# The Fine-Grained Complexity of Multi-Dimensional Ordering Properties


**Haozhe An** ✉
University of Maryland, College Park, MD, USA

**Mohit Gurumukhani** ✉
University of California at San Diego, CA, USA

**Russell Impagliazzo** ✉
University of California at San Diego, CA, USA

**Michael Jaber** ✉
University of California at San Diego, CA, USA

**Marvin Künnemann** ✉
Institute for Theoretical Studies, ETH Zürich, Switzerland

**Maria Paula Parga Nina** ✉
University of California at San Diego, CA, USA



## Abstract

We define a class of problems whose input is an $n$-sized set of $d$-dimensional vectors, and where the problem is first-order definable using comparisons between coordinates. This class captures a wide variety of tasks, such as complex types of orthogonal range search, model-checking first-order properties on geometric intersection graphs, and elementary questions on multidimensional data like verifying Pareto optimality of a choice of data points.

Focusing on constant dimension $d$, we show that any $k$-quantifier, $d$-dimensional such problem is solvable in $O(n^{k-1} \log^{d-1} n)$ time. Furthermore, this algorithm is conditionally tight up to subpolynomial factors: we show that assuming the 3-uniform hyperclique hypothesis, there is a $k$-quantifier, $(3k-3)$-dimensional problem in this class that requires time $\Omega(n^{k-1-o(1)})$.

Towards identifying a single representative problem for this class, we study the existence of complete problems for the 3-quantifier setting (since 2-quantifier problems can already be solved in near-linear time $O(n \log^{d-1} n)$, and $k$-quantifier problems with $k > 3$ reduce to the 3-quantifier case). We define a problem Vector Concatenated Non-Domination $\mathsf{VCND}_d$ (Given three sets of vectors $X, Y$ and $Z$ of dimension $d, d$ and $2d$, respectively, is there an $x \in X$ and a $y \in Y$ so that their concatenation $x \circ y$ is not dominated by any $z \in Z$, where vector $u$ is dominated by vector $v$ if $u_i \leq v_i$ for each coordinate $1 \leq i \leq d$), and determine it as the "unique" candidate to be complete for this class (under fine-grained assumptions).



**2012 ACM Subject Classification** Theory of computation → Complexity theory and logic; Theory of computation → Problems, reductions and completeness; Theory of computation → Verification by model checking

**Keywords and phrases** Fine-grained complexity, First-order logic, Orthogonal vectors

**Digital Object Identifier** 10.4230/LIPIcs.IPEC.2021.3

**Funding** *Russell Impagliazzo*: Work supported by the Simons Foundation and NSF grant CCF-1909634.
*Marvin Künnemann*: Research supported by Dr. Max Rössler, by the Walter Haefner Foundation, and by the ETH Zürich Foundation. Part of this research was performed while the author was employed at MPI Informatics.

**Acknowledgements** We would like to thank Rex Lei, Jiawei Gao, and Victor Vianu for helpful comments and discussion.




16th International Symposium on Parameterized and Exact Computation (IPEC 2021).
Editors: Petr A. Golovach and Meirav Zehavi; Article No. 3; pp. 3:1–3:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

Algorithmic problems based on comparing elements according to a total ordering relation are as fundamental as they are useful. Any introductory algorithms textbook starts with sorting and other comparison-based problems. For higher dimensional data, problems involving comparisons for multiple components, such as range queries, are equally fundamental in computational geometry. In databases, queries need to handle data with many fields that can be compared (beyond other relations on the data), such as listing all employees who are not managers of another employee, with seniority in one range and salary in another.

In this paper, we give a general, systematic study of the complexity of multi-dimensional comparison problems. We define complexity classes capturing the notion of "multi-dimensional comparison problems", as appropriate in geometry and in databases, with the classes $PTO_d$ representing geometric problems in $d$ dimensional data, and $TO_d$ representing problems that combine ordering and other relations for such data, as would be found in databases. We then identify the maximum complexity of problems in these classes under standard assumptions in fine-grained complexity, and relate the classes to each other and other studied complexity classes. For many subclasses, we find natural complete or hard problems where progress on better algorithms for these problems would result in better algorithms for the entire subclass.

While our results are varied, with upper bounds, conditional lower bounds and completeness results, a consistent theme emerges. Our classes are intermediate between two previously studied classes of logically defined problems, first-order in the sparse representation (e.g., graph problems in adjacency list format) and first-order in the dense representation (e.g, graph problems in adjacency matrix format). While orderings are dense relations, with quadratically many pairs for which they hold, they are a special case that can be represented succinctly, by giving an array of ranks for each element. What emerges in our results is that multi-dimensional ordering problems are very tightly connected to first-order in the sparse representation, and not directly connected to the dense representation. Thus, while they give substantially different settings, we give many senses in which sparse relations can be coded in terms of orders, and where orderings can be reduced to sparse relations.

### 1.1 A class of geometric ordering problems: $PTO_{k,d}$

As an example for multi-dimensional comparison problems, consider 2D orthogonal range searching: given a set of 2-dimensional data points $D$, answer Boolean queries of the form

$$\exists x \in D : x \in [\ell_1, u_1] \times [\ell_2, u_2],$$

where $[\ell_1, u_1] \times [\ell_2, u_2]$ is a given *orthogonal range*. Note that here, we may without loss of generality replace each point's coordinate in dimension $d$ by its *rank* among the coordinates in dimension $d$ of all points in $D$. Typical variants include to report, count or optimize over all elements in the query range. A long line of research starting in the 70s, including [37, 42, 22, 11, 40, 19], gives fast algorithms for such tasks, e.g., an algorithm to preprocess $D$ such as to answer queries in time $O(\log \log n)$ using space $O(n \log \log n)$, see [19]. Many more complex algorithmic tasks can be solved using orthogonal range techniques, see [25, 8] for an overview.

Also more complex tasks than mere orthogonal range searching arise naturally: In a set of $d$-dimensional data points $D$, consider a *feature* (or property) $F$ of the data points that can be described as being contained in an orthogonal range $[\ell_1, u_1] \times \cdots \times [\ell_d, u_d]$. Given a family $\mathcal{F}$ of such features, there are several natural questions to ask:

- decide if all features are present in the dataset:
  $\forall F = [\ell_1, u_1] \times \cdots \times [\ell_d, u_d] \in \mathcal{F} \;\; \exists x \in D : \;\; x \in F$
- decide if some data point displays all features:
  $\exists x \in D \;\; \forall F = [\ell_1, u_1] \times \cdots \times [\ell_d, u_d] \in \mathcal{F} : \;\; x \in F$
- decide if two different features are equivalent on $D$:
  $\exists F_1 \in \mathcal{F} \;\; \exists F_2 \in \mathcal{F} \;\; \forall x \in D : \;\; F_1 \neq F_2 \wedge (x \in F_1 \leftrightarrow x \in F_2)$.

Some of these questions can be quickly answered using orthogonal range reporting queries, for others it seems that already the output size of single such query might pose a possibly unnecessary bottleneck. Furthermore, some features might be comparison-based, but more complex than a simple orthogonal range, e.g.,[1]

$$x \in F(\ell_1, u_1, \ldots, \ell_d, u_d) \iff$$
$$(x_1 \in [\ell_1, u_1] \rightarrow (x_2 \in [\ell_2, u_2])) \wedge (x_1 \notin [\ell_1, u_1] \rightarrow (x_3, \ldots, x_d) \in [\ell_3, u_3] \times \cdots \times [\ell_d, u_d])).$$

In such cases, it would not be immediate whether orthogonal range search techniques can be used at all.

We formalize a notion of "multi-dimensional comparison problems" by introducing a class of problems $PTO_{k,d}$ (for "purely total ordering property") of model-checking a $k$-quantifier first-order property on a relational structure with $d$ total ordering relations (each succinctly represented as a sorted list of objects) as well as unary relations (to enable comparison of coordinates with constants). In particular, this class contains any property $\psi$ of the form

$$\psi = Q_1 x^{(1)} Q_2 x^{(2)} \ldots Q_k x^{(k)} : \phi(x^{(1)}, \ldots, x^{(k)}),$$

where $Q_i \in \{\exists, \forall\}$, $x^{(i)}$ ranges over a set of $d$-dimensional vectors (which we also call *objects*), and $\phi$ is an arbitrary Boolean formula involving only comparisons of the form $x_i^{(a)} \leq x_i^{(b)}$ with $1 \leq a, b \leq k$ (here, $x_i^{(a)}, x_i^{(b)}$ denotes the $i$-th dimension of $x^{(a)}, x^{(b)}$, respectively), as well as comparisons with constants. We will refer to $d$ as the *dimension* of a formula $\psi \in PTO_{k,d}$. For this paper throughout, we think of $\phi$ as fixed formula, and thus $k, d$ are constants. See Section 2 for further details.

The class $PTO_{k,d}$ includes all problems as mentioned above, but also tasks such as verifying Pareto optimality of a given set of $d$-dimensional data points with respect to a superset, or given a set of $d$-dimensional geometric objects, determine whether there are $k$ distinct such objects whose bounding boxes intersect.

We furthermore extend this class to $TO_{k,d}$, where we allow, beyond $d$ total ordering relations, also arbitrary additional relations (represented explicitly). These two classes encompass in particular the following types of problems:

- Model-checking first order properties of **geometric intersection graphs**: Presence of an edge in an intersection graph of axis-parallel boxes can be decided using comparisons of coordinates. Thus, any $k$-quantifier first-order property on such geometric intersection graphs in $\mathbb{R}^d$ can be formulated as a problem in $PTO_{k,d}$, such as finding $k$ pair-wise non-intersecting $d$-dimensional axis-parallel unit-cubes [38].[2]
- **temporal logic**: using a single total ordering relation, we may represent *precedence* in a time domain. Thus, we may express temporal logical statements involving expressions over future or past events in $TO_{k,1}$.

[1] The given expression could model the following feature: if a person is of working age ($x_1 \in [\ell_1, u_1]$), use criterion $x_2 \in [\ell_2, u_2]$, otherwise use $(x_3, \ldots, x_d) \in [\ell_3, u_3] \times \cdots \times [\ell_d, u_d]$.

[2] For even more involved types of algorithmic tasks beyond $k$-quantifier first-order properties, see, e.g., [20] (All-Pairs Shortest Paths) or [24] (NP-hard problems).

- relational databases with **ordered types**: in relational databases, we may use totally ordered data types (salaries of employees, time events, rank in a sorted list, etc.) as succinct representation to enable comparisons. In this context, studying the complexity of a problem in $TO_{k,d}$ corresponds to studying the *data complexity* of a fixed query.

## 1.2 Our results

Let $k \geq 2$. We show that any problem in $PTO_{k,d}$ involving $n$ objects can be solved in time $O(n^{k-1} \log^{d-1} n)$ which is $\tilde{O}(n^{k-1})$ for any constant dimension $d$. We extend this algorithm to run in time $O(m^{k-1} \log^{d-1} m)$ for sentences in $TO_{k,d}$, where $m$ denotes the sum of the number of objects and the size of the additional relations, i.e., the number of tuples contained in the relation. We show the matching conditional lower bound that there is some sentence in $PTO_{k,3k-3}$ that requires time $\Omega(n^{k-1-o(1)})$ under the 3-uniform hyperclique hypothesis [36, 2, 15, 34] – this hypothesis postulates that $n^{k\pm o(1)}$ running time is essentially best possible for finding cliques in hypergraphs. (See the full version of this paper for further details.)

Beyond these general upper and lower bounds, we also seek to identify *hard* or even *complete* problems for this class. Such problems capture the full generality of these classes, in the sense that finding a significantly improved algorithm for this problem would give an improved algorithm for all problems in the class. We use the following fine-grained notion of hardness/completeness: Formally, let $P$ be a problem whose best known algorithm runs in time $T_P(n)$ and let $C$ be a class of problems whose best known algorithms runs in time $T_C(n)$. We say $P$ is *hard* for a class of problems $C$, if any $T_P(n)^{1-\epsilon}$-time algorithm for $P$ with $\epsilon > 0$ gives a $T_C(n)^{1-\epsilon'}$-time algorithm for all problems in $C$ for some $\epsilon' > 0$. We say that $P$ is *complete* for $C$, if it is hard for $C$ and contained in $C$. In particular, if $P$ is complete for $C$, then $P$ admits substantial improvements over time $T_P(n)$ if and only if *all* problems in $C$ admit substantial improvements over $T_C(n)$.

We identify such problems for specific quantifier structures. In particular, we focus on the 3-quantifier case, since all 2-quantifier $O(1)$-dimensional total order properties can be solved in near-linear time $\tilde{O}(n)$ (Theorem 2), and all $k$-quantifier properties with $k > 3$ can be reduced to the 3-quantifier case via brute forcing (Corollary 4). Focusing on $PTO_{k,d}$, we obtain the following results (see Table 1):

1. For existentially quantified pure total ordering properties (denoted by $PTO_{\exists\exists\exists,d}$), we give an $\tilde{O}(n^{2\omega/(\omega+1)}) = \tilde{O}(n^{1.407})$ time algorithm and identify the well-studied triangle detection in sparse graphs as a complete problem[3].
2. For the quantifier structure $\forall\exists\exists$, we also give an $\tilde{O}(n^{2\omega/(\omega+1)}) = \tilde{O}(n^{1.407})$ time algorithm by showing that the problem of *counting*, for each edge in a sparse graph, the number of triangles containing this edge is *hard* for the class $PTO_{\forall\exists\exists,d}$. Since we reduce to a counting problem rather than a member of this class, we do not obtain a completeness result, however.
3. For the quantifier structure $\exists\forall\exists$, we were unable to find a complete or hard problem. However, we give evidence that this quantifier structure does not contain a complete problem for $PTO_{k,d}$ by showing that all $PTO_{\exists\forall\exists,d}$ problems have a $\tilde{O}(n)$-time nondeterministic and co-nondeterministic algorithm. Since we also show a $n^{2-o(1)}$ SETH[4]-based

---

[3] Strictly speaking, we identify the following 3-dimensional problem (which is linear-time equivalent to triangle detection in sparse graphs) as complete for $PTO_{\exists\exists\exists,d}$: $\exists x, y, z : x_1 = z_1 \wedge x_2 = y_2 \wedge y_3 = z_3$.

[4] Strong Exponential Time Hypothesis (SETH) for CNF-SAT: For all $\epsilon > 0$, there exists a $k$ so that $k$-CNF-SAT cannot be solved in time $O(2^{n(1-\epsilon)})$ [33].

lower bound for $PTO_{3,d}$ when $d \to \infty$, this rules out existence of such a complete problem using deterministic reductions under NSETH, a nondeterministic variant of SETH [17]. We also give a conditional lower bound of $n^{2-o(1)}$ under the Hitting Set conjecture.

4. Finally, for the seemingly most difficult quantifier structure of $\exists\exists\forall$, we show $n^{2-o(1)}$-time conditional lower bounds under SETH and the 3-uniform hyperclique hypothesis, and identify the following complete problem for $PTO_{\exists\exists\forall,d}$, which we call Vector Concatenated Non-Domination $\mathsf{VCND}_d$: Given three sets of vectors $X, Y$ and $Z$ of dimension $d, d$ and $2d$, respectively, is there an $x \in X$ and a $y \in Y$ so that their concatenation $x \circ y$ is not dominated by any $z \in Z$, where vector $u$ is dominated by vector $v$ if $u_i \le v_i$ for each coordinate $1 \le i \le d$.

Note that this covers all quantifier structures for $k = 3$, as deciding $Q_1Q_2Q_3\phi$ with $Q_i \in \{\exists, \forall\}$ is equivalent to deciding $\overline{Q_1}\,\overline{Q_2}\,\overline{Q_3}\,\overline{\phi}$ where $\overline{\forall} = \exists, \overline{\exists} = \forall$ and $\overline{\phi}$ is the negation of $\phi$.

These results identify the $\mathsf{VCND}_d$ problem as the essentially *only* candidate (up to fine-grained equivalence) to be complete for $PTO_{3,d}$ under NSETH: It is complete for $\exists\exists\forall$, and all problems with a different 3-quantifier structure have either improved deterministic or (co-)nondeterministic algorithms, and thus cannot be complete without major consequences in fine-grained complexity. It remains a challenge to prove or disprove completeness of $\mathsf{VCND}_d$ for $PTO_{3,d}$ (beyond its completeness for $PTO_{\exists\exists\forall,d}$).

Since the above results motivate $\mathsf{VCND}_d$ as a central problem for $PTO_{k,d}$, we work towards algorithmic improvements for this problem. In particular, we obtain an $\tilde{O}(n^{2-\frac{1}{2^d}})$-time algorithm for $\mathsf{VCND}$ whenever one set of vectors is of dimension 2 and the other is of dimension $d$. Note that obtaining such an $O(n^{2-\epsilon(d)})$ time algorithm with $\epsilon(d) > 0$ for general $\mathsf{VCND}_d$ would refute the 3-uniform hyperclique hypothesis by our conditional lower bound and completeness result.

Finally, we show that our algorithmic results extend to the class $TO_{k,d}$ (see Section 3 for details), while all hardness results trivially apply, since they are already proven for the subclass $PTO_{k,d}$. Generally speaking, this shows that the database setting (with additional sparse relations) does not increase the fine-grained complexity compared to the geometric setting of purely total ordering properties.

## 1.3 Previous work

This work continues a relatively new direction, fine-grained complexity of complexity classes. *Fine-grained complexity* aims to not only qualitatively classify problems as "easy" or "hard", but (to the extent possible) pin-point their exact complexities. We now have a wide collection of standard algorithmic problems where any significant improvements in algorithmic running time would refute one or more conjectures about well-studied problems, such as the $k$-SUM problem [26], All Pairs Shortest Paths [44, 3, 36], SAT [33, 41], or Orthogonal Vectors [6, 14, 1, 12, 16, 39, 35, 9, 2, 13]. Recent work in fine-grained complexity has gone from considering problems one at a time to following traditional complexity in considering *classes* of problems. Fine-grained reductions often cut across the usual complexity classes (with reductions from $NP$-complete problems to first-order properties, for example), but on the other hand, fine-grained complexity distinguishes between problems with the same traditional complexities (e.g., two different $NP$-complete problems might have very different properties in fine-grained complexity). Nevertheless, there are now a number of classes of problems, grouped by logical structure or common format, whose fine-grained complexity is at least partially understood: dense first-order properties [43]; sparse first order properties [17, 29, 15]; several extensions of first order [28]; and certain formats of dynamic programming problems [35, 27].

The most closely related previous work to our results are [43, 29]. Both of these papers consider the class of *first-order definable properties*, the first for the dense case (where each relation is given as a matrix, aka adjacency matrix format), and the second for the sparse case (where the input is given as a list of tuples in the relations, e.g., for graphs, adjacency list format). This class is natural both in terms of computational complexity, where it is the uniform version of $AC_0$ ([30]), and in database theory, because these are the queries expressible in basic SQL [7]. First-order logic can also express many polynomial time computable problems: Orthogonal Vectors, $k$-Orthogonal Vectors, $k$-Clique, $k$-Independent Set, $k$-Dominating Set, etc. Not only were the likely complexities of the hardest problems (as a function of number of quantifiers) given, but in the second paper, a natural complete problem was identified, the Orthogonal Vectors problem (OV). The conclusion was that there were substantial improvements possible in the worst-case complexity of model checking for first-order properties if and only if the known algorithms for Orthogonal Vectors can be substantially improved. Using a recent sub-polynomial improvement in OV algorithms by [4, 21], they obtained a similar improvement in model checking for every first-order property. [28] extends this work to related logics such as transitive closure logics, first-order logic on totally ordered sets, and first-order logic with function symbols. They show that model checking for first-order logic with a single total ordering is actually equivalent to that for unordered structures under fine-grained reductions. In contrast, we show that for even two orderings, the model checking problem becomes substantially harder, meaning we require new techniques to characterize the complexity of problems on multi-dimensional data.

There is also work on classes of problems that are related in spirit, but do not form a well-studied complexity class. V.-Williams and Williams [44] study problems related to shortest paths in graphs, and shows that many are subcubic-time equivalent. Künnemann et al. [35] study dynamic programming problems with a similar structure and give a unified treatment of their fine-grained complexities. Gao [27] extends this class of dynamic programming problems from lines to tree-like structures such as bounded treewidth graphs.

## 2 Preliminaries

The following notion of *fine-grained reductions* was introduced in [44].

▶ **Definition 1** (Fine-grained reduction). *Let $(\Pi_1, T_1(m)) \leq_{FGR} (\Pi_2, T_2(m))$ denote that for every $\epsilon > 0$ there is a $\delta > 0$ and a Turing reduction from $\Pi_1$ to $\Pi_2$ so that the time for the reduction (not counting oracle calls) is $O(T_1(m)^{1-\delta})$ and $\sum_q (T_2(|q|))^{1-\epsilon} \in O(T_1(m)^{1-\delta})$, where the sum is over all oracle calls $q$ made by the reduction on an instance of size $m$.*

In other words, if there is some $\epsilon > 0$ such that problem $\Pi_2$ is in $\mathsf{TIME}((T_2(m))^{1-\epsilon})$, then problem $\Pi_1$ is in $\mathsf{TIME}((T_1(m))^{1-\delta})$ for some $\delta > 0$, i.e., if $\Pi_2$ can be solved substantially faster than $T_2$ then $\Pi_1$ can be solved substantially faster than $T_1$. If both $T_1$ and $T_2$ are $\Theta(m^2)$, the reduction is called a subquadratic reduction. We say that $\Pi_1$ and $\Pi_2$ are *fine-grained equivalent* if there is a fine-grained reduction from $\Pi_1$ to $\Pi_2$ and vice versa.

We use this notation not only on single problems but also on classes of problems. Let $C_1$ and $C_2$ be classes problems. $(C_1, T_1(m)) \leq_{\text{FGR}} (C_2, T_2(m))$ if for all problems $\Pi_1 \in C_1$ there is a $\Pi_2 \in C_2$ so that $(\Pi_1, T_1(m))$ fine-grained reduces to $(\Pi_2, T_2(m))$.

### Details on $PTO_{k,d}$ and $TO_{k,d}$

In this paper, we consider the fine-grained complexity of model checking problems definable in first-order logic on structures with $d$ binary relations $x \leq_i y$, $1 \leq i \leq d$, where each binary relation is a total pre-order of the universe (i.e., transitive, reflective, total, but not necessarily anti-symmetric.)

**Total orders.** We use $x \leq_i y$ to represent the $i$'th relation in our family holding between $x$ and $y$. Such a relation is dense, holding for $\Theta(n^2)$ pairs of elements. However, we can represent such a representation succinctly, by giving an array which for each element specifies its rank in a list sorted by the ordering relation (with some elements having the same rank, if inequality holds in both directions). It is in this succinct format that ordering relations are described for our problems.

Equivalently, we may represent all ordering relations by representing each object $x$ as a $d$-dimensional vector $(x_1, \ldots, x_d)$, where $x_i$ denotes the rank of $x$ in the $i$'th ordering relation. Thus, it is equivalent to write $x \leq_i y$ or $x_i \leq y_i$, and we will switch between these two based on which seems clearer for the given circumstance.

The vectors we get in this way are very special, in that the coordinates are always positive integers from 1 to $n$. However, also problems defined about $d$ dimensional vectors over any totally ordered domain (such as $\mathbb{R}$) fall into our setting, since in $O(n \log n)$ time we can replace each $x_i$ with its rank in the set of $i$'th coordinates of vectors.

**Unary relations.** We also allow unary relations, or, equivalently, comparisons to constants. More precisely, any unary relation $U$ is represented as a list of objects for which $U$ holds. Apart from allowing us to put objects into categories (sometimes called *colored* properties), this enables us to express comparisons of coordinates with constants: To express whether $x \leq_i \gamma$ for some constant $\gamma$, we introduce a unary relation symbol $U_i^{\leq\gamma}$ that holds for all $x$ with $x_i \leq \gamma$. Thus from now on, it suffices to declare constants $\gamma$ explicitly, and afterwards we may express arbitrary comparisons like $x_i \neq \gamma$ or $x_i > \gamma$. Note that since we always consider fixed formulas $\psi$, each considered property will use $O(1)$ constants for comparisons.

**Definition of $PTO_{k,d}$.** We denote the class of purely total ordering model-checking problems for first-order formulas in pre-orderings and unary relations specified as above where the formula has $d$ distinct ordering relations and $k$ total occurrences of quantifiers by $PTO_{k,d}$. $PTO_k$ is the union of $PTO_{k,d}$ over all constants $d$. We can further divide $PTO_k$ into $2^k$ sub-classes based on the quantifier structure, so for example $PTO_{\exists\exists\exists}$ is the sub-class of $PTO_3$ where the model-checking problems are for formulas of the form $\exists x \exists y \exists z \Phi(x, y, z)$ where $\Phi$ is quantifier-free. We let $n$ be the size of the universe of the structure, which is also, up to constant factors, the size in terms of $O(\log n)$-bit words required to specify all total pre-orderings and unary relations. Algorithm time for problems in $PTO$ is thus measured in terms of $n$. In this format, it is a constant time operation to evaluate whether any relation is true or false for specified elements.

**Definition of $TO_{k,d}$.** We generalize $PTO_{k,d}$ to the class $TO_{k,d}$ by also allowing the formula and models to have any constant number of sparse relations of any constant arity. These are specified as lists of tuples where the relation holds. Let the problem size be denoted by $m$, which is equal to the sum of the number of elements $n$ and the number of tuples.

We assume all algorithms start with quasi-linear time preprocessing steps to create data structures such as hash tables or binary search trees that allow fast determination (constant time or logarithmic time) of whether a relation holds for given elements, and allows one to list the tuples in a relation that contain a given element in at most poly-log time + poly-log time times the number of such tuples.

**On the difference.** $PTO_{k,d}$ is a more "geometric" class of problems, and so it is interesting when we can reduce combinatorial problems to this class. Therefore, we will focus on these classes when giving conditional hardness results. $TO_{k,d}$ is closer to the type of problems

**Table 1** Our results for $PTO_{k,d}$, where we assume that $d$ is an arbitrarily large constant.

| **Quantifier structure** | 3 quantifiers | | $k$ quantifiers, $k > 3$ | |
|---|---|---|---|---|
| $\ldots\exists\exists\forall$ (sym.: $\ldots\forall\forall\exists$)<br>complete: $\mathsf{VCND}_d$<br>(Thm. 13) | $\tilde{O}(n^2)$ | $n^{2-o(1)}$ for $d = 6$<br>(3-unif. HC, Thm. 14)<br>$n^{2-o(1)}$ for $d \to \infty$<br>(SETH, Thm. 15) | $\tilde{O}(n^{k-1})$ | $n^{k-1-o(1)}$ for $d = 3k - 3$<br>(3-unif. HC, Thm. 14) |
| $\ldots\exists\forall\exists$ (sym.: $\ldots\forall\exists\forall$)<br>complete: open | $\tilde{O}(n^2)$<br><br>$\tilde{O}(n)$<br>(co-)nondet. | $n^{2-o(1)}$ for $d \to \infty$<br>(Hitting Set, Thm. 11) | $\tilde{O}(n^{k-1})$<br><br>$\tilde{O}(n^{k-2})$<br>(co-)nondet. | <br><br>$n^{k-2-o(1)}$ for $d = 2$<br>(SETH, Thm. 12) |
| $\ldots\forall\exists\exists$ (sym.: $\ldots\exists\forall\forall$)<br>complete: open<br>hard: ETC<br>(Thm. 8) | $\tilde{O}(n^{\frac{2\omega}{\omega+1}})$<br>$= O(n^{1.41})$ | | $\tilde{O}(n^{k-\frac{\omega+3}{\omega+1}})$<br>$= O(n^{k-1.59})$ | |
| $\ldots\exists\exists\exists$ (sym.: $\ldots\forall\forall\forall$)<br>complete: triangle det.<br>(Thm. 6) | $\tilde{O}(n^{\frac{2\omega}{\omega+1}})$<br>$= O(n^{1.41})$ | | $\tilde{O}(n^{k-\frac{\omega+3}{\omega+1}})$<br>$= O(n^{k-1.59})$ | |

that might arise in applications such as database queries. Therefore, we will focus on $TO_{k,d}$ when giving algorithms or other upper bounds on complexity. Since $PTO_{k,d} \subseteq TO_{k,d}$, lower bounds for $PTO_{k,d}$ are stronger results, and upper bounds for $TO_{k,d}$ are stronger results.

**Further examples of problems in $PTO_{k,d}$.** To define further well-studied problems in $PTO_{k,d}$, we say that a vector $u$ *dominates* vector $v$ if $u_i \geq v_i$ for all $1 \leq i \leq d$, and denote this by $u \geq_{dom} v$. Furthermore, given a set of $d$-dimensional real vectors $A$, we say that vector set $B$ is *Pareto-optimal for $A$* if for every $a \in A$ there is a $b \in B$ with $b \geq_{dom} a$.

- Vector Domination Problem (see, e.g. [31]): Given two sets of $d$-dimensional real vectors $A$ and $B$, are there two vectors $u \in A$ and $v \in B$ such that $u \geq_{dom} v$?
- Pareto Optimality Verification (see, e.g. [32]): Given a set $A$ of vectors, and a candidate vector set $B$, determine whether $B$ is indeed Pareto optimal for $A$.

From the definition, both problems are in $PTO_{2,d}$. As we will see, they can be solved in time $O(n \log^{d-1} n)$. For superconstant dimension $d$, [31, 18] give further improvements.

## 3 Technical Overview

In this section, we give the main ideas for all of our results, see Table 1 for an overview. Due to space constraints, the proofs are deferred to the full version of this paper.

One of our main results is an upper-bound on model-checking sentences in $PTO_{k,d}$ and $TO_{k,d}$.

▶ **Theorem 2.** *There is an algorithm running in time $O(n \log^{d-1}(n))$ for model-checking a two-quantifier formula $Q_1 x Q_2 y \varphi(x, y)$ with $d$ ordering relations and unary predicates.*

Specifically, we obtain this result using the following lemma, which we obtain by a reduction to orthogonal range counting.

▶ **Lemma 3.** *Given a formula $\varphi(x, y)$ with $d$ ordering relations and unary predicates and two sets $X$, $Y$ of vectors in $\mathbb{R}^d$, there is an $O(n \log^{d-1}(n))$ time algorithm that returns an array $A$ indexed by each $x \in X$ so that $A[x]$ is the number of $y \in Y$ so that $\varphi(x, y)$ is true.*

Combining the above theorem with exhaustive search over the first $k-2$ quantifiers yields

▶ **Corollary 4.** *Model-checking formulas in $PTO_{k,d}$ is in* $\mathsf{TIME}(n^{k-1}\log^{d-1}(n))$.

If we have additional explicitly represented relations, more work is required. For such cases, throughout the paper, we will always assume that these relations are *sparse*, i.e., the total input size is $m = O(n)$. In this case, we obtain the same asymptotic running time.

▶ **Theorem 5.** *Model-checking formulas in $TO_{k,d}$ is in* $\mathsf{TIME}(m^{k-1}\log^{d-1}(m))$.

The idea is to reduce the problem to the purely totally ordered case by assuming that all sparse relations are empty; using Lemma 3 for the 2-quantifier case, we can obtain for each $x$ the number of $y$ satisfying the condition. We then repair these counts to the true values by iterating over the additional sparse relations, similar to the baseline algorithm in [29].

Note that in Section 3.4, we discuss a lower bound proving these baseline algorithms to be conditionally optimal under fine-grained hardness assumptions.

In the remainder of the section, we distinguish our results based on the quantifier structure. Since any $k$-quantifier formula with $k > 3$ reduces to the 3-quantifier setting via brute force over the first $k-3$ quantifiers, we only regard 3-quantifier structures.

## 3.1 Quantifier Structures Ending in ∃∃∃

Recall that informally, we call a problem *complete* for a class if it is contained in the class and model-checking any sentence in the class reduces to our problem. For sentences in $PTO_{k,d}$ ending in ∃∃∃, we show that detecting triangles in a sparse graph is complete for this class. By current running time bounds for the problem [10], we obtain a running time of $\tilde{O}(n^{2\omega/(\omega+1)}) = \tilde{O}(n^{1.407\ldots})$.

▶ **Theorem 6.** *The triangle detection problem in sparse graphs is fine-grained equivalent to a problem that is complete for model-checking ∃∃∃ formulas with only ordering relations and unary relations.*

More precisely, the following ordering property is shown to be complete: $\exists x \exists y \exists z : x_1 = z_1 \wedge x_2 = y_2 \wedge y_3 = z_3$ which is easy to be seen equivalent to triangle detection in sparse graphs.

Intuitively, we reduce to this problem as follows: Given a formula $\exists x \exists y \exists z \phi(x, y, z)$, we can determine whether $\phi(x, y, z)$ holds once we know all comparisons between $x, y, z$ in each dimension $i$. A challenge here is to reduce comparisons like $x_i < y_i$ to an equality check: Similar to a trick used in [45], we do this by guessing the highest-order bit of *divergence* between $x_i$ and $y_i$ to obtain a "proof" only involving equalities; since we may assume that $1 \leq x_i, y_i \leq n$ (by working in *rank* space), there are only $O(\log n)$ choices for a single comparison. The key observation is that the quantifier structure is sufficiently well behaved to make this reduction work: we only need to guess these bits of divergence for $O(d)$ many comparisons and can express correctness of all proofs for comparisons between $x$ and $z$ using equality on the first dimension, between $x$ and $y$ using the second dimension, and between $y$ and $z$ using the third dimension. In total, this results in an admissible blow-up of $\log^{O(d)} n$.

We turn to the setting with additional sparse relations, i.e., formulas in $TO_{\exists\exists\exists,d}$. Here we establish the triangle *counting* problem in sparse graphs as hard for the class. Since the approach of [10] also gives a counting algorithm in the same running time as detection, we establish the same algorithmic upper bound.

▶ **Theorem 7.** *Every problem in $TO_{\exists\exists\exists,d}$ reduces to the problem of counting the number of triangles in a sparse graph via reductions that preserve time up to polylog factors.*

Handling the additional sparse relations is highly non-trivial. In particular, to obtain our result, we first show that the triangle counting problem is hard for model-counting $\exists\exists\exists$ formulas in the sparse setting of [29], which is interesting in its own right.

Since triangle detection is a classical problem, improving the bound of $O(n^{1.407})$ for $\exists\exists\exists$ structures already in the purely total ordering case would be a major algorithmic result.

## 3.2 Quantifier Structures Ending in $\forall\exists\exists$

For quantifier structures ending in $\forall\exists\exists$, we obtain a hard problem: We show that every problem in $TO_{\forall\exists\exists,d}$ (and thus also $PTO_{\forall\exists\exists,d}$) reduces to that of determining, for each edge in a sparse graph, how many triangles contain this edge; we call this problem *Edgewise Triangle Counting (ETC)*. Again, currently the best algorithm for this problem is essentially the same as that for triangle detection and counting [10].

▶ **Theorem 8.** *Edgewise Triangle Counting is hard for model-checking $TO_{\forall\exists\exists,d}$ formulas.*

Since the high-level arguments for this results substantially build on the hardness result for $TO_{\exists\exists\exists,d}$, we defer all details for this result to the full version of this paper.

## 3.3 Quantifier Structures ending in $\exists\forall\exists$

For the quantifier structure of $\exists\forall\exists$, we are unable to establish a complete problem. However, this quantifier structure admits (co-)nondeterministic algorithms that are faster than the baseline algorithm.

▶ **Theorem 9.** *Model-checking formulas in $PTO_{k,d}$ ending in $\exists\forall\exists$ can be done in nondeterministic and co-nondeterministic time $O(n^{k-2}\log^{d-1}(n))$.*

The main idea is as follows: Consider any $\exists x\forall y Qz\phi(x,y,z)$ property. For the nondeterministic algorithm, we simply (nondeterministically) guess $x$ and solve the remaining 2-quantifier problem $\forall y Qz\phi(x,y,z)$ in time $O(n\log^{d-1} n)$ using the baseline algorithm. For the co-nondeterministic algorithm, we need to verify that $\forall x\exists y\overline{Q}z\overline{\phi}(x,y,z)$. Here, for every $x$, we (nondeterministically) guess a witness $y_x$ and solve the remaining $\overline{Q}z\overline{\phi}(x,y_x,z)$ formula using the approach of Theorem 2.

For the case of total ordering properties with additional sparse relations, this approach is not directly applicable: If, e.g., all guessed witnesses $y_x$ happen to participate in many tuples of the sparse relations, we have to repeatedly solve problems with a large input size. We remedy this problem by taking care of such large degree witness $y_x$ explicitly; while this incurs a certain slow-down, we can limit it to a factor of $O(\sqrt{n})$.

▶ **Theorem 10.** *Model-checking formulas in $TO_{k,d}$ ending in $\exists\forall\exists$ can be done in nondeterministic and co-nondeterministic time $O(m^{k-3/2}\log^{d-1}(m))$.*

As a consequence of the above nondeterministic algorithms, assuming NSETH [17], we cannot establish hardness beyond $n^{k-2-o(1)}$ for $PTO_{\exists\forall\exists,d}$ using deterministic SETH-based reductions. However, by reducing from a problem with low (co-)nondeterministic complexity, specifically, the Hitting Set conjecture [5], we can give a conditional lower bound already for $PTO_{\exists\forall\exists,d}$ (as $d\to\infty$) that matches our baseline algorithm.

▶ **Theorem 11.** *Assuming the Hitting Set conjecture, for all $\epsilon > 0$, there exists some $d$ and a $PTO_{\exists\forall\exists,d}$ sentence that cannot be solved in time $O(n^{2-\epsilon})$.*

The proof of this result is reminiscent to some reductions in [23]. We reduce from Hitting Set (given sets of vectors $A, B \subseteq \{0,1\}^{c\log n}$ for arbitrary $c$, determine whether some $a \in A$ is non-orthogonal to all $b \in B$) to a formula $\exists x\forall y\exists z\psi(x,y,z)$ as follows: We think of $x$ ranging over vectors $a \in A$, $y$ ranging over $b \in B$, and think of $z$ as a "proof" of the fact that $a, b$ are non-orthogonal, given by a prover Merlin. There is a trade-off between size of the proofs and the required dimension to represent the vectors, which we set in a way that bounds the number of possible proofs to $O(n)$, resulting in a dimension $d$ growing only with $c$ (independently of $n$).

We also give a conditional lower bound from SETH for $k > 3$ that matches the NSETH barrier following from the (co-)nondeterministic algorithms. Notably, this lower bound already applies to dimension $d = 2$.

▶ **Theorem 12.** *Assuming SETH, there exists some $PTO_{k,2}$ sentence ending in $\exists\forall\exists$ that cannot be solved in time $O(n^{k-2-\epsilon})$ for any $\epsilon > 0$.*

We reduce the $k$-Orthogonal Vectors problem into an $\exists^k\forall\exists$-quantified 2-dimensional formula. Intuitively, the first $k$ existential quantifiers choose $k$ vectors, the $\forall$-quantifier ranges over all vector-dimensions to test, and crucially, the final $\exists$-quantifier enables to guess which of the $k$ vectors has a 0-coordinate in this vector-dimension. Here, the final $\exists$-quantifier is instrumental in making the formula's dimension independent of the vector dimensions.

## 3.4 Quantifier Structures ending in ∃∃∀

For sentences in $PTO_{k,d}$ ending in $\exists\exists\forall$, we obtain the complete problem $\mathsf{VCND}_d$: Given three sets of vectors $X, Y$ and $Z$ of dimension $d, d$ and $2d$, respectively, determine if there an $x \in X$ and a $y \in Y$ so that their concatenation $x \circ y$ is not dominated by any $z \in Z$.

▶ **Theorem 13.** *For all $d$, there exists a $d'$ such that $\mathsf{VCND}_{d'}$ is complete for model-checking $\exists\exists\forall$ formulas in $PTO_{k,d}$.*

This is one of our most interesting results. We reduce a formula $\exists x \in X\exists y \in Y\forall z \in Z : \psi(x,y,z)$ to $\mathsf{VCND}_d$ as follows: We carefully divide all pairs in $X \times Y$ into instances $(X_1,Y_1),\dots,(X_L,Y_L)$ such that for each instance $(X_\ell,Y_\ell)$, all comparisons $x_i < y_i, x_i = y_i, x_i > y_i$ for all dimensions $i$ are uniform among pairs $x \in X_\ell, y \in Y_\ell$. Thus, for each $\ell$, we may simplify $\psi$ to a formula $\psi_\ell$ not involving comparisons between $x$ and $y$. In particular, we may express $\psi_\ell$ in CNF, where each clause is a disjunction of $\{<,\leq,\geq,>\}$-comparisons between $x_i$ and $z_i$ or between $y_i$ and $z_i$ (in some dimension $i$). Since all such clauses need to be fulfilled simultaneously, for each $z \in Z$ and clause $C$, we introduce some $z_C$ chosen such that the clause $C$ is falsified if and only if $x \circ y$ are dominated by $z_C$.

We show a matching conditional lower bound of $n^{k-o(1)}$ for $PTO_{\exists^k\forall,d}$ under the 3-uniform hyperclique hypothesis.

▶ **Theorem 14.** *For $k \geq 2$ and $h \geq 3$, under the $h$-uniform HyperClique hypothesis, there is a sentence in $PTO_{k+1,hk}$ ending in $\exists\exists\forall$ that requires time $\Omega(n^{k-o(1)})$.*

We use the first $k$ quantifiers to represent a choice of clique nodes, each represented in its own dimension, and use the $\forall$ quantifier to check that no forbidden configuration is used (a non-edge in the given hypergraph). Naively, this would create $\Theta(n^3)$ rather than $O(n)$ objects, which we remedy by reducing from finding hypercliques of size $hk$ (rather than $k$).

We also establish a SETH-based lower bound directly for $\mathsf{VCND}_d$. The reduction is very similar to our Hitting-Set-based lower bound for ∃∀∃-structures.

▶ **Theorem 15.** *Assuming SETH, for every $\epsilon > 0$, there is a $d$ such that* $\mathsf{VCND}_d$ *requires time $\Omega(n^{2-\epsilon})$.*

**Specialized algorithm for $\mathsf{VCND}_d$.** Since our completeness results establishes $\mathsf{VCND}_d$ as a central problem for the study of $PTO_{k,d}$, we consider special cases of the problem. In particular, if one of these sets contains vectors of dimension 2, while the other contains vectors of dimension $d$, we show the following algorithm, which uses the Erdös-Szekeres Theorem as main ingredient. We use this to extract lists of vectors so that when we restrict to any dimension, the vectors appear in monotonic increasing or decreasing order. This way, the vectors that dominate some fixed vector $x$ form an interval, which allows us to take advantage of fast segment trees that solve an interval covering problem.

▶ **Theorem 16.** *There is a $\widetilde{O}(n^{2-\frac{1}{2^d}})$ time algorithm for* VCND *when one set of vectors is of dimension 2 and the other is of dimension $d$.*

Note that such an improvement to $\tilde{O}(n^{2-\epsilon(d)})$ with $\epsilon(d) > 0$ for the general $\mathsf{VCND}_d$ problem would refute the 3-uniform hyperclique hypothesis by Theorem 14. In the appendix, we also give an algorithm for very high-dimensional $\mathsf{VCND}_d$.

## 4 Conclusion and open problems

We have introduced general classes $TO_{k,d}, PTO_{k,d}$ of multidimensional ordering problems as model-checking problems for $k$-quantifier first-order formulas over $d$ succinctly represented ordering relations (with or without additional explicitly represented relations). We gave a conditionally tight algorithm running in time $O(m^{k-1} \log^d m)$ for all these problems. For $PTO_{k,d}$, we gave complete or hard problems for most quantifier structures, and identified a problem $\mathsf{VCND}_d$ as the essentially only candidate to be complete for $PTO_{k,d}$.

The main open problem is to prove or disprove that $\mathsf{VCND}_d$ is complete for $PTO_{k,d}$. The major challenge here is to reduce ∃∀∃-quantified ordering problems to the ∃∃∀-quantified $\mathsf{VCND}_d$. Such a reduction is possible in the unordered setting [29], but its unclear how to make this approach work in our setting. Likewise, can we prove that a hybrid version of $\mathsf{VCND}_d$ and the orthogonal vectors problem (which is complete for the sparse-relational setting [29]) is complete for $TO_{k,d}$? An intermediate step could be to find a complete problem for ∃∀∃-quantified ordering problems.

A further general algorithmic question is to study existence of improved algorithms for very small constant dimensions $d$, such as $d = 1$ and $d = 2$, in particular the existence of $O(n^{2-\epsilon(d)})$ time algorithms with $\epsilon(d) > 0$, for 3-quantifier problems. In this direction, we have given an $O(n^{2-\frac{1}{2^d}})$-time algorithm for the central VCND problem where one set of vectors has dimension 2 and the other has dimension $d$. Note that by our results, such an algorithm for the general $\mathsf{VCND}_d$ problem would refute the 3-uniform HyperClique conjecture. Can we classify which problems admit such improved algorithms for small dimensions?

## References

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

**2** Amir Abboud, Karl Bringmann, Holger Dell, and Jesper Nederlof. More consequences of falsifying SETH and the orthogonal vectors conjecture. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 253–266. ACM, 2018.

**3** Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, apsp and diameter. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1681–1697. SIAM, 2014.

**4** Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 218–230. SIAM, 2015.

**5** Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 377–391. SIAM, 2016. doi:10.1137/1.9781611974331.ch28.

**6** Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *International Colloquium on Automata, Languages, and Programming*, pages 39–51. Springer, 2014.

**7** Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of databases: the logical level*. Addison-Wesley Longman Publishing Co., Inc., 1995.

**8** Pankaj K. Agarwal. Range searching. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 40. CRC Press LLC, 3rd edition, 2017.

**9** Udit Agarwal and Vijaya Ramachandran. Fine-grained complexity for sparse graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 239–252, 2018.

**10** Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.

**11** Stephen Alstrup, Gerth Stølting Brodal, and Theis Rauhe. New data structures for orthogonal range searching. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 198–207. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892088.

**12** Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 51–58. ACM, 2015.

**13** Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Towards tight approximation bounds for graph diameter and eccentricities. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 267–280. ACM, 2018.

**14** Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless seth fails. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 661–670. IEEE, 2014.

**15** Karl Bringmann, Nick Fischer, and Marvin Künnemann. A fine-grained analogue of schaefer's theorem in P: dichotomy of exists^k-forall-quantified first-order graph properties. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 31:1–31:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.31.

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

**17** 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 Madhu Sudan, editor, *Proceedings of the 2016 ACM*

*Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270. ACM, 2016. doi:10.1145/2840728.2840746.

18 Timothy M. Chan. Orthogonal range searching in moderate dimensions: k-d trees and range trees strike back. *Discret. Comput. Geom.*, 61(4):899–922, 2019.

19 Timothy M. Chan, Kasper Green Larsen, and Mihai Patrascu. Orthogonal range searching on the ram, revisited. In Ferran Hurtado and Marc J. van Kreveld, editors, *Proceedings of the 27th ACM Symposium on Computational Geometry, Paris, France, June 13-15, 2011*, pages 1–10. ACM, 2011. doi:10.1145/1998196.1998198.

20 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. *JoCG*, 10(1):27–41, 2019. doi:10.20382/jocg.v10i1a2.

21 Timothy M Chan and Ryan Williams. Deterministic APSP, Orthogonal Vectors, and More: Quickly derandomizing Razborov-Smolensky. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1246–1255. SIAM, 2016.

22 Bernard Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17(3):427–462, 1988. doi:10.1137/0217026.

23 Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 21–40. SIAM, 2019. doi:10.1137/1.9781611975482.2.

24 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for eth-tight algorithms and lower bounds in geometric intersection graphs. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 574–586. ACM, 2018. doi:10.1145/3188745.3188854.

25 Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. *Orthogonal Range Searching*, pages 95–120. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. doi:10.1007/978-3-662-04245-8_5.

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

27 Jiawei Gao. On the fine-grained complexity of least weight subsequence in multitrees and bounded treewidth dags. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, volume 148 of *LIPIcs*, pages 16:1–16:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.IPEC.2019.16.

28 Jiawei Gao and Russell Impagliazzo. The fine-grained complexity of strengthenings of first-order logic. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:9, 2019. URL: https://eccc.weizmann.ac.il/report/2019/009.

29 Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 2162–2181, 2017.

30 Neil Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999. doi:10.1007/978-1-4612-0539-5.

31 Russell Impagliazzo, Shachar Lovett, Ramamohan Paturi, and Stefan Schneider. 0-1 integer linear programming with a linear number of constraints. *CoRR*, abs/1401.5512, 2014. arXiv:1401.5512.

32 Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A satisfiability algorithm for sparse depth-2 threshold circuits. *CoRR*, abs/1212.4548, 2012. arXiv:1212.4548.

33 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 653–662. IEEE, 1998.

**34** Marvin Künnemann and Dániel Marx. Finding small satisfying assignments faster than brute force: A fine-grained perspective into boolean constraint satisfaction. In *Proc. 35th Computational Complexity Conference (CCC 2020)*, 2020. To appear.

**35** 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)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:15, 2017.

**36** Andrea Lincoln, Virginia Vassilevska Williams, and Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1236–1252. Society for Industrial and Applied Mathematics, 2018.

**37** George S. Lueker. A data structure for orthogonal range queries. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 28–34. IEEE Computer Society, 1978. `doi:10.1109/SFCS.1978.1`.

**38** Dániel Marx and Anastasios Sidiropoulos. The limited blessing of low dimensionality: when 1-1/d is the best possible exponent for d-dimensional geometric problems. In Siu-Wing Cheng and Olivier Devillers, editors, *30th Annual Symposium on Computational Geometry, SOCG'14, Kyoto, Japan, June 08 - 11, 2014*, page 67. ACM, 2014. `doi:10.1145/2582112.2582124`.

**39** Daniel Moeller, Ramamohan Paturi, and Stefan Schneider. Subquadratic algorithms for succinct stable matching. In *International Computer Science Symposium in Russia*, pages 294–308. Springer, 2016.

**40** Yakov Nekrich. Orthogonal range searching in linear and almost-linear space. *Comput. Geom.*, 42(4):342–351, 2009. `doi:10.1016/j.comgeo.2008.09.001`.

**41** Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In *SODA*, volume 10, pages 1065–1075. SIAM, 2010.

**42** D.E. Willard. *Predicate-oriented Database Search Algorithms*. Center for Research in Computing Technology: Center for Research in Computing Technology. Garland Pub., 1979. URL: `https://books.google.de/books?id=iLQmAAAAMAAJ`.

**43** Ryan Williams. Faster decision of first-order graph properties. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, page 80. ACM, 2014.

**44** Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 645–654. IEEE, 2010.

**45** Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013. `doi:10.1137/09076619X`.