"[2] $O(n)$ for any input of size n.

Consequently, given Gap-ETH (Conjecture 1), X cannot be $(1-\delta)$-approximated in subexponential time for a sufficiently small constant $\delta > 0$. To simplify the reductions shown in the subsequent sections, we may restrict the instances of MAX-3SAT as was done in [22]. That is, we make use of the observation in footnote 5 of [40], so that we may assume that there is some constant Δ such that no variable of the formula appears in more than Δ clauses, and hence there are only $O(n)$ clauses, where n is the number of variables.

▶ **Conjecture 1** (Gap-ETH [21, 40]). *There exist constants $\varepsilon, c > 0$ so that no algorithm can distinguish a satisfiable 3SAT formula from those that cannot have more than $(1-\varepsilon)$-fraction of clauses being simultaneously satisfied in 2^{cn} time where n denotes the number of variables in the input instance.*

Our results are:

▶ **Theorem 2.** *Let L be a finite set of homogeneous linear functions whose coefficients are in $\mathbb{Z}$.*

- **(i)** *If L contains only functions with 1 or 2 variables, then* SUBSET-CSAT*(L) admits a* PTAS *and can be exactly solved in $2^{O(n^c)}$ time for some constant $c < 1$.*
- **(ii)** *Otherwise,* SUBSET-CSAT*(L) is strongly* **APX**-*hard.*

We observe here that it is *necessary* to limit our attention to hardness of approximation to within a *constant factor*. The problems we consider can easily be approximated to within a superconstant factor in $2^{o(n)}$ time. Thus strong APX-hardness differs from other hardness of approximation notions (which do not rely on strongly-exponential runtimes), for which it is interesting to consider larger approximation factors. We observe further that not all problems in case (i) are easy to compute exactly, nor are all problems in case (ii) hard to

[1] We assume that $|T| \geq r$ to avoid degenerate cases, which can be identified in $O(n^r)$ time.

[2] The size parameter is determined by problems: typically the number of variables in a formula or the number of nodes in a graph.

approximate to within a constant factor. An example problem for the former is finding a maximum independent set for *c-far unit-disk graphs*, an **NP**-hard problem [41, 56]. We defer the discussions to Appendix A. As for the latter, if all terms in L have positive sign, then a linear-time 1/2-approximation algorithm exists. Moreover, the constant c of case (i) depends on the coefficients of functions in L and the inapproximability constant of case (ii) depends on the number of variables of L.

We need some notions for the next result. We say an $r \times k$ matrix M is ***strongly full rank*** if $k \leq r$ and every $k \times k$ submatrix of M is full rank. Let $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$ be vectors of the same dimensionality, and let $\mathbf{M} = (\mathbf{v_1}|\mathbf{v_2}|\ldots|\mathbf{v_k})$ be the matrix where $M_{ij} = \mathbf{v_j}[i]$. We call $\mathbf{M}$ the ***aggregation*** of $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$. We say a vector space is in ***general position*** if it has a set of basis vectors whose aggregation is strongly full rank.

▶ **Theorem 3.** *Let L be a finite set of homogeneous linear functions whose coefficients are in $\mathbb{Z}$. For each* Subset-DSAT*(L), if the solutions to $\bigvee_{\ell \in L} \ell(\mathbf{x}) =$ False form a vector space in general position and has dimension at least 2 (hence $\mathbf{x}$ is a vector of at least 3 variables), then* Subset-DSAT*(L) is strongly* **APX***-hard.*

Applications. We show how to apply Theorem 2 and Theorem 3 to extend previous hardness results.

(1) Max-General: Given a set S of n points in $\mathbb{R}^2$, find a largest $T \subseteq S$ so that T contains no three distinct colinear points, i.e. finding a largest subset in general position. This problem is known to be **APX**-hard [25].

Here we show how to extend the **APX**-hardness result simply by encoding Max-General as a Subset-CSAT(L) problem for some L. Let $S = \{(a, a^3) : a \in Q\}$ for any set Q of integers. It is known [28] that Q has no three distinct integers that sum to 0 if and only if S has no three distinct colinear points. Therefore,

$$\text{Subset-CSAT}(L_{\text{GP}} := \{\ell(x, y, z) = x + y + z\})$$

can be reduced to Max-General by a linear-time reduction. Together with Theorem 2, one has that Max-General is strongly **APX**-hard.

Note that Subset-CSAT(L_{GP}) can be interpreted as the Max-3SUM problem, and Max-General is a typical example of a Max-3SUM-hard problem. More examples can be found in Section 3.

(2) Max-GolombRuler: Given a set S of n integers in $\mathbb{Z}$, find a largest $T \subseteq S$ so that T has $|T|^2$ distinct pairwise sums. This problem is known to be **NP**-hard to approximate to within an additive constant $c > 0$ [42].

We show how to improve the above inapproximability by encoding Max-GolombRuler as a Subset-CSAT(L) problem for some L. Observe that S has fewer than $|S|^2$ distinct pairwise sums if and only if either there exist four distinct numbers $a, b, c, d \in S$ so that $a + b = c + d$, or there exist three distinct numbers $a, b, c \in S$ so that $a + b = 2c$. To remove the fewest elements from S so that neither of the two cases hold is the same as solving

$$\text{Subset-CSAT}(L_{\text{GR}} := \{\ell_1(x, y, z, w) = x + y - z - w, \ell_2(x, y, z, w) = x + y - 2z\}).$$

Hence, by Theorem 2, Max-GolombRuler is strongly **APX**-hard.

(3) Max-C_3-Free: Given an undirected graph, find a largest node-induced subgraph (in terms of the number of nodes) that contains no cycle of length 3, i.e. a triangle. This problem is known to be **NP**-hard [35].

We show how to extend the **NP**-hardness result by encoding Max-C_3-Free as a Subset-CSAT(L) problem for some L with a restricted input $\bar{S}$. We restrict $\bar{S}$ to be a set such that for every six distinct integers $a_1, a_2, \ldots, a_6 \in \bar{S}$, there are at most two triples summing to 0. We construct an undirected graph $G = (V, E)$ as follows. Initially, $V \leftarrow \emptyset$, $E \leftarrow \emptyset$. For each $a \in \bar{S}$, add v_a to V. For each triple $a, b, c \in \bar{S}$ summing to 0, add edges $\{v_a, v_b\}, \{v_b, v_c\}, \{v_a, v_c\}$ to E. Given this construction, G has a C_3-free node-induced subgraph of k nodes if and only if

$$\text{Subset-CSAT}(L_{\text{C3}} := \{\ell(x, y, z) = x + y + z\}) \text{ with input } \bar{S}$$

has output of size k, which is strongly **APX**-hard as shown in Corollary 15. Hence, Max-C_3-Free is strongly **APX**-hard.

(4) Max-kAP-Free for each $k \geq 3$: Given two integers n and m, decide whether there exists a subset of $S = \{1, 2, \ldots, n\}$ of size at least m so that the subset contains no k distinct integers that form a k-term arithmetic progression. The tally representation of YES-instances of this problem defines a ***sparse language***, which cannot be **NP**-complete unless $\mathbf{P} = \mathbf{NP}$ [45]. An analogous situation also arises in other problems, such as in lattice problems in statistical physics (survey in [57]) or in determining Ramsey numbers (survey in [49]). More generally, if we assume ETH, no optimization problem that has $2^{o(n/\log n)}$ feasible instances can be strongly **APX**-hard. We refer readers to Section 7 for more discussion.

The current best algorithms for Max-kAP-Free [29, 24, 2] rely on branch-and-bound and have to invoke many Max-kAP-Free subproblems, that is, with an arbitrary $S \subseteq \{1, 2, \ldots, n\}$. A hardness result for the subproblem would suggest the limit of solving Max-kAP-Free by branch-and-bound algorithms. We show that it is strongly **APX**-hard.

We encode Max-kAP-Free as

$$\text{Subset-DSAT}(L_{k\text{AP}} := \{\ell_i(x_1, x_2, \ldots, x_k) = x_i - 2x_{i+1} + x_{i+2} : i \in [1, k-2]\})$$

where $|L_{k\text{AP}}| = k - 2$ and set $\mathbf{v_1} = (1, 3, \ldots, 2k-1)$, $\mathbf{v_2} = (2, 4, \ldots, 2k)$ as two basis vectors in the solution space of $\bigvee_{\ell \in L_{k\text{AP}}} \ell(\mathbf{x})$. Because $\mathbf{M} = (\mathbf{v_1}|\mathbf{v_2})$ is strongly full rank, by Theorem 3 we are done.

Our Techniques. We outline the techniques used in the proofs of Theorem 2 and Theorem 3. Both the algorithmic and the hardness results rely on Turán's Theorem [55, 53]. As originally stated, Turán's Theorem [55] said that for every n-node undirected simple graph G, if G has no clique of $r + 1$ nodes for an integer $r \geq 2$, then G has no more than $(1 - 1/r)n^2/2$ edges. In our proofs, when we refer to "Turán's Theorem", we refer to the second formulation of Turán's Theorem [53], that is:

▶ **Theorem 4** (Turán's Theorem [55, 53]). *Every n-node m-edge undirected simple graph has an independent set of size at least $\frac{n^2}{n+2m}$.*

We now describe our approach, and the role Turán's Theorem plays in obtaining our results.

(1) Our Algorithmic Results: Let L_{2^-} be any finite set that contains only homogeneous linear functions with 1 or 2 variables, with coefficients in $\mathbb{Z}$. In Theorem 2, we claim that Subset-CSAT(L_{2^-}) admits a PTAS and can be solved exactly in $2^{O(n^c)}$ time for some constant $c < 1$.

To obtain a PTAS or an exact algorithm for Subset-CSAT(L_{2^-}), we reduce it to finding a maximum independent set for the graph class $\mathcal{G}$ that contains all subgraphs of ***c-nearest neighborhood graphs***, defined in [26, 43], for some constant c. By a generalization of Lipton and Tarjan's algorithm [36], Max Independent Set for $\mathcal{G}$ can be solved efficiently. Lipton and Tarjan show how to approximate the Max Independent Set for planar graphs by exploiting the fact that every planar graph has a node separator of size $O(n^{1/2})$ whose removal partitions the graph into two balanced disconnected subgraphs. Their algorithm can be generalized to any graph class $\mathcal{H}$ that satisfies all the following properties:

- For every graph H in $\mathcal{H}$, any subgraph of H is a graph in $\mathcal{H}$.
- Every h-node graph H in $\mathcal{H}$ has a node separator of size $O(h^c)$ for some constant $c < 1$, whose removal partitions H into two balanced disconnected subgraphs, and the separator can be found in time polynomial in h.
- Every h-node H in $\mathcal{H}$ has an independent set of size $\Omega(h)$.

In Section 2, we will see that $\mathcal{G}$ satisfies all the above properties, and we generalize Lipton and Tarjan's algorithm for any graph class that fulfills all the required properties. We remark that Lipton and Tarjan [36] use the Four Color Theorem [6, 7] to prove the last property for planar graphs. However, since $\mathcal{G}$ contains non-planar graphs, we need to replace the Four Color Theorem with Turán's Theorem to show the last property for $\mathcal{G}$.

(2) Our Hardness Results: If a finite set L of homogeneous linear functions satisfies the condition for case (2) of Theorem 2 (resp. Theorem 3), then Subset-CSAT(L) (resp. Subset-DSAT(L)) is strongly **APX**-hard.

We show the hardness results by a reduction that maps from problem instances of Max Independent Set for sparse large-girth graphs to those of Subset-CSAT(L) or Subset-DSAT(L), so that if the former problem instance has an independent set of size k, then the latter problem instance has an output set of size $f(k)$ for some function f. The existence of the hardness reduction is secured by a probabilistic proof based on the Schwartz-Zippel Lemma [48, 58] as well as some tricks that prohibit the polynomials indicating the probability of desired events from vanishing, that is, that the desired events never happened.

Since all of our claims are applied to deterministic algorithms, we show how to derandomize the probabilistic construction by noting that the construction still works even when the random variables are constant-wise independent. We then use a standard technique to derandomize algorithms that use constant-wise independent random variables [37, 38]. Then, we prove that the reduction is approximation-preserving, again by Turán's Theorem.

We complete the proof by showing that Max Independent Set is strongly **APX**-hard even for sparse large-girth graphs.

Related Work. Our strong **APX**-hardness results apply to Max-rSUM and to some similar problems, which can be viewed as replacing the sum function with more general functions. A similar generalization from rSUM problems [32, 16] to a wider class of problems has also been found useful in studies of the time complexity of rSUM-hard problems in **P**, because the sum function may be not sufficient to encode an rSUM-hard problem but a more general function may [10].

We present a class of optimization problems that are strongly **APX**-hard because of a simple syntactic criterion. In that respect, there is some similarity to prior work on the MaxOnes problem. In [34], syntactic criteria were presented for certain MaxOnes problems, that imply **APX**-hardness. Related topics were also discussed in [9, 33]. Our results are not closely related to [9, 33, 34]; the full version of our paper will compare and contrast our results in more detail.

Paper Organization. In Section 2, we show the algorithmic results. Then, in Section 3, we exhibit our main techniques by proving the strong **APX**-hardness of a simple case Max-3SUM, implying strong **APX**-hardness for a list of Max-3SUM-hard problems via previously-known approximation-preserving reductions from 3SUM-hardness. In Section 4 and Section 5, we generalize the techniques used in Section 3 to prove Theorem 2. We prove Theorem 3 in Section 6, and relate strong **APX**-hardness to the density of languages in Section 7. Then, in Appendix A, we reduce the maximum independent set problem for some intersection graphs to the 2-variate case of Theorem 2 part (ii). In Appendix B, we prove the strong **APX**-hardness of some problems, which are used as source problems for the hardness reductions used in Sections 3 to 6. Finally, we give an inapproximability constant for each intractable problem in our syntactically-defined class in Appendix C.

2 Algorithmic Results

In this section, we prove the algorithmic results stated in Theorem 2, that is, for any finite set L_{2^-} that contains only functions with 1 or 2 variables, Subset-CSAT(L_{2^-}) admits a PTAS and can be exactly solved in $2^{O(n^c)}$ time for some constant $c < 1$. Some of these problems are known to be **NP**-hard; see Appendix A. If L_{2^-} contains a homogeneous linear function with 1 variable, then it suffices to remove 0 from S. Thus, in what follows, we consider Subset-CSAT(L_2) where L_2 is a finite set of homogeneous linear functions with precisely 2 variables. Note that every $\ell(\mathbf{x}) \in L_2$ still has r input variables, but only 2 of the r variables are used.

Given a problem Subset-CSAT(L_2), we construct an undirected simple graph $G_{L_2} = (V, E)$ where $V = \{v_a : a \in S\}$ and

$$E = \{(v_a, v_b) : \ell(\mathbf{x}) = 0 \text{ when } x_i = a, x_j = b \text{ for some } i \neq j \in [1, r], \ell(\mathbf{x}) \in L_2\}.$$

Because L_2 contains only linear functions with 2 variables, finding a maximum independent set for G_{L_2} is equivalent to solving Subset-CSAT(L_2). In what follows, we show that G_{L_2} is a subgraph of some ***c*-nearest neighborhood graph** (Lemma 6), defined below, and show that Max Independent Set for the graph class that consists of subgraphs of c-nearest neighborhood graphs admits a PTAS and can be solved exactly in subexponential time (Theorem 7). We assume that the underlying point set of c-nearest neighborhood graphs (or subgraphs of c-nearest neighborhood graphs) is given. This assumption holds for our case because the c-nearest neighborhood graphs used in our proofs are induced by a point set, and their subgraphs are induced by a subset of the same point set.

▶ **Definition 5** (c-nearest neighborhood graphs [26]). *Given a set P of points in $\mathbb{R}^d$, the c-nearest neighborhood graph of P is a graph $G_P = (V, E)$ whose $V = \{v_a : a \in P\}$ and*

$$E = \{(v_a, v_b) \in V^2 : a \text{ is the } i\text{-th nearest neighbor of } b \text{ for some } i \leq c\},$$

where ties are broken arbitrarily.

▶ **Lemma 6.** *G_{L_2} is a subgraph of some c_{L_2}-nearest neighborhood graph of an n-point set P_{L_2} in $\mathbb{Z}^{d_{L_2}}$ for some constants c_{L_2}, d_{L_2}.*

Proof. We construct the n-point set P_{L_2} by projecting each $v_a \in V(G_{L_2})$ into a point in $\mathbb{Z}^{t+2}$ for some constant $t \geq 0$ as follows. Define

$$\begin{aligned} D_{L_2} &= \{d : d \text{ is prime and } d \text{ divides } c, \text{ where } c \text{ is a coefficient of some } \ell(\mathbf{x}) \in L_2\} \\ &= \{d_1, d_2, \ldots, d_t\} \text{ where } t := |D_{L_2}|. \end{aligned}$$

Since L_2 is a finite set and each $\ell(\mathbf{x}) \in L_2$ has constant coefficients, t is a constant. Given D_{L_2}, for each $v_a \in V(G_{L_2})$ we write a as the unique factorization

$$a = (d_1)^{a_1} \cdots (d_t)^{a_t} (-1)^{a_{t+1}} a_{t+2} \text{ where } a_{t+1} \in \{0,1\}, a_{t+2} > 0 \text{ and } d \nmid a_{t+2} \text{ for all } d \in D_{L_2},$$

based on which we map v_a into the point $p_a := (a_1, a_2, \ldots, a_{t+2})$ for each $v_a \in V(G_{L_2})$.

Since each $\ell(\mathbf{x}) \in L_2$ has constant coefficients with prime divisors in D_{L_2}, if $\ell(\mathbf{x}) = 0$ when we set $x_i = a$ and $x_j = b$ for some $i \neq j \in [1, r]$, then $a_{t+2} = b_{t+2}$ and $a_i - b_i = O(1)$ for each $i \in [1, t+1]$. This yields that for every $(v_a, v_b) \in E(G_{L_2})$ the Euclidean distance between their associated points p_a, p_b is a constant, i.e. $||p_a - p_b||_2$ is a constant.

Let $C = \max_{(v_a,v_b)\in E(G), i\in[1,t+2]} |a_i - b_i|$. Then, for every edge $(v_a, v_b) \in E(G_{L_2})$, p_a is the i-th nearest neighbor of p_b for some $i \leq (2C+1)^{t+2}$, and vice versa. By setting

$$c_{L_2} = (2C+1)^{t+2} \text{ and } d_{L_2} = t+2,$$

we are done. ◀

▶ **Theorem 7.** Max Independent Set *for $\mathcal{H}$ admits a* PTAS *and can be solved exactly in subexponential time, where $\mathcal{H}$ is any graph class that satisfies all the following properties.*

(a) *For every graph H in $\mathcal{H}$, any subgraph of H is a graph in $\mathcal{H}$.*

(b) *Every h-node graph H in $\mathcal{H}$ has a node separator of size $O(h^c)$ for some constant $c < 1$, whose removal partitions H into two balanced disconnected subgraphs, and the separator can be found in time polynomial in h.*

(c) *Every h-node H in $\mathcal{H}$ has an independent set of size $\alpha(H) = \Omega(h)$.*

Proof. We show this by generalizing Lipton and Tarjan's algorithm for Max Independent Set on planar graphs, whose approximate version has the following pseudocode:

Input: an h-node undirected simple graph $H \in \mathcal{H}$

1 Find a node separator C of size $O(h/s^{\varepsilon})$ whose removal partitions H into disconnected subgraphs $H_1, H_2, \ldots, H_t$, each of which has fewer than s nodes, where $s \in (1, h)$ is a function of h and ε is some constant > 0;

2 Compute a maximum independent set I_i in H_i for each $i \in [1, t]$ by exhaustive search;

Output: $I_1 \cup I_2 \cup \cdots \cup I_t$

We need to argue that such a node separator C exists, given the properties of $\mathcal{H}$. We initialize a computation tree $\mathcal{T}$ as follows. Initially, $\mathcal{T}$ has only a root node, associated with H. Then, if there exists a leaf node $a \in \mathcal{T}$ associated with a graph H_a that has more than s nodes, we find a node separator C_a to partition H_a into two balanced disconnected subgraphs H_{a_1} and H_{a_2}. Such a C_a must exist by Properties (a) and (b). Then we link a with two child nodes, a_1 and a_2, whose associated graphs are H_{a_1} and H_{a_2}. Finally, each leaf node in $\mathcal{T}$ has fewer than s nodes. We let the subgraphs associated with leaf nodes in $\mathcal{T}$ be $H_1, H_2, \ldots, H_t$, and let the union of separators found during the construction of $\mathcal{T}$ be C.

By Property (b), C can be constructed in time polynomial in h. The following shows why the size of C is $O(h/s^{\varepsilon})$ for some constant $\varepsilon > 0$. We label each node $a \in \mathcal{T}$ with a height $t(a)$, i.e. the maximum length among all a-to-descendant-leaf paths. Let s_i for $i \geq 1$ be the lower bound on $|H_a|$ for all $a \in \mathcal{T}$ with height i. Since the found separator C_a partitions graph H_a into two balanced subgraphs, both of which have a constant fraction of the nodes in H_a, one can set $s_1 = s$ and $s_i = \Delta s_{i-1}$ for some constant $\Delta > 1$. The total number of nodes in the separators associated with of all nodes in $\mathcal{T}$ with height i is thus

$$\sum_{a \in \mathcal{T}, t(a)=i} |C_a| \leq \delta \left(\sum_{a \in \mathcal{T}, t(a)=i} |H_a|^{1-\varepsilon} \right) \leq \delta \frac{h}{s_i^{\varepsilon}}$$

where δ is a constant determined in Property (b) and the last inequality holds due to Hölder's inequality. Putting it all together, we get

$$|C| = \sum_{a \in \mathcal{T}} |C_a| \leq \delta \sum_{i=1}^{\infty} \frac{h}{(\Delta^{i-1} s)^{\varepsilon}} = O\left(\frac{h}{s^{\varepsilon}}\right).$$

To devise a polynomial-time approximation algorithm, we set $s = \log h$. Thus, the exhaustive search in Step 2 can be done in polynomial time. By the maximality of I_i, we have $\alpha(H) \leq \sum_{i \in t} |I_i| + O(h/\log^{\varepsilon} h)$. Together with $\alpha(H) = \Omega(h)$ due to Property (c), $\sum_{i \in t} |I_i| = (1 - o(1))\alpha(H)$, yielding a $(1 - o(1))$-approximation algorithm.

To devise a subexponential-time exact algorithm, we set $s = h^{\delta}$ for some constant $\delta \in (0, 1)$. Thus, the separator C has size $O(h^{1-\delta\varepsilon})$. Then we try all possible independent sets I_C of C, to be included in the output independent set, in $O(h^2 2^{h^{1-\delta\varepsilon}})$ time. For each I_C, we remove the neighbor nodes of I_C in $H_1, H_2, \ldots, H_t$. Then, we exhaustively search for a maximum independent set in the rest of H_i for each $i \in [1, t]$. These exhaustive searches can be done in $O(h^3 2^{h^{\delta}})$ time. As a result, $I_C \cup I_1 \cup \cdots \cup I_t$ is a maximum independent set for some I_C, and this exact algorithm takes

$$O(h^5 2^{h^{\delta} + h^{1-\delta\varepsilon}})$$

time, which is subexponential for any constant $\delta \in (0, 1)$. ◀

It remains to show that the graph class $\mathcal{G}$ that consists of subgraphs of c-nearest neighborhood subgraphs for some constant c satisfies all the properties listed in Theorem 7. It is clear that Property (a) holds for $\mathcal{G}$. It was shown in [26] that for any h-point set P, G_P has a node separator of size $O(c^{1/d} h^{1-1/d})$ whose removal partitions G_P into two balanced disconnected subgraphs. Moreover, such a node separator can be computed deterministically in $O(ch \log c + h \log h)$ time. For any resulting subgraph H of G_P, whose nodes are associated with a point set $P' \subseteq P$, one can construct the supergraph $G_{P'}$ of H and use the node separator of $G_{P'}$ as the node separator for H. Analogously, the size of the node separator and the running time to find it match the requirement. Thus, Property (b) holds for $\mathcal{G}$. Since any h-node subgraph of c-nearest neighborhood graphs have $O(h)$ edges for any constant c, by Turán's Theorem, Property (c) holds.

3 Hardness of Max-3SUM

In this section, we prove the hardness of Subset-CSAT($L_{3S} := \{\ell(x, y, z) = x + y + z\}$) and defer a proof for the general case in Theorem 2 to Section 4. The proof of the hardness of approximating Subset-CSAT(L_{3S}) will serve as intuition for the general case. The hardness of Subset-CSAT(L_{3S}) implies the hardness of the maximization version of numerous 3SUM-hard problems whose hardness reductions satisfy the following observation.

▶ **Observation 8.** There are many rSUM-hard decision problems $\mathcal{P}$ whose hardness reductions can be directly restated as SPTAS reductions from Max-rSUM to Max-$\mathcal{P}$.

Examples [28, 11, 13] include:

- Max-General: Given $S \subset \mathbb{R}^2$, find a largest $T \subseteq S$ so that T contains no three colinear points. This is one of the applications mentioned in Section 1.
- Max-$\delta\Delta$-Free: Given $S \subset \mathbb{R}^2$, find a largest $T \subseteq S$ so that T contains no three distinct points that form a triangle with area less than δ, for any fixed constant δ.
- Max-3AP-Free: Given $S \subset \mathbb{Z}$, find a largest $T \subseteq S$ so that T contains no three distinct integers that form an arithmetic progression. A more general case Max-kAP-Free for each $k > 3$ needs the hardness results shown in Section 6. We note here that a subset containing no 4-term arithmetic progressions may have 3-term arithmetic progressions, so the hardness of Max-3AP-Free does not immediately imply the hardness of Max-kAP-Free for each $k > 3$, whose proof relies on another system Subset-DSAT(L) for some L whose $|L| = k - 2$.
- Max-3L1P: Given S, a set of lines in $\mathbb{R}^2$, find a largest $T \subseteq S$ so that T contains no three distinct lines that intersect at a point.

NP-hardness. We claim the existence of a polynomial-time many-one reduction from instances of Max Independent Set to instances of Subset-CSAT(L_{3S}). Let n-node m-edge graph $G = (V, E)$ be an instance of Max Independent Set. We need a mapping f from $V \cup E$ to a set S of $n + m$ integers so that G has an independent set of size k iff Subset-CSAT(L_{3S}) with input S has output of size $k + m$. We show that such a set S exists by the probabilistic method [5] and show how to construct S deterministically in time polynomial in n, using derandomization [37, 38].

▶ **Lemma 9.** Subset-CSAT(L_{3S}) *is* **NP***-hard.*

Proof. To implement a mapping $f : V \cup E \to S$, we will use an ***n-order superposable set*** w.r.t. the function $\ell(x, y, z) = x + y + z \in L_{3S}$, which we define as follows. For any set B of n integers $X_1, X_2, \ldots, X_n$, we define the auxiliary set A_ℓ induced by B and ℓ to be

$$\{Y_{ij} : \ell(X_i, X_j, Y_{ij}) = 0, i, j \in [1, n], i < j\}.$$

We say B is an n-order superposable set if A_ℓ contains only integers, $|B \cup A_\ell| = n + \binom{n}{2}$, and for every three distinct integers $a_1, a_2, a_3 \in B \cup A_\ell$, $\ell(a_1, a_2, a_3) = 0$ only if $\{a_1, a_2, a_3\} = \{X_i, X_j, Y_{ij}\}$ for some $i, j \in [1, n], i < j$.

Given the superposable set B, one can realize a mapping $f : V \cup E \to S$, where $f(v_i) = X_i$ for each $v_i \in V$ and $f(\{v_i, v_j\}) = Y_{ij}$ for each $\{v_i, v_j\} \in E$. The following lemma will establish that the image set S and graph G preserve the relation required in the many-one reduction.

▶ **Lemma 10.** *An n-node m-edge graph $G = (V, E)$ has an independent set of size k iff* Subset-CSAT(L_{3S}) *with input $S = f(V \cup E)$ has output of size $k + m$.*

Proof.

(⇒) For each independent set I of G, $I \cup E$ corresponds to a set $T = \{f(a) : a \in I \cup E\}$, a subset of S. Since I is an independent set, for every edge $\{v_i, v_j\}$, the two integers $f(v_i)$, $f(v_j)$ are not simultaneously contained in T. By the definition of a superposable set, T is a valid output for Subset-CSAT(L_{3S}) with input S since it does not contain all three of $f(v_i), f(v_j), f(\{v_i, v_j\})$, for each pair of $i, j \in [1, n]$, $i < j$.

(⇐) Let T be a valid output for Subset-CSAT(L_{3S}) with input S. For each edge $\{v_i, v_j\} \in E$, if both $f(v_i), f(v_j) \in T$, then $f(\{v_i, v_j\}) \notin T$ because T is a valid output. In that case, one can modify T by replacing $f(v_i)$ with $f(\{v_i, v_j\})$. Such a modification does not change the size of T but reduces the number of pairs of $f(v_i), f(v_j)$ in T whose corresponding nodes v_i, v_j are adjacent in G. One can repeat the change until no such $f(v_i), f(v_j)$ pair exists in T. Hence, G has an independent set of size at least k. ◀

Let $R_p(n)$ be a set of n integers $X_1, X_2, \ldots, X_n$ sampled uniformly at random from the universe $U = \mathbb{Z}_p$, for some prime p. In Lemma 11, we prove that, for sufficiently large p, $R_p(n)$ is a superposable set with positive probability. We choose $\mathbb{Z}_p$ to facilitate the derandomization. However, if a set is superposable under $\mathbb{Z}_p$, then it is superposable under $\mathbb{Z}$. After the construction, we use this superposable set under $\mathbb{Z}$.

▶ **Lemma 11.** *The probability that $R_p(n)$ is an n-order superposable set is $1 - O(n^6/p)$.*

Proof. We note that for any pair of different linear polynomials, assigning an integer sampled uniformly at random from a universe U to each variable in the polynomials makes the two polynomials equal in $\mathbb{Z}_p$ with probability $p_{eq} = 1/|U|$, by a simple version of the Schwartz-Zippel Lemma [48, 58]. Here $U = \mathbb{Z}_p$ and $1/|U| = 1/p$. In subsequent sections, we will replace U with another set and will rely more heavily on the Schwartz-Zippel Lemma.

To show $B = R_p(n)$ is superposable, we consider the two probabilities:

$$\Pr\left[|B \cup A_\ell| < n + \binom{n}{2}\right] \leq \sum_{X_i, X_j \in B} p_{eq} + \sum_{X_i \in B, Y_{ij} \in A_\ell} p_{eq} + \sum_{Y_{ij}, Y_{i'j'} \in A_\ell} p_{eq} = O(n^4/p)$$

and

$$\Pr\left[\ell(a_1, a_2, a_3) = 0 \text{ for some } \{a_1, a_2, a_3\} \notin \Gamma\right] \leq \sum_{a_1, a_2, a_3 \in B \cup A_\ell} p_{eq} = O(n^6/p),$$

where $\Gamma := \{\{X_i, X_j, Y_{ij}\} : i, j \in [n], i < j\}$. We are done by applying the Union bound to the two failure probabilities. ◀

Observe that a fully random assignment to the variables of the polynomials is not necessary to make the two polynomials equal with probability as small as $1/p$. Instead, since L_{3S} contains only $\ell(x, y, z) = x + y + z$, if the variables $X_1, X_2, \ldots, X_n$ are assigned 6-wise-independently, the probability p_{eq} is still $1/p$. This observation yields a polynomial-time construction of the superposable set, as follows.

▶ **Lemma 12.** *One can construct an n-order superposable set in time polynomial in n.*

Proof. Exhaustively explore the polynomial-size probability space of 6-wise independence to find the superposable set, which is known to exist [37, 38]. ◀

We complete the proof of Lemma 9 by combining Lemmas 10, 11, and 12. ◀

Strong APX-hardness. In the **NP**-hardness reduction, we have presented a mapping $f : V \cup E \to S$, so that every n-node m-edge graph G has an independent set of size k iff Subset-CSAT(L_{3S}) with input S has output of size $k+m$. In order to demonstrate the strong **APX**-hardness of Subset-CSAT(L_{3S}), it suffices to restrict the Max Independent Set problem to sparse graphs. Thus we will give an SPTAS reduction from Max Independent Set for sparse graphs, which is strongly **APX**-hard (Lemma 24), to Subset-CSAT(L_{3S}).

▶ **Lemma 13.** *There is an* SPTAS *reduction from* Max Independent Set *for sparse graphs to* Subset-CSAT(L_{3S}).

Proof. We use the same reduction as in the proof of Lemma 9, which has the property that independent sets of size k correspond to a solution of Subset-CSAT(L_{3S}) with size $\geq k+m$. But by Turán's Theorem, we have that any sparse graph has an independent set of size $\Omega(m)$. Thus any solution that approximates Subset-CSAT(L_{3S}) to within a factor of $(1-\varepsilon)$ for some constant $\varepsilon > 0$ maps to a solution that approximates Max Independent Set for sparse graphs to within a factor of $(1-O(\varepsilon))$. It is easy to verify that the size of the output of the reduction is linear in the size of the input. ◀

By Lemma 13 and Lemma 24, we get:

▶ **Theorem 14.** Subset-CSAT(L_{3S}) *is strongly* **APX***-hard.*

The above SPTAS reduction is based on the hardness of Max Independent Set for sparse graphs (Lemma 24), which specifies additional structure on the input set S for Subset-CSAT(L_{3S}). Our reduction still works if the graph class is replaced with another graph class $\mathcal{G}$, as long as every n-node graph in $\mathcal{G}$ has $O(n)$ edges and has an independent set of size $\Omega(n)$, and Max Independent Set for $\mathcal{G}$ is strongly **APX**-hard. Such a replacement is useful for proving further hardness results. For example, by Lemma 25 and Turán's Theorem, the source problem of the reduction used in the proof of Theorem 14 can be replaced with Max Independent Set for triangle-free sparse graphs. This yields the following corollary.

▶ **Corollary 15.** Subset-CSAT(L_{3S}) *with input S, in which for every 6 distinct integers, there are at most two triples summing to 0, is strongly* **APX***-hard.*

4 Hardness of Subset-CSAT(L_{S_r})

We generalize the hardness result of Subset-CSAT(L_{3S}) in Section 3 to Subset-CSAT(L_{S_r}) where $L_{S_r} := \{\ell(\mathbf{x}) = \mathbf{c}\cdot\mathbf{x}\}$, and $\ell(\mathbf{x})$ is any linear function with coefficients $\mathbf{c} \in (\mathbb{Z}\setminus\{0\})^r$, for any $r \geq 3$.

Here we extend the definition of n-order superposable set for any r-variate homogeneous linear function $\ell(\mathbf{x})$. Let $t := r-3$. For any set

$$B = \{X_i : i \in [1,n]\} \cup \{X_{ijk} : i,j \in [1,n], i<j, k \in [1,t]\}$$

of $n + t\binom{n}{2}$ integers, we define the auxiliary set A_ℓ induced by B and ℓ to be

$$A_\ell = \{Y_{ij} : \ell(X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}) = 0, i,j \in [1,n], i<j\}.$$

Let $\Gamma = \{\mathcal{S}_{ij} := \{X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}\} : i,j \in [1,n], i<j\}$. We say B is an n-order superposable set if A_ℓ contains only integers, $|B \cup A_\ell| = n + (t+1)\binom{n}{2}$, and for every r distinct integers $a_1, a_2, \ldots, a_r \in B \cup A_\ell$, $\ell(a_1, a_2, \ldots, a_r) = 0$ only if $\{a_1, a_2, \ldots, a_r\} \in \Gamma$.

Let $G = (V, E)$ be a problem instance of Max Independent Set for sparse graphs. Given the superposable set B, we define a mapping $f : V \cup E \to 2^{B \cup A_\ell}$, where $f(v_i) = \{X_i\}$ for each $v_i \in V$ and $f(\{v_i, v_j\}) = \{X_{ij1}, \ldots, X_{ijr}, Y_{ij}\}$ for each $\{v_i, v_j\} \in E$. As in the proof of Lemma 9, if an n-order superposable set B can be constructed in time polynomial in n, then Subset-CSAT(L_{S_r}) is **NP**-hard. Moreover, the hardness-reduction is approximation-preserving for $\ell(\mathbf{x})$ simply by replacing $(1-O(\varepsilon))$ with $(1-O(r\varepsilon))$ in the proof of Lemma 13. Hence, Lemma 16 immediately follows by constructing B in polynomial-time.

▶ **Lemma 16.** Subset-CSAT(L_{S_r}) *is strongly* **APX***-hard.*

Proof. Recall that $\ell(\mathbf{x}) = \mathbf{c} \cdot \mathbf{x} = \sum_{i=1}^{r} c_i x_i$ where $c_i \in \mathbb{Z} \setminus \{0\}$ for each $i \in [1, r]$, and $t := r - 3$. Let $m = \ell cm(c_1, c_2, \ldots, c_r)$. We construct an n-order superposable set B by sampling X_i for each $i \in [1, n]$ and X_{ijk} for $i, j \in [1, n], i < j, k \in [1, t]$ from the universe $U = \mathbb{Z}_p \cap m\mathbb{Z}$ for some prime p. We choose the universe U in this way because no matter what the sampled values of X_i's and X_{ijk}'s are, they make all Y_{ij}'s integral. Before the sampling is performed, each X_i, X_{ijk} in B can be seen as an independent random variable and each Y_{ij} in A_ℓ can be seen as some linear combination of these independent random variables.

We show that the sampled B is an n-order superposable set with positive probability by bounding the probabilities of following bad events. Let E_1 indicate the event that $|B \cup A_\ell| < n + (t+1)\binom{n}{2}$. We claim that $\Pr[E_1] = n^c/(p/m)$ for some constant $c > 0$. To see this, we note that every two distinct random variables $a_1, a_2 \in B \cup A_\ell$ are different linear combinations of the random variables in $\{X_1, X_2, \ldots, X_n\}$. Since $X_1, X_2, \ldots, X_n$ are sampled independently from U, $\Pr[a_1 = a_2] = 1/(p/m)$. Together with the Union bound, the claimed bound for $\Pr[E_1]$ holds.

Let E_2 indicate the event that $\ell(a_1, a_2, \ldots, a_r) = 0$ for some $\{a_1, \ldots, a_r\} \notin \Gamma$. We claim that for every r distinct integers in $B \cup A_\ell$, $\ell(a_1, a_2, \ldots, a_r)$ cannot be a zero function if $(a_1, a_2, \ldots, a_r) \notin \Gamma$. We express a_k for each $k \in [1, r]$ as a linear combination of the random variables in B. To make $\ell(a_1, a_2, \ldots, a_r)$ a zero function, each variable in B either does not appear or appear more than once in a_k's expressions, for all $k \in [1, r]$. This observation implies that if an X_{ijk} in B for some $i, j \in [n], i < j, k \in [1, t]$ appears in the r expressions and $\ell(a_1, a_2, \ldots, a_r)$ is a zero function, then $X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}$ also appear in the r expressions. Hence, in this case, $\{a_1, a_2, \ldots, a_r\} \in \Gamma$.

The remaining case is that X_{ijk} does not appear in any of the r expressions. In this case, to make $\ell(a_1, a_2, \ldots, a_r)$ a zero function, the only possible case happens when $r = 3$ and $\{a_1, a_2, a_3\} = \{Y_{ij}, Y_{jk}, Y_{ik}\}$ for some $i, j, k \in [n]$, $i < j < k$. However,

$$\ell(a_1, a_2, a_3) = c_1 \left(\frac{c_1 X_i + c_2 X_j}{-c_3} \right) + c_2 \left(\frac{c_1 X_j + c_2 X_k}{-c_3} \right) + c_3 \left(\frac{c_1 X_i + c_2 X_k}{-c_3} \right)$$

cannot be a zero function because X_j's coefficient is non-zero, or

$$\ell(a_1, a_2, a_3) = c_1 \left(\frac{c_1 X_i + c_2 X_j}{-c_3} \right) + c_2 \left(\frac{c_1 X_i + c_2 X_k}{-c_3} \right) + c_3 \left(\frac{c_1 X_j + c_2 X_k}{-c_3} \right)$$

cannot be a zero function unless $(c_1, c_2, c_3) = (\Delta, -\Delta, \Delta)$ for some $\Delta \neq 0$, which can be avoided by sorting the variables in ℓ by their coefficients. The same argument works when (a_1, a_2, a_3) equals other permutations of (Y_{ij}, Y_{jk}, Y_{ik}). Hence, the claim is true. There are a polynomial number of non-zero linear functions $\ell(a_1, a_2, \ldots, a_r)$ that cannot be zeroed by the random assigned values of X_i, X_{ijk} for $i, j \in [1, n], i < j, k \in [1, t]$. Therefore the failure rate is $n^c/(p/m)$ for some constant $c > 0$.

Given the bounds on failure probability, the randomly sampled B is an n-order superposable set with positive probability by picking p polynomially large in n. After a derandomization step similar to that in Lemma 12, we have B constructed in deterministic polynomial time. ◀

5 Hardness of Subset-CSAT(L_r)

We extend the hardness result of Subset-CSAT(L_{S_r}) to

$$\text{Subset-CSAT}(L_r := \{\ell_1(\mathbf{x}), \ell_2(\mathbf{x}), \ldots, \ell_k(\mathbf{x})\}),$$

where $\ell_i(\mathbf{x}) \in \mathbb{Z}[\mathbf{x}]$ for each $i \in [1,k]$, $\mathbf{x} = (x_1, x_2, \ldots, x_r)$, and at least one $\ell_i(\mathbf{x})$ uses at least 3 of the r input variables. Showing the strong **APX**-hardness of Subset-CSAT(L_r) proves Theorem 2.

We begin by defining a ***canonical representation*** for the $\ell_i(\mathbf{x})$'s. Observe that

$$\text{Subset-CSAT}(\{\ell_1(x,y,z,w) = x + y - z, \ell_2(x,y,z,w) = y + w - x\})$$

equals Subset-CSAT($\{\ell(x,y,z,w) = x+y-z\}$), which also equals Subset-CSAT($\{\ell(x,y,z) = \delta(x+y-z)\}$) for any constant $\delta \neq 0$, because in the definition of Subset-CSAT(L), we assume that the output has size at least r. Let Coef($\ell_i(\mathbf{x})$) be the multi-set of coefficients in $\ell_i(\mathbf{x})$. We say that $\ell_i(\mathbf{x})$ and $\ell_j(\mathbf{x})$ are in the same equivalence class if Coef($\ell_i(\mathbf{x})$) = $\{\delta c : c \in \text{Coef}(\ell_j(\mathbf{x}))\}$ for some non-zero constant δ. Thus, we can remove redundant functions in L, if any, by removing $\ell_i(\mathbf{x})$ from L if $\ell_i(\mathbf{x})$ and $\ell_j(\mathbf{x})$ are from the same equivalence class, for some $j < i$. Given the succinct representation of L, let $\ell_*(\mathbf{x})$ be the $\ell_i(\mathbf{x})$ in L that has the largest number of variables. If there is a tie, then pick any of them.

▶ **Lemma 17.** *Consider an r-variate homogeneous linear function $\ell_*(\mathbf{x})$ where $r \geq 3$, and an r'-variate homogeneous linear function $\ell(\mathbf{x})$ where $r' \leq r$. Let $t := r - 3$. Then for any constant $\varepsilon > 0$ there exists a randomized algorithm that constructs with probability at least $1-\varepsilon$ an n-order superposable set $B = \{X_i : i \in [1,n]\} \cup \{X_{ijk} : i,j \in [n], i < j, k \in [1,t]\}$ and the auxiliary set $A_{\ell_*} = \{Y_{ij} : \ell(X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}) = 0, i,j \in [1,n], i < j\}$ induced by B and ℓ_*, so that for every r distinct integers in $B \cup A_{\ell_*}$, $\ell(a_1, a_2, \ldots, a_{r'}) = 0$ only if either of the following two cases applies:*

- *$r' = r$ and $\{a_1, a_2, \ldots, a_{r'}\} = \{X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}\}$,*
- *$r' = r = 3$ and $\{a_1, a_2, a_3\} = \{Y_{s_1s_2}, Y_{s_2s_3}, Y_{s_1s_3}\}$ for some $1 \leq s_1 < s_2 < s_3 \leq n$.*

Proof. Set each element in B to be a random variable, and therefore each element in A_{ℓ_*} is a linear combination of $r-1$ random variables and none of them in the linear combination has coefficient 0. To make $\ell(\mathbf{x})$ a zero function by setting r' distinct variables from $(B \cup A_{\ell_*})$, it is necessary that each variable in B either does not appear among any of the r' picked variables or appears in at least two of them, noting that X_i is considered to "appear" in X_i and Y_{ij} for any $j \in [1,n]$. There are two cases. If X_{ijk} for some $i < j, i,j \in [n], k \in [1,t]$ is one of the r' picked variables, then $\ell(\mathbf{x})$ is zero only if the r' picked variables are exactly $X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}$. Otherwise, for every $i < j$, $i,j \in [n]$, $k \in [1,t]$, X_{ijk} is *not* picked as one of the r' variables. In this case, to make $\ell(\mathbf{x})$ zero it is necessary that $t = 0$ (or equivalently $r = 3$), $r' = r$, and the r' picked variables are either X_i, X_j, Y_{ij} for some $i,j \in [n]$ or $Y_{s_1s_2}, Y_{s_2s_3}, Y_{s_3s_1}$ for some $1 \leq s_1 < s_2 < s_3 \leq n$.

Therefore, if we let $B = R_p(n)$, then the probability that $\ell(a_1, a_2, \ldots, a_{r'}) = 0$ for some $a_1, a_2, \ldots, a_{r'}$ other than the two given ones (the only cases that may make $\ell(\mathbf{x})$ as a zero function) is $1/p$. By the Union bound over all possible r' distinct values from $B \cup A_{\ell_*}$ in which there are $O(n^2)$ elements, we get the success probability of our random assignment is at least $1 - n^{2r'}/p$. Picking a sufficiently large p completes the proof. ◀

We apply Lemma 17 to each $\ell(\mathbf{x})$ in the succinct representation of L, take the Union bound over the failure probabilities, by the aforementioned derandomization step, we get:

▶ **Lemma 18.** *For any set $L_r(\mathbf{x})$ of r-variate homogeneous linear functions, if the function $\ell_*(\mathbf{x})$ in $L(\mathbf{x})$ that uses the largest number of variables is r'-variate for some $r' \geq 3$, let $t := r' - 3$, there exists a deterministic polynomial-time algorithm that can construct an n-order superposable set $B = \{X_i : i \in [1,n]\} \cup \{X_{ijk} : i,j \in [n], i < j, k \in [1,t]\}$ w.r.t. ℓ_* and the auxiliary set $A_{\ell_*} = \{Y_{ij} : \ell_*(X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij})\}$ induced by B and ℓ_* so that $\bigwedge_{\ell \in L} \ell(a_1, a_2, \ldots, a_r)$ evaluates to* False *only if either of the two following cases applies:*

- $\{a_1, a_2, \ldots, a_r\} = \{X_i, X_j, X_{ij1}, \ldots, X_{ijt}, Y_{ij}\}$,
- $r = 3$ *and* $\{a_1, a_2, a_3\} = \{X_{ij}, X_{jk}, X_{ik}\}$.

Proof of Theorem 2. Given Lemma 18, one can reuse the many-one reduction mentioned previously, but restrict the input graph to be triangle-free (i.e. girth ≥ 3), so that a_1, a_2, a_3 in the second case of Lemma 18 cannot simultaneously appear in the set S, i.e. the input of the reduction target. By Lemma 25, the maximum independent set problem for sparse graphs of girth ≥ 3 is strongly **APX**-hard, implying that Subset-CSAT(L_r) is strongly **APX**-hard. ◀

6 Hardness of Subset-DSAT($L_\vee$)

In this section, we will show the strong **APX**-hardness of

$$\textsc{Subset-DSAT}(L_\vee := \{\ell_1(\mathbf{x}), \ell_2(\mathbf{x}), \ldots, \ell_k(\mathbf{x})\}),$$

where $\ell_i(\mathbf{x}) \in \mathbb{Z}[\mathbf{x}]$ for each $i \in [1,k]$, $\mathbf{x} = (x_1, x_2, \ldots, x_r)$, and the solutions to $\bigvee_{\ell \in L} \ell(\mathbf{x}) =$ False form a vector space in general position and has dimension d at least 2. That is, we prove Theorem 3.

To prove Theorem 3 for $d = r - 1$, one can use the proof of Theorem 2. For $d < r - 1$ in general, the number of dependent random variables induced by the superposable set B is no longer 1, thus requiring the solution set of to be in general position. We need to modify the definition of the superposable set w.r.t. such a $\chi_{L_\vee}(\mathbf{x})$, as described below.

Proof of Theorem 3. For any n-node, m-edge graph $G = (V, E)$ that has $m = O(n)$ and girth at least $r + 1$, we construct a set B of independent random variables and an auxiliary set $A_{\chi_{L_\vee}}$ so that

$$B = \{X_i : i \in V\} \cup \{X_{ijk} : (i,j) \in E, i < j, k \in [1, d-2]\}, \text{ and}$$

$$A_{\chi_{L_\vee}} = \{Y_{ij1}, \ldots, Y_{ij(r-d)} : \chi_{L_\vee}\left(X_i, X_j, X_{ij1}, \ldots, X_{ij(d-2)}, Y_{ij1}, \ldots, Y_{ij(r-d)}\right) = 0, \\ (i,j) \in E, i < j\},$$

where the solution space of $\chi_{L_\vee}(\mathbf{x}) = 0$ is in general position and has dimension $d \geq 2$. Hence, for every $(i,j) \in E, i < j$, $\left(Y_{ij1}, Y_{ij2}, \ldots, Y_{ij(r-d)}\right)$ is unique.

Here we define the Y_{ijk} explicitly. Let $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_d}$ be a set of basis vectors (column vectors) in $\mathbb{Z}^r$ of the solution set of $\chi_{L_\vee}(\mathbf{x}) = 0$. Let $\mathbf{A}$ be the aggregation of $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_d}$ where $\mathbf{A} = (\mathbf{v_1}|\mathbf{v_2}|\cdots|\mathbf{v_d})$. Let $\mathbf{Q}$ be the square matrix composed of the upper d rows of $\mathbf{A}$. By the definition of general position, $\mathbf{A}$ is strongly full rank, $\mathbf{Q}$ is full rank, and thus $\mathbf{z}$ is uniquely defined by

$$\mathbf{Qz} = \begin{bmatrix} X_i \\ X_j \\ X_{ij1} \\ \vdots \\ X_{ij(d-2)} \end{bmatrix}, \text{ and we set } \mathbf{Az} = \mathbf{A}\left(\mathbf{Q}^{-1}\begin{bmatrix} X_i \\ X_j \\ X_{ij1} \\ \vdots \\ X_{ij(d-2)} \end{bmatrix}\right) = \begin{bmatrix} X_i \\ X_j \\ X_{ij1} \\ \vdots \\ X_{ij(d-2)} \\ Y_{ij1} \\ \vdots \\ Y_{ij(r-d)} \end{bmatrix}.$$

Thus, each of $Y_{ij1}, \ldots, Y_{ij(r-d)}$ is a nontrivial linear combination of $X_i, X_j, X_{ij1}, \ldots, X_{ij(d-2)}$. Note that $\mathbf{AQ}^{-1}$ is also strongly full rank, yielding that any nontrivial linear combination of d variables from the set $\{X_i, X_j, X_{ij1}, \ldots, X_{ij(d-2)}, Y_{ij1}, \ldots, Y_{ij(r-d)}\}$ cannot be a zero function. We are ready to prove that B is superposable w.r.t. E, that is:

▶ **Lemma 19.** *For any distinct $a_1, a_2, \ldots, a_r \in B \cup A_{\chi_{L_\vee}}$, $\chi_{L_\vee}(a_1, a_2, \ldots, a_r)$ is a zero function only if $\{a_1, a_2, \ldots, a_r\} = \{X_i, X_j, X_{ij1}, \ldots, X_{ij(d-2)}, Y_{ij1}, \ldots, Y_{ij(r-d)}\}$ for some $(i, j) \in E, i < j$.*

Proof. If $\{a_1, a_2, \ldots, a_r\} \subseteq \{X_i : i \in [1, n]\}$, then $\chi_{L_\vee}(a_1, a_2, \ldots, a_r)$ cannot be a zero function because the X_i's are independent variables and each X_i appears at most once in any linear function $\ell_j(\mathbf{x})$ that comprises $\chi_{L_\vee}$. Thus, to zero $\chi_{L_\vee}(a_1, a_2, \ldots, a_r)$ we may assume that

$$a_p \in \mathcal{S}_{ij} := \{X_{ij1}, \ldots, X_{ij(d-2)}, Y_{ij1}, \ldots, Y_{ij(r-d)}\} \text{ for some } p \in [1, r], (i, j) \in E, i < j.$$

Say a_p appears in some homogeneous linear function $\ell_q(\mathbf{x})$ that comprises $\chi_{L_\vee}$. In order to make $\chi_{L_\vee}(a_1, \ldots, a_r)$ a zero function, one must make $\ell_q(\mathbf{x})$ a zero function. We disprove the possibility of making $\ell_q(\mathbf{x})$ zeroed as follows. If $\ell_q(\mathbf{x})$ picks $\geq d$ variables from $\mathcal{S}_{ij}$, then each of $a_1, \ldots, a_r$ can be represented by a linear combination of random variables in $\mathcal{S}_{ij}$. In other words, $\{a_1, \ldots, a_r\} \subseteq (\mathcal{S}_{ij} \cup \{X_i, X_j\})$ because $d \geq 2$. If $\ell_q(\mathbf{x})$ picks $d-1$ variables from $\mathcal{S}_{ij}$, then to make $\ell_q(\mathbf{x})$ zeroed, $\ell_q(\mathbf{x})$ needs to pick two variables a_w and a_z where a_w is from $\mathcal{S}_{ik} \cup \{X_i\}$ and a_z is from $\mathcal{S}_{j\ell} \cup \{X_j\}$. Note that $k \neq \ell$ because G has girth $r + 1 \geq d + 2 \geq 4$. This would lead to a contradiction since if we solve the system by the $d-1$ variables from $\mathcal{S}_{ij}$ as well as a_w, then a_z can be represented by linear combination of variables from $\mathcal{S}_{ij} \cup \mathcal{S}_{ik} \cup \{X_i\}$, contradicting that $a_z \in \mathcal{S}_{j\ell}$, $\ell \neq k$, and $d \geq 2$. If $\ell_q(\mathbf{x})$ picks $\leq d-2$ variables from $\mathcal{S}_{ij}$, since the rest of variables can be partitioned into subsets, each of which sum to a multiple of X_i, or a multiple of X_j, but not a linear combination of X_i and X_j due to G having girth at least $r+1$, therefore $\ell_q(\mathbf{x})$ cannot be zeroed since this effectively picks $\leq d$ variables from $\mathcal{S}_{ij} \cup \{X_i, X_j\}$. ◀

Lastly, the exact construction of the superposable set is similar to that in Theorems 2. By Lemma 19 and the Swartz-Zippel Lemma, we know that sampling $\{X_i : i \in [1, n]\} \cup \{X_{ijk} : (i, j) \in E, i < j, k \in [1, d-2]\}$ uniformly at random from $(\det(\mathbf{Q})\mathbb{Z})^{n+(d-2)m}$ yields a superposable set with positive probability. We pick every X_i and X_{ijk} as multiples of $\mathbf{det}(\mathbf{Q})$ to ensure that all dependent variables Y_{ijk}'s are in $\mathbb{Z}$. Then, after derandomization using techniques for constant-wise independence, the construction takes time polynomial in n. Setting $g = r+1$, we know that Subset-DSAT$(L_\vee)$ is strongly **APX**-hard by Lemma 25. ◀

An implication of Theorem 3 is the strong **APX**-hardness of finding the maximum-cardinality k-term AP-free subset S for any fixed $k \geq 3$, noting that S may contain elements that form an i-term arithmetic progression for $i < k$ but not $i \geq k$. This problem can be encoded as

$$\textsc{Subset-DSAT}(L_{k\text{AP}} := \{\ell_i(x_1, x_2, \ldots, x_k) = x_i - 2x_{i+1} + x_{i+2} : i \in [1, k-2]\}),$$

and the solution set of $\sum_{\ell(\mathbf{x}) \in K_{k\text{AP}}} \ell^2(\mathbf{x}) = 0$ contains the plane

$$(x_1, x_2, \ldots, x_k) = \alpha(1, 3, \ldots, 2k-1) + \beta(2, 4, \ldots, 2k) \text{ for constant } \alpha, \beta \in \mathbb{R}.$$

Therefore,

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ \vdots & \vdots \\ 2k-1 & 2k \end{bmatrix} \text{ in which every } 2 \times 2 \text{ submatrix } \begin{bmatrix} 2i-1 & 2j-1 \\ 2i & 2j \end{bmatrix}$$

is full rank. By Theorem 3, we get:

▶ **Corollary 20.** *Finding a maximum-cardinality k-term AP-free subset of a given integral set S for any fixed $k \geq 3$ is strongly* **APX***-hard.*

Sharpness of $d \geq 2$. Not every problem in the class $\textsc{Subset-DSAT}(L)$ is hard to approximate. If the solution set of $\chi(\mathbf{x}) = \sum_{\ell(\mathbf{x}) \in L} \ell^2(\mathbf{x})$ is a point[3], then it suffices to remove an integer in the set S that coincides with the coordinate of the point. If it is a line, for example $\alpha(1, 2, 4, 8)$ for $\alpha \in \mathbb{R}$, then a greedy algorithm can solve this case in P by removing the last coordinate for every tuple of 4 integers that are multiples of (1, 2, 4, 8).

7 A Sparsity Bound, Assuming ETH

Define a ***sparse language*** as one where there are $n^{O(1)}$ length-n Yes-instances. Mahaney's theorem [39, 45] states that if $\mathbf{P} \neq \mathbf{NP}$, then there is no **NP**-hard sparse language. Buhrman and Hitchcock prove a stronger (optimal) bound from a stronger hypothesis [15]: if **PH** doesn't collapse, then there is no **NP**-hard set with $2^{n^{o(1)}}$ length-n Yes-instances. If one assumes ETH, then an even stronger bound holds for strongly **APX**-hard problems.

▶ **Theorem 21.** *If X is an optimization problem such that X has $2^{o(n/\log n)}$ strings of length n, then X cannot be strongly* **APX***-hard unless* ETH *fails.*

Proof. This proof is based on the proof of Mahaney's theorem presented in [45]. Assume that X is strongly **APX**-hard, and we will present a subexponential-time algorithm to solve 3SAT. Let

$$L = \{(\varphi, a) : \varphi \text{ has a satisfying assignment } a' \text{ lexicographically smaller than } a\}.$$

L is in nondeterministic linear time, and hence by [19] there is a reduction g of L to 3SAT such that, on input (φ, a) of size n, $g(\varphi, a)$ is a 3CNF formula with $O(n \log n)$ variables, each of which appears in $O(1)$ clauses.

[3] $\mathbf{0}$ must be a solution of $\chi(\mathbf{x})$ because the $\ell_i(\mathbf{x})$'s are homogeneous.

Since X is strongly **APX**-hard, there is a function f such that, for any 3CNF formula ψ of size n, $f(\psi)$ has size $O(n)$, where f yields the SPTAS reduction from Max-3SAT to X.

Consider any satisfiable formula φ with n variables; let a_φ be its lexicographically smallest satisfying assignment. Hence, $(\varphi, a) \in L$ if and only if $a \geq a_\varphi$, lexicographically.

We now present an algorithm for finding a_φ that runs in subexponential time. (If the algorithm fails to find a satisfying assignment, then φ is not satisfiable.) We start with a search space of size 2^n. Let $C = 2^{o(n)}$ be greater than the number of strings in X of length $m = O(n \log n)$, where the output of the reduction $g(\varphi, a)$ has length m. Find C assignments $a_1, \ldots, a_C$ that are evenly spaced among the current search space, and compute $z_i = f(g(\varphi, a_i))$ for $1 \leq i \leq C$.

If there are $i < j$ such that $z_i = z_j$, then $g(\varphi, a_i)$ is in 3SAT iff $g(\varphi, a_j)$ is, and thus a_φ does not lie in the segment $(a_i, a_j]$, and thus we can reduce the size of our search space by a factor of $1/C$.

Otherwise, there are C distinct elements z_i of the form $f(g(\varphi, a_i))$, which is greater than the number of relevant elements of X that can be in the range of f. Thus at least one of the formulae $g(\varphi, a_i)$ must be unsatisfiable, since f maps it to an infeasible instance of X. But if any formula $g(\varphi, a_i)$ is unsatisfiable, it follows that $g(\varphi, a_1)$ is unsatisfiable, and hence a_φ does not lie in the segment $[0^n, a_1]$, and thus again we can reduce the size of our search space by a factor of $1/C$.

We now repeat the process with a new set of C checkpoints. As in [45], the bookkeeping that is necessary to keep track of the current search space and to compute the new checkpoints does not get too complicated, and after a small number of iterations the entire search space is of size at most C, at which point we can check directly in subexponential time if any of the remaining assignments satisfies φ.

This algorithm can thus determine if φ is satisfiable or not, which is at least as hard as solving 3SAT. ◀

References

1. Amir Abboud and Virginia Vassilevska Williams. Popular Conjectures Imply Strong Lower Bounds for Dynamic Problems. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS, pages 434–443, Washington, DC, USA, 2014. IEEE Computer Society. doi:10.1109/FOCS.2014.53.
2. Tanbir Ahmed, Janusz Dybizbanski, and Hunter S. Snevily. Unique Sequences Containing No k-Term Arithmetic Progressions. *Electr. J. Comb.*, 20(4):P29, 2013.
3. M. Ajtai, P. Erdös, J. Komlós, and E. Szemerédi. On Turán's theorem for sparse graphs. *Combinatorica*, 1(4):313–317, 1981.
4. M. Ajtai, J. Komlós, and E. Szemerédi. A Dense Infinite Sidon Sequence. *European Journal of Combinatorics*, 2(1):1–11, 1981.
5. Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, 1992.
6. K. Appel and W. Haken. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, September 1977.
7. K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. Part II: Reducibility. *Illinois Journal of Mathematics*, 21(3):491–567, September 1977.
8. Arturs Backurs and Piotr Indyk. Edit Distance Cannot Be Computed in Strongly Subquadratic Time (Unless SETH is False). *SIAM J. Comput.*, 47(3):1087–1097, 2018. doi:10.1137/15M1053128.
9. Brenda S. Baker. Approximation Algorithms for NP-complete Problems on Planar Graphs. *J. ACM*, 41(1):153–180, January 1994.

10 Luis Barba, Jean Cardinal, John Iacono, Stefan Langerman, Aurélien Ooms, and Noam Solomon. Subquadratic Algorithms for Algebraic 3SUM. *Discrete & Computational Geometry*, 61(4):698–734, 2019. doi:10.1007/s00454-018-0040-y.

11 Gill Barequet and Sariel Har-Peled. Polygon Containment and Translational Min-Hausdorff-Distance Between Segment Sets are 3Sum-hard. *Int. J. Comput. Geometry Appl.*, 11(4):465–474, 2001.

12 Piotr Berman and Marek Karpinski. On Some Tighter Inapproximability Results (Extended Abstract). In *26th International Colloquium in Automata, Languages and Programming (ICALP)*, pages 200–209, 1999.

13 Prosenjit Bose and Stefan Langerman. Weighted Ham-Sandwich Cuts. In *Discrete and Computational Geometry, Japanese Conference, JCDCG, Revised Selected Papers*, pages 48–53, 2004.

14 Karl Bringmann. Why Walking the Dog Takes Time: Frechet Distance Has No Strongly Subquadratic Algorithms Unless SETH Fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 661–670, 2014.

15 Harry Buhrman and John M. Hitchcock. NP-hard sets are exponentially dense unless coNP $\subseteq$ NP/poly. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, (CCC)*, pages 1–7. IEEE Computer Society, 2008. doi:10.1109/CCC.2008.21.

16 Jean Cardinal, John Iacono, and Aurélien Ooms. Solving k-SUM Using Few Linear Queries. In *24th Annual European Symposium on Algorithms, ESA*, pages 25:1–25:17, 2016.

17 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 743–754, 2017.

18 Miroslav Chlebík and Janka Chlebíková. Approximation Hardness for Small Occurrence Instances of NP-hard Problems. In *5th Italian Conference on Algorithms and Complexity (CIAC)*, pages 152–164. Springer, 2003.

19 Stephen A. Cook. Short Propositional Formulas Represent Nondeterministic Computations. *Inf. Process. Lett.*, 26(5):269–270, 1988. doi:10.1016/0020-0190(88)90152-4.

20 Carlos Cotta, Iván Dotú, Antonio J. Fernández, and Pascal Van Hentenryck. A Memetic Approach to Golomb Rulers. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature*, PPSN, pages 252–261, Berlin, Heidelberg, 2006. Springer-Verlag.

21 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016.

22 Irit Dinur and Pasin Manurangsi. ETH-hardness of approximating 2-CSPs and directed Steiner network. In *9th Innovations in Theoretical Computer Science Conference, ITCS*, pages 36:1–36:20, 2018.

23 Apostolos Dollas, William T. Rankin, and David Mccracken. New Algorithms for Golomb Ruler Derivation and Proof of the 19 Mark Ruler. *IEEE Transactions on Information Theory*, 44:379–382, 1998.

24 J. Dybizbański. Sequences containing no 3-term arithmetic progressions. *Elec. J. of Comb.*, 19(2):15–19, 2012.

25 David Eppstein. *Forbidden Configurations in Discrete Geometry*. Cambridge University Press, 2018.

26 David Eppstein, Gary L. Miller, and Shang-Hua Teng. A Deterministic Linear Time Algorithm for Geometric Separators and its Applications. *Fundam. Inform.*, 22(4):309–329, 1995. doi:10.3233/FI-1995-2241.

27 S. Fajtlowicz. The independence ratio for cubic graphs. In *8th Southeastern Conf. on Combinatorics, Graph Theory, and Computing*, pages 273–277. LSU, 1977.

28 Anka Gajentaan and Mark H. Overmars. On a Class of $O(N^2)$ Problems in Computational Geometry. *Comput. Geom. Theory Appl.*, 5(3):165–185, October 1995.

29 William I. Gasarch, James Glenn, and Clyde P. Kruskal. Finding large 3-free sets I: The small n case. *J. Comput. Syst. Sci.*, 74(4):628–655, 2008.

30 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

31 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, December 2001. doi:10.1006/jcss.2001.1774.

32 Allan Grønlund Jørgensen and Seth Pettie. Threesomes, Degenerates, and Love Triangles. *J. ACM*, 65(4):22:1–22:25, 2018. doi:10.1145/3185378.

33 Sanjeev Khanna and Rajeev Motwani. Towards a Syntactic Characterization of PTAS. In *28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 329–337. ACM, 1996.

34 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The Approximability of Constraint Satisfaction Problems. *SIAM J. Comput.*, 30(6):1863–1920, December 2001.

35 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

36 Richard J. Lipton and Robert Endre Tarjan. Applications of a Planar Separator Theorem. *SIAM J. Comput.*, 9(3):615–627, 1980. doi:10.1137/0209046.

37 Michael Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986.

38 Michael Luby and Avi Wigderson. Pairwise Independence and Derandomization. *Found. Trends Theor. Comput. Sci.*, 1(4):237–301, August 2006.

39 Stephen R. Mahaney. Sparse Complete Sets of NP: Solution of a Conjecture of Berman and Hartmanis. *J. Comput. Syst. Sci.*, 25(2):130–143, 1982. doi:10.1016/0022-0000(82)90002-2.

40 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 78:1–78:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.

41 Nimrod Megiddo and Kenneth J. Supowit. On the Complexity of Some Common Geometric Location Problems. *SIAM J. Comput.*, 13(1):182–196, 1984.

42 Christophe Meyer and Periklis A. Papakonstantinou. On the Complexity of Constructing Golomb Rulers. *Discrete Appl. Math.*, 157(4):738–748, February 2009.

43 Gary L. Miller, Shang-Hua Teng, William Thurston, and Stephen A. Vavasis. Separators for Sphere-packings and Nearest Neighbor Graphs. *J. ACM*, 44(1):1–29, January 1997. doi:10.1145/256292.256294.

44 Owen J. Murphy. Computing Independent Sets in Graphs with Large Girth. *Discrete Appl. Math.*, 35(2):167–170, January 1992.

45 Mitsunori Ogiwara and Osamu Watanabe. On Polynomial-Time Bounded Truth-Table Reducibility of NP Sets to Sparse Sets. *SIAM J. Comput.*, 20(3):471–483, 1991.

46 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*, SODA, pages 1065–1075, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

47 Liam Roditty and Virginia Vassilevska Williams. Fast Approximation Algorithms for the Diameter and Radius of Sparse Graphs. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC, pages 515–524, New York, NY, USA, 2013. ACM. doi:10.1145/2488608.2488673.

48 J. T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, October 1980.

49 Pascal Schweitzer. *Problems of unknown complexity – Graph isomorphism and Ramsey theoretic numbers*. PhD thesis, Universität des Saarlandes, 2009.

50 James B. Shearer. A note on the independence number of triangle-free graphs. *Discrete Mathematics*, 46(1):83–87, 1983.

51 Stephen W. Soliday, Abdollah Homaifar, and Gary L. Lebby. Genetic Algorithm Approach to the Search for Golomb Rulers. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 528–535, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. URL: `http://dl.acm.org/citation.cfm?id=645514.658082`.

52 William Staton. Some Ramsey-Type Numbers and the Independence Ratio. *Transactions of the American Mathematical Society*, 256:353–370, 1979.

53 T. Tao and V. H. Vu. *Additive Combinatorics*. Cambridge University Press, 2009.

54 Jorge Tavares, Francisco B. Pereira, and Ernesto Costa. Understanding the role of insertion and correction in the evolution of Golomb rulers. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, pages 69–76, 2004. `doi:10.1109/CEC.2004.1330839`.

55 Pál Turán. Egy gráfelméleti szélsőérték feladatról. *Matematikai és Fizikai Lapok*, 48:436–452, 1941.

56 D. W. Wang and Yue-Sun Kuo. A Study on Two Geometric Location Problems. *Inf. Process. Lett.*, 28(6):281–286, 1988.

57 D. J. A. Welsh. The Computational Complexity of Some Classical Problems from Statistical Physics. In *In Disorder in Physical Systems*, pages 307–321. Clarendon Press, 1990.

58 Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 216–226. Springer, 1979.

A Reducing Some MIS Problems to Subset-CSAT(L)

We reduce the problem of finding a maximum independent set for *c-far unit-disk graphs* to Subset-CSAT(L) for some 2-variable L. A unit disk graph G is an intersection graph of unit disks in the plane. We say a unit-disk graph is *c-far* if for each pair of disks the Euclidean distance between their centers does not fall within the interval $[0, c) \cup (2, 2 + c)$ for some constant $c > 0$. It is known that the maximum independent set problem remains **NP**-hard for c-far unit-disk graphs [41, 56], even when the locations of disks are given.

▶ **Theorem 22.** *There exists a polynomial-time many-one reduction from finding a maximum independent set for c-far unit-disk graphs to Subset-CSAT(L) for some 2-variable L.*

Proof. The reduction comes as follows. Let $D = \{d_1, d_2, \ldots, d_n\}$ be the set of the n disks and let $x(d_i)$ and $y(d_i)$ denote the x- and y-coordinate of disk d_i for each $i \in [1, n]$. We discretize the locations of disks in D so that $x(d_i)$ and $y(d_i)$ for all $i \in [1, n]$ are mapped to multiples of ε where ε is set as $c/6$. Observe that, if two disks intersect before the discretization, then their distance is in the range $[4c/6, 2 + 2c/6]$; if two disks do *not* intersect before the discretization, then their distance now falls within $[2 + 4c/6, \infty)$. If we enlarge the radius of all disks from 1 to $1 + 3c/12$, then the discretization does not alter whether two disks intersect or not. In other words, if two disks intersect, then the center of one disk is located at one of the $O(1/\varepsilon^2)$ discretized coordinates surrounding the center of the other.

Consequently, if we map each disk d_i to an integer $2^{x(d_i)/\varepsilon}3^{y(d_i)/\varepsilon}$, noting that the exponents are integers for each $i \in [1, n]$, and set

$$L_{\text{udisk}} = \{\ell(a, b) = 2^{r_1}3^{r_2}a - 2^{r_3}3^{r_4}b : \varepsilon\sqrt{(r_1 - r_3)^2 + (r_2 - r_4)^2} < 2 + c/2, r_1, r_2, r_3, r_4 \in \mathbb{N}\},$$

then it is clear that Subset-CSAT(L_{udisk}) is a restatement of finding a maximum independent set for c-far unit-disk graphs. ◀

Combining Theorem 22 and Theorem 2, we get:

▶ **Corollary 23.** *Finding a maximum independent set for c-far unit-disk graphs admits a* PTAS.

We remark here that one can have a result analogous to Corollary 23 for c-far intersection graphs whose underlying shape is a unit symmetric convex set. This result is not as general as for ordinary intersection graphs because c-farness implies that all nodes in the intersection graph have a constant degree.

B Initial Hardness Results

Our hardness proofs are based on the strong **APX**-hardness of Max Independent Set for sparse large-girth graphs, which can be shown by the following chain of reductions.

Let Max-3SAT-Δ be the subproblem of Max-3SAT so that there exists a constant Δ such that no variables of the formula appears in more than Δ clauses. Let Max-IS be the maximum independent set problem. In what follows, we will show that Max-3SAT-Δ $\leq_{\text{SPTAS}}$ Max-IS for sparse graphs $\leq_{\text{SPTAS}}$ Max-IS for sparse large-girth graphs.

▶ **Lemma 24.** Max Independent Set *for sparse graphs, i.e. with a linear number of edges, is strongly* **APX***-hard.*

Proof. Let $I_{3\text{SAT}}$ be an instance of Max-3SAT-Δ. We assume that $I_{3\text{SAT}}$ has n variables and m clauses, and each clause in $I_{3\text{SAT}}$ has exactly 3 literals. Otherwise, one can duplicate some literal in each of the 1-literal and 2-literal clauses. Given $I_{3\text{SAT}}$, we construct a graph $G = (V, E)$ as I_{MIS} as follows. For each $i \in [1, m]$, we add three nodes $v_{i_1}, v_{i_2}, v_{i_3}$ to V, link each pair of the three nodes with an edge, and label $v_{i_1}, v_{i_2}, v_{i_3}$ with the corresponding literal in the i-th clause. Then, for every pair of nodes in V, if their labels are literals which are negations of each other, then link an edge between them. Consequently, G has $3m$ nodes and at most $3m + 9\binom{\Delta}{2}n = O(m)$ edges. It can be checked that $I_{3\text{SAT}}$ can have t clauses simultaneously satisfied if and only if I_{MIS} has an independent set of size t. Moreover, the problem instances have size linear to each other. This gives a SPTAS reduction. ◀

▶ **Lemma 25.** *For every constant $c \geq 3$,* Max Independent Set *for sparse graphs of girth $\geq c$ is strongly* **APX***-hard.*

Proof. Let I_s (resp. $I_{s,g\geq c}$) be an instance of Max Independent Set for sparse graphs (resp. Max Independent Set for sparse graphs of girth $\geq c$). One can map I_s to $I_{s,g\geq c}$ by replacing each edge (v_a, v_b) with a path from v_a to v_b with $2c$ internal nodes, as shown in [44]. Hence, $I_{s,g\geq c}$ has girth $\geq 6c+3$, and I_s has an independent set t if and only if $I_{s,g\geq c}$ has an independent set of size $t + cm$. Every $(1-\varepsilon)$-approximation for $I_{s,g\geq c}$ determines that $I_{s,g\geq c}$ has an independent set of size $(1-\varepsilon)(t+cm)$, which corresponds to I_s having an independent set of size $(1-\varepsilon)t - \varepsilon cm = (1 - O(\varepsilon))t$, where the equality holds because c is a constant and $t = \Omega(n) = \Omega(m)$ by Turán's Theorem. Moreover, the problem instances have size linear to each other. This gives a SPTAS reduction. ◀

C Inapproximability Constants

Lastly, for each problem in the syntactically-defined class that does not admit a PTAS, we determine an inapproximability constant $1-\varepsilon$, so that it cannot be $(1-\varepsilon)$-approximated unless $\mathbf{P} = \mathbf{NP}$. We use the facts that Max Independent Set on 3-regular graphs cannot be approximated to within the constant $C_3 = 139/140 + \varepsilon$ for any constant $\varepsilon > 0$ [12], and Max Independent Set on 3-regular triangle-free graphs can not be approximated to within the constant $C_{3\Delta} = 1422/1432 + \varepsilon$ for any constant $\varepsilon > 0$ [18].

We first apply Lemma 13 to bound an inapproximability constant $1-\delta_r$ based on C_3 and then replace the use of Turán's Theorem in Lemma 13 with the AKS Theorem [4] and Staton's result [52] to bound the claimed inapproximability constant $1-\varepsilon_r$ based on $C_{3\Delta}$.

Since Max Independent Set on 3-regular graphs cannot be approximated to within C_3, from Lemma 13 we have following theorem:

▶ **Lemma 26.** *For every homogeneous, r-variate ($r \geq 3$), linear function $\ell(\mathbf{x})$,* Subset-CSAT$(\{\ell(\mathbf{x})\})$ *cannot be approximated to within any constant factor larger than $1-\delta_r$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where $\delta_r = \frac{1-C_3}{7+6(r-3)}$.*

Simply replacing C_3 with $C_{3\Delta}$ cannot increase δ_r because $C_3 < C_{3\Delta}$ and such a replacement in Lemma 26 makes δ_r smaller. Instead, we replace the use of Turán's Theorem, which applies to general graphs, with the AKS Theorem (see Theorem 27), which works for triangle-free graphs. In [3, 50], the constant in the big-Omega notation in AKS Theorem is bounded above by 1/100 and 1/8, respectively. Though the size of an independent set guaranteed by the AKS theorem is asymptotically larger than that of Turán theorem, it is numerically smaller when $d = 3$.

▶ **Theorem 27** (AKS Theorem [4])**.** *Every d-regular triangle-free graph has an independent set of size $\Omega(n \log d/d)$.*

Note that the constant in the big-Omega notation is universal for every d. For a particular value of d the constant can be larger. In particular, in [52] Staton shows that every 3-regular triangle-free graph has an independent set of size $5m/21$, which is more than the $m/6$ guaranteed by Turán's theorem. The constant 5/21 is tight due to Fajtlowicz [27]. Based on this improved guarantee of the size of an independent set, we obtain the following result.

▶ **Lemma 28.** *For each homogeneous, r-variate, linear function $\ell(\mathbf{x})$,* Subset-CSAT$(\{\ell(\mathbf{x})\})$ *cannot be approximated to within any constant factor larger than $1-\varepsilon_r$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where $\varepsilon_r = 1 - \frac{1-C_{3\Delta}}{5.2+4.2(r-3)}$.*

Lemma 28 and the proof of Theorem 2 together imply that:

▶ **Theorem 29.** *Let L be a finite set of homogeneous linear functions whose coefficients are in $\mathbb{Z}$. If L contains a homogeneous r-variate linear function $\ell(\mathbf{x})$ for some $r \geq 3$, then* Subset-CSAT(L) *cannot be approximated to within any constant factor larger than $1-\varepsilon_r$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where $\varepsilon_r = 1 - \frac{1-C_{3\Delta}}{5.2+4.2(r-3)}$.*

To obtain the inapproximability constants for Subset-DSAT(L), we need Lemma 30.

▶ **Lemma 30.** Max Independent Set *for graphs whose maximum degree ≤ 3 and girth $\geq g$ cannot be approximated to within any constant factor larger than $1-\varepsilon_g$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where $\varepsilon_g < \frac{1}{140(6\lceil (g-3)/6\rceil+1)}$.*

Proof. We prove this by giving a PTAS reduction from Max Independent Set for 3-regular graphs $G_{3r} = (V_{3r}, E_{3r})$ to Max Independent Set for graphs $G_{g^+} = (V_{g^+}, E_{g^+})$ of girth $\geq g$. We obtain G_{g^+} from G_{3r} by replacing each edge in E_{3r} with a path of length $2t+1$ ($t \in \mathbb{Z}$), connecting $2t$ new nodes. Hence, the smallest cycle in G_{g^+} is $3+6t$. We pick $t = \lceil (g-3)/6 \rceil$ so that G_{g^+} has no cycle of length $< g$.

It is known [44] that G_{g^+} has an independent set of size $t|E_{3r}| + k$ iff G_{3r} has an independent set of size k. Every $(1-\varepsilon)$-approximation algorithm for Max Independent Set of G_{g^+} can find an independent set of size $(1-\varepsilon)(t|E_{3r}| + k)$, which corresponds to an

independent set of size $(1-\varepsilon)k - \varepsilon t|E_{3r}| \geq (1-(6t+1)\varepsilon)k$ in G_{3r}, where the last inequality follows from the fact that $k \geq |E_{3r}|/6$ for every 3-regular graph, due to Turán's Theorem [53].

Based on [12], Max Independent Set for 3-regular graphs cannot be approximated to within $1-\varepsilon_{3r}$ for any $\varepsilon_{3r} < 1/140$. Thus, ε cannot be less than $\frac{1}{140(6t+1)} = \frac{1}{140(6\lceil(g-3)/6\rceil+1)}$. ◀

In the proof of Theorem 3, the girth g is set as $r+1$, where r denotes $|\mathbf{x}|$. Hence, we get:

▶ **Theorem 31.** *Let L be a finite set of homogeneous linear functions whose coefficients are in $\mathbb{Z}$. For each* Subset-DSAT*(L), if the solutions to $\bigvee_{\ell\in L}\ell(\mathbf{x}) =$ False form a vector space in general position and has dimension at least 2, then* Subset-DSAT*(L) cannot be approximated to within any constant factor larger than $1-\varepsilon_r$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where $\varepsilon_r = \frac{1}{140(6\lceil(r-2)/6\rceil+1)}$.*