Lower Bounds for Semialgebraic Range Searching and Stabbing Problems

Peyman Afshani ✉
Aarhus University, Denmark

Pingan Cheng ✉
Aarhus University, Denmark

Abstract

In the semialgebraic range searching problem, we are given a set of n points in $\mathbb{R}^d$ and we want to preprocess the points such that for any query range belonging to a family of constant complexity semialgebraic sets (Tarski cells), all the points intersecting the range can be reported or counted efficiently. When the ranges are composed of simplices, then the problem is well-understood: it can be solved using $S(n)$ space and with $Q(n)$ query time with $S(n)Q^d(n) = \tilde{O}(n^d)$ where the $\tilde{O}(\cdot)$ notation hides polylogarithmic factors and this trade-off is tight (up to $n^{o(1)}$ factors). Consequently, there exists "low space" structures that use $O(n)$ space with $O(n^{1-1/d})$ query time and "fast query" structures that use $O(n^d)$ space with $O(\log^{d+1} n)$ query time. However, for the general semialgebraic ranges, only "low space" solutions are known, but the best solutions match the same trade-off curve as the simplex queries, with $O(n)$ space and $\tilde{O}(n^{1-1/d})$ query time. It has been conjectured that the same could be done for the "fast query" case but this open problem has stayed unresolved.

Here, we disprove this conjecture. We give the first nontrivial lower bounds for semilagebraic range searching and other related problems. More precisely, we show that any data structure for reporting the points between two concentric circles, a problem that we call 2D annulus reporting problem, with $Q(n)$ query time must use $S(n) = \overset{o}{\Omega}(n^3/Q(n)^5)$ space where the $\overset{o}{\Omega}(\cdot)$ notation hides $n^{o(1)}$ factors, meaning, for $Q(n) = O(\log^{O(1)} n)$, $\overset{o}{\Omega}(n^3)$ space must be used. In addition, we study the problem of reporting the subset of input points between two polynomials of the form $Y = \sum_{i=0}^{\Delta} a_i X^i$ where values $a_0, \cdots, a_\Delta$ are given at the query time, a problem that we call polynomial slab reporting. For this, we show a space lower bound of $\overset{o}{\Omega}(n^{\Delta+1}/Q(n)^{\Delta^2+\Delta})$, which shows for $Q(n) = O(\log^{O(1)} n)$, we must use $\overset{o}{\Omega}(n^{\Delta+1})$ space. We also consider the dual problems of semialgebraic range searching, semialgebraic stabbing problems, and present lower bounds for them. In particular, we show that in linear space, any data structure that solves 2D annulus stabbing problems must use $\Omega(n^{2/3})$ query time. Note that this almost matches the upper bound obtained by lifting 2D annuli to 3D. Like semialgebraic range searching, we also present lower bounds for general semialgebraic slab stabbing problems. Again, our lower bounds are almost tight for linear size data structures in this case.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Computational Geometry, Data Structures and Algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2021.8

Related Version *Full Version*: `https://arxiv.org/abs/2012.00704`

Funding Supported by DFF (Det Frie Forskningsråd) of Danish Council for Independent Research under grant ID DFF–7014–00404.

Acknowledgements The authors would like to thank Esther Ezra for sparking the initial ideas behind the proof.

1 Introduction

We address one of the biggest open problems of the recent years in the range searching area. Our main results are lower bounds in the pointer machine model of computation that essentially show that the so-called "fast query" version of the semialgebraic range reporting problem is "impervious" to the algebraic techniques. Our main result reveals that to obtain polylogarithmic query time, the data structure requires $\overset{o}{\Omega}(n^{\Delta+1})$ space[1], where the constant depends on Δ, n is the input size, and $\Delta+1$ is the number of parameters of each "polynomial inequality" (these will be defined more clearly later). Thus, we refute a relatively popular recent conjecture that data structures with $\overset{o}{O}(n^d)$ space and polylogarithmic query time could exist, where d is the dimension of the input points. Surprisingly, the proofs behind these lower bounds are simple, and these lower bounds could have been discovered years ago as the tools we use already existed decades ago.

Range searching is a broad area of research in which we are given a set P of n points in $\mathbb{R}^d$ and the goal is to preprocess P such that given a query range $\mathcal{R}$, we can count or report the subset of P that lies in $\mathcal{R}$. Often $\mathcal{R}$ is restricted to a fixed family of ranges, e.g., in simplex range counting problem, $\mathcal{R}$ is a simplex in $\mathbb{R}^d$ and the goal is to report $|P \cap \mathcal{R}|$, or in halfspace range reporting problem, $\mathcal{R}$ is a halfspace and the goal is to report $P \cap \mathcal{R}$. Range searching problems have been studied extensively and they have numerous variants. For an overview of this topic, we refer the readers to an excellent survey by Agarwal [17].

Another highly related problem which can be viewed as the "dual" of this problem is range stabbing: we are given a set R of ranges as input and the goal is to preprocess R such that given a query point p, we can count or report the ranges of R containing p efficiently. Here, we focus on the reporting version of range stabbing problems.

1.1 Range Searching: A Very Brief Survey

1.1.1 Simplex Range Searching

Simplices is one of the most fundamental family of queries. In fact, if the query is decomposable (such as range counting or range reporting queries), then simplices can be used as "building blocks" to answer more complicated queries: for a query $\mathcal{R}$ which is a polyhedral region of $O(1)$ complexity, we can decompose it into $O(1)$ disjoint simplices (with a constant that depends on d) and thus answering $\mathcal{R}$ can be reduced to answering $O(1)$ simplicial queries.

Simplicial queries were hotly investigated in 1980s and this led to development of two important tools in computational geometry: cuttings and partition theorem and both of them have found applications in areas not related to range searching.

1.1.1.1 Cuttings and Fast Data Structures

"Fast query" data structures can answer simplex range counting or reporting queries in polylogarithmic query time but by using $O(n^d)$ space and they can be built using cuttings. In a nut-shell, given a set H of n hyperplanes in $\mathbb{R}^d$, a $\frac{1}{r}$-cutting, is a decomposition of $\mathbb{R}^d$ into $O(r^d)$ simplices such that each simplex is intersected by $O(n/r)$ hyperplanes of H. These were developed by some of the pioneers in the range searching area, such as Clarkson [15], Haussler and Welzl [19], Chazelle and Friedman [8], Matoušek [20], finally culminating in a

---

[1] $\overset{o}{\Omega}(\cdot)$, $\overset{o}{O}(\cdot)$, $\overset{o}{\Theta}(\cdot)$ notations hide $n^{o(1)}$ factors and $\tilde{\Omega}(\cdot)$, $\tilde{O}(\cdot)$, $\tilde{\Theta}(\cdot)$ notations hide $\log^{O(1)} n$ factors.

result of Chazelle [11] who optimized various aspects of cuttings. Using cuttings, one can answer simplex range counting, or reporting queries with $O(n^d)$ space and $O((\log n)^{d+1} + k)$ query time (where k is the output size) [21]. The query time can be lowered to $O(\log n)$ by increasing the space slightly to $O(n^{d+\varepsilon})$ for any constant $\varepsilon > 0$ [14]. An interested reader can refer to a book on cuttings by Chazelle [12].

1.1.1.2 The Partition Theorem and Space-efficient Data Structures

At the opposite end of the spectrum, simplex range counting or reporting queries can be answered using linear space but with higher query time of $O(n^{1-1/d})$, using partition trees and the related techniques. This branch of techniques has a very interesting history. In 1982, Willard [24] cleverly used ham sandwich theorem to obtain a linear-sized data structure with query time of $O(n^{\gamma})$ for some constant $\gamma < 1$ for simplicial queries in 2D. After a number of attempts that either improved the exponent or generalized the technique to higher dimensions, Welzl [23] in 1982 provided the first optimal exponent for the partition trees, then Chazelle et al. [14] provided the first near-linear size data structure with query time of roughly $O(n^{1-1/d})$. Finally, a data structure with $O(n)$ space and $O(n^{1-1/d})$ query time was given by Matoušek [21]. This was also simplified recently by Chan [7].

1.1.1.3 Space/Query Time Trade-off

It is possible to combine fast query data structures and linear-sized data structures to solve simplex queries with $S(n)$ space and $Q(n)$ query time such that $S(n)Q(n)^d = \tilde{O}(n^d)$. This trade-off between space and query time is optimal, at least in the pointer machine model and in the semigroup model [1, 13, 9].

1.1.1.4 Multi-level Structures, Stabbing and Other Related Queries

By using multi-level data structures, one can solve more complicated problems where both the input and the query shapes can be simplicial objects of constant complexity. The best multi-level data structures use one extra $\log n$ factor in space and query time per level [7] and there exist lower bounds that show space/query time trade-off should blow up by at least $\log n$ factor per level [2]. This means that problems such as simplex stabbing (where the input is a set of simplices and we want to output the simplices containing a given query point) or simplex-simplex containment problem (where the input is a set of simplices, and we want to output simplices fully contained in a query simplex) all have the same trade-off curve of $S(n)Q(n)^d = \tilde{O}(n^d)$ between space $S(n)$ and query time $Q(n)$.

Thus, one can see that the simplex range searching as well as its generalization to problems where both the input and the query ranges are "flat" objects is very well understood. However, there are many natural query ranges that cannot be represented using simplices, e.g., when query ranges are spheres in $\mathbb{R}^d$. This takes us to semialgebraic range searching.

1.1.2 Semialgebraic Range Searching

A semialgebraic set is defined as a subset of $\mathbb{R}^d$ that can be described as the union or intersection of $O(1)$ ranges, where each range is defined by d-variate polynomial inequality of degree at most Δ, defined by at most B values given at the query time; we call B the *parametric dimension*. For instance, with $B = 3$, $\Delta = 2$, and given three values a, b and c at the query time, a circular query can be represented as $\left\{(X, Y) \in \mathbb{R}^2 | (X - a)^2 + (Y - b)^2 \leq c^2\right\}$. In semialgebraic range searching, the queries are semialgebraic sets.

Before the recent "polynomial method revolution", the tools available to deal with semialgebraic range searching were limited, at least compared to the simplex queries. One way to deal with semialgebraic range searching is through linearization [25]. This idea maps the input points to $\mathbb{R}^L$, for some potentially large parameter L, such that each polynomial inequality can be represented as a halfspace. Consequently, semialgebraic range searching can be solved with the space/query time trade off of $S(n)Q(n)^L = \tilde{O}(n^L)$. The exponent of $Q(n)$ in the trade-off can be improved (increased) a bit by exploiting that in $\mathbb{R}^L$, the input set actually lies in a d-dimensional surface [5]. It is also possible to build "fast query" data structures but using $O(n^{2B-4+\varepsilon})$, but only in specific cases [5] (see [17] for details).

In 2009, Zeev Dvir [16] proved the discrete Kakeya problem with a very elegant and simple proof, using a polynomial method. Within a few years, this led to revolution in discrete and computational geometry, one that was ushered in by Katz and Guth's almost tight bound on Erdős distinct distances problem [18]. For a while, the polynomial method did not have much algorithmic consequences but this changed with the work of Agarwal, Matoušek, and Sharir [6] where they showed that at least as long as linear-space data structures are considered, semialgebraic range queries can essentially be solved within the same time as simplex queries (ignoring some lower order terms). Later developments (and simplifications) of their approach by Matoušek and Patáková [22] lead to the current best results: a data structure with linear size and with query time of $\tilde{O}(n^{1-1/d})$.

1.1.2.1 Fast Queries for Semialgebraic Range Searching: an Open Problem

Nonetheless, despite the breakthrough results brought on by the algebraic techniques, the fast query case still remained unsolved, even in the plane: e.g., the best known data structures for answering circular queries with polylogarithmic query time still use $\tilde{O}(n^3)$ space, by using linearization to $\mathbb{R}^3$. The fast query case of semialgebraic range searching has been explicitly mentioned as a major open problem in multiple recent publications[2]. In light of the breakthrough result of Agarwal et al. [6], it is quite reasonable to conjecture that semialgebraic range searching should have the same trade-off curve of $S(n)Q(n)^d = \tilde{O}(n^d)$.

Nonetheless, the algebraic techniques have failed to make sufficient advances to settle this open problem. The best known result is given recently by Agarwal et al. [4]. They showed it is possible to build "fast query" semialgebraic range searching data structures using $O(n^{B+\epsilon})$ space. In general, B can be much larger than d and thus it leaves a big gap between current best upper bound and the conjectured one. Given that it took a revolution caused by the polynomial method to advance our knowledge of the "low space" case of semialgebraic range searching, it is not too outrageous to imagine that perhaps equally revolutionary techniques are needed to settle the "fast query" case of semialgebraic range searching.

1.1.3 Semialgebraic Range Stabbing

Another important problem is semialgebraic stabbing, where the input is a set of n semialgebraic sets, i.e., "ranges", and queries are points. The goal is to output the input ranges that contain a query point. Here, sometimes "fast query" data structures are possible, for example by observing that an arrangements of n disks in the plane has $O(n^2)$ complexity

[2] To quote Agarwal et al. [6],"[a] very interesting and challenging problem is, in our opinion, the fast-query case of range searching with constant-complexity semialgebraic sets, where the goal is to answer a query in $O(\log n)$ time using roughly $O(n^d)$ space." The same conjecture is repeated in a different survey [3] and it is also emphasized that the question is even open for disks in the plane, "... whether a disk range-counting query in $\mathbb{R}^2$ be answered in $O(\log n)$ time using $O(n^2)$ space?".

and thus counting or reporting the disks stabbed by a query point can be done with $O(n^2)$ space and $O(\log n)$ query time. However, it seems difficult to make advancements in the "low space" side of things; the only known data structure with $O(n)$ space is one that uses linearization to 3D that results in $\tilde{O}(n^{2/3})$ query time.

1.2 Our Results

Our main results are lower bounds in the pointer machine model of computation for four central problems defined below. In the *2D polynomial slab reporting* problem, given a set $\mathcal{P}$ of n points in $\mathbb{R}^2$, the task is to preprocess $\mathcal{P}$ such that given a query 2D polynomial slab $\mathcal{R}$, the points contained in the polynomial slab, i.e., $\mathcal{R} \cap \mathcal{P}$, can be reported efficiently. Informally, a 2D polynomial slab is the set of points (x, y) such that $P(x) \le y \le P(x) + w$, for some univariate polynomial $P(x)$ of degree Δ and value w given at the query time. In the *2D polynomial slab stabbing* problem, the input is a set of n polynomial slabs and the query is a point q and the goal is to report all the slabs that contain q. Similarly, in the *2D annulus reporting* problem, the input is a set P of n points in $\mathbb{R}^2$ and the query is a "annulus", the region between two concentric circles. Finally, in *2D annulus stabbing* problem, the input is a set of n annuli, the query is a point q and the goal is to report all the annuli that contain q.

For polynomial slab queries, we show that if a data structure answers queries in $Q(n)+O(k)$ time, where k is the output size, using $S(n)$ space, then $S(n) = \mathring{\Omega}(n^{\Delta+1}/Q(n)^{\Delta^2+\Delta})$; the hidden constants depend on Δ. So for "fast queries", i.e., $Q(n) = \tilde{O}(1)$, $\mathring{\Omega}(n^{\Delta+1})$ space must be used. This is *almost tight* as the exponent matches the upper bounds obtained by linearization as well as the recent upper bound of Agarwal et al. [4]! Also, we prove that any structure that answers polynomial slab reporting queries in $Q(n) + O(k)$ time must use $\Omega(n^{1+1/\Delta}/Q(n)^{(\Delta+1)/\Delta^2})$ space. In the "low space" setting, when $S(n) = O(n)$, this gives $Q(n) = \Omega(n^{1-1/(\Delta+1)})$. This is once again *almost tight*, as it matches the upper bounds obtained by linearization for when $S(n) = \tilde{O}(n)$.

For the annulus reporting problem, our bound sharpens to $S(n) = \mathring{\Omega}(n^3/Q(n)^5)$. For the annulus stabbing problem, we show $S(n) = \Omega(n^{3/2}/Q(n)^{3/4})$, e.g., in "low space" setting when $S(n) = O(n)$, we must have $Q(n) = \Omega(n^{2/3})$; compare this with simplex stabbing queries can be solved with $O(n)$ space and $\tilde{O}(\sqrt{n})$ query time. As before, this is almost tight, as it matches the upper bounds obtained by linearization to 3D for when $S(n) = \tilde{O}(n)$.

Somewhat disappointedly, no revolutionary new technique is required to obtain these results. We use novel ideas in the construction of "hard input instances" but otherwise we use the two widely used pointer machine lower bound frameworks by Chazelle [10], Chazelle and Rosenberg [13], and Afshani [1]. Our results are summarized in Table 1.

2 Preliminaries

We first review the related geometric reporting data structure lower bound frameworks. The model of computation we consider is (an augmented version of) the pointer machine model.

In this model, the data structure is a directed graph M. Let $\mathcal{S}$ be the set of input elements. Each cell of M stores an element of $\mathcal{S}$ and two pointers to other cells. Assume a query q requires a subset $\mathcal{S}_q \subset \mathcal{S}$ to be output. For the query, we only charge for the pointer navigations. Let M_q be the smallest connected subgraph, s.t., every element of $\mathcal{S}_q$ is stored in at least one element of M_q. Clearly, $|M|$ is a lower bound for space and $|M_q|$ is a lower bound for query time. Note that this grants the algorithm unlimited computational power as well as full information about the structure of M.

In this model, there are two main lower bound frameworks, one for range reporting [10, 13], and the other for its dual, range stabbing [1]. We describe them in detail here.

2.1 A Lower Bound Framework for Range Reporting Problems

The following result by Chazelle [10] and later Chazelle and Rosenberg [13] provides a general lower bound framework for range reporting problems. In the problem, we are given a set $\mathcal{S}$ of n points in $\mathbb{R}^d$ and the queries are from a set $\mathcal{R}$ of ranges. The task is to build a data structure such that given any query range $\mathcal{R} \in \mathcal{R}$, we can report the points intersecting the range, i.e., $\mathcal{R} \cap \mathcal{S}$, efficiently.

▶ **Theorem 1** (Chazelle [10] and Chazelle and Rosenberg [13]). *Suppose there is a data structure for range reporting problems that uses at most $S(n)$ space and can answer any query in $Q(n) + O(k)$ time where n is the input size and k is the output size. Assume we can show that there exists an input set $\mathcal{S}$ of n points satisfying the following: There exist m subsets $q_1, q_2, \cdots, q_m \subset \mathcal{S}$, where $q_i, i = 1, \cdots, m$, is the output of some query and they satisfy the following two conditions: (i) for all $i = 1, \cdots, m$, $|q_i| \geq Q(n)$; and (ii) the size of the intersection of every α distinct subsets $q_{i_1}, q_{i_2}, \cdots, q_{i_\alpha}$ is bounded by some value $c \geq 2$, i.e., $|q_{i_1} \cap q_{i_2} \cap \cdots \cap q_{i_\alpha}| \leq c$. Then $S(n) = \Omega(\frac{\sum_{i=1}^{m} |q_i|}{\alpha 2^{O(c)}}) = \Omega(\frac{mQ(n)}{\alpha 2^{O(c)}})$.*

To use this framework, we need to exploit the property of the considered problem and come up with a construction that satisfies the two conditions above. Often, the construction is randomized and thus one challenge is to satisfy condition (ii) in the worst-case. This can be done by showing that the probability that (ii) is violated is very small and then using a union bound to prove that with positive probability the construction satisfies (ii) in the worst-case.

Table 1 Our Results, $*$ indicates this paper. In the table, $\mathring{\Omega}(\cdot)$ and $\mathring{O}(\cdot)$ notations hide $n^{o(1)}$ factors, and $\tilde{O}(\cdot)$ notation hides $\log^{O(1)} n$ factors.

Problem	**Lower Bound**	**Upper Bound**
2D Polynomial Slab Reporting	$S(n) = \mathring{\Omega}\left(\frac{n^{\Delta+1}}{Q(n)^{\Delta^2+\Delta}}\right)^*$	$S(n) = \tilde{O}\left(\frac{n^{\Delta+1}}{Q(n)^{2\Delta}}\right)$ [5, 6, 21]
When $Q(n) = \mathring{O}(1)$	$\boldsymbol{S(n) = \mathring{\Omega}\left(n^{\Delta+1}\right)}^*$	$\boldsymbol{S(n) = \mathring{O}\left(n^{\Delta+1}\right)}$ [5, 6, 21]
2D Annulus Reporting	$S(n) = \mathring{\Omega}\left(\frac{n^3}{Q(n)^5}\right)^*$	$S(n) = \tilde{O}\left(\frac{n^3}{Q(n)^4}\right)$ [5, 6, 21]
When $Q(n) = \mathring{O}(1)$	$\boldsymbol{S(n) = \mathring{\Omega}\left(n^3\right)}^*$	$\boldsymbol{S(n) = \mathring{O}\left(n^3\right)}$ [5, 6, 21]
2D Polynomial Slab Stabbing	$S(n) = \Omega\left(\frac{n^{1+1/\Delta}}{Q(n)^{(\Delta+1)/\Delta^2}}\right)^*$	$S(n) = \tilde{O}\left(\frac{n^2}{Q(n)^{(\Delta+1)/\Delta}}\right)^a$
When $S(n) = \mathring{O}(n)$	$\boldsymbol{Q(n) = \mathring{\Omega}\left(n^{1-1/(\Delta+1)}\right)}^*$	$\boldsymbol{S(n) = \mathring{O}\left(n^{1-1/(\Delta+1)}\right)}$
2D Annulus Stabbing	$S(n) = \Omega\left(\frac{n^{3/2}}{Q(n)^{3/4}}\right)^*$	$S(n) = \tilde{O}\left(\frac{n^2}{Q(n)^{3/2}}\right)^b$
When $S(n) = \mathring{O}(n)$	$\boldsymbol{Q(n) = \mathring{\Omega}\left(n^{2/3}\right)}^*$	$\boldsymbol{Q(n) = \mathring{O}\left(n^{2/3}\right)}$

[a] The subdivision formed by n degree Δ polynomial slabs has complexity $O(n^2)$ (for some constant depending on Δ). We partition the subdivision into vertical strips where for any strip any slab intersecting it fully span the strip and the number of slab changes of adjacent strips is $O(1)$. Consider these strips from left to right, we are solving a special dynamic slab stabbing problem. We can solve this problem by building a persistent interval tree using $O(n^2)$ space that answers each query in time $O(\log n + k)$. On the other hand, we can solve the problem in $O(n)$ space and $O(n^{1-1/(\Delta+1)} + k)$ time by linearization. Combining these two solutions using [21] gives the tradeoff.

[b] Similar to 2D polynomial slab stabbing.

2.2 A Lower Bound Framework for Range Stabbing Problems

Range stabbing problems can be viewed as the dual of range reporting problems. In this problem, we are given a set $\mathcal{R}$ of n ranges, and the queries are from a set $\mathcal{Q}$ of n points. The task is to build a data structure such that given any query point $q \in \mathcal{Q}$, we can report the ranges "stabbed" by this query point, i.e., $\{\mathcal{R} \in \mathcal{R} : \mathcal{R} \cap q \neq \emptyset\}$, efficiently. A recent framework by Afshani [1] provides a simple way to get the lower bound of such problems.

▶ **Theorem 2** (Afshani [1]). *Suppose there is a data structure for range stabbing problems that uses at most $S(n)$ space and can answer any query in $Q(n) + O(k)$ time where n is the input size and k is the output size. Assume we can show that there exists an input set $R \subset \mathcal{R}$ of n ranges that satisfy the following: (i) every query point of the unit square U is contained in at least $t \geq Q(n)$ ranges; and (ii) the area of the intersection of every $\alpha < t$ ranges is at most v. Then $S(n) = \Omega(\frac{t}{v 2^{O(\alpha)}}) = \Omega(\frac{Q(n)}{v 2^{O(\alpha)}})$.*

This is very similar to framework of Theorem 1 but often it requires no derandomization.

3 2D Polynomial Slab Reporting and Stabbing

We first consider the case when query ranges are 2D polynomial slabs. The formal definition of 2D polynomial slabs is as follows.

▶ **Definition 3.** *Let $P(x) = \sum_{i=0}^{\Delta} a_i x^i$, where $a_\Delta \neq 0$, be a degree Δ univariate polynomial. A 2D polynomial slab is a pair $(P(x), w)$, where $P(x)$ is called the base polynomial and $w > 0$ the width of the polynomial slab. The polynomial slab is then defined as $\{(x, y) \in \mathbb{R}^2 : P(x) \leq y \leq P(x) + w\}$.*

3.1 2D Polynomial Slab Reporting

We consider the 2D polynomial slab reporting problem in this section, where the input is a set $\mathcal{P}$ of n points in $\mathbb{R}^2$, and the query is a polynomial slab. This is an instance of semialgebraic range searching where we have two polynomial inequalities where each inequality has degree Δ and it is defined by $\Delta+1$ parameters given at the query time (thus, $B = \Delta+1$). Note that $\Delta+1$ is also the dimension of linearization for this problem, meaning, the 2D polynomial slab reporting problem can be lifted to the simplex range reporting problem in $\mathbb{R}^{\Delta+1}$. Our main result shows that for fast queries (i.e., when the query time is polylogarithmic), this is tight, by showing an $\overset{o}{\Omega}(n^{\Delta+1})$ space lower bound, in the pointer machine model of computation.

To do that, we will use Chazelle's framework. In our construction of a hard input instance, a derandomization process will be needed. We do this using the following two general lemmas. For the proofs of these lemmas, see the full version of the paper.

▶ **Lemma 4.** *Let $\mathcal{P}$ be a set of n points chosen uniformly at random in a square S of side length n in $\mathbb{R}^2$. Let $\mathcal{R}$ be a set of ranges in S such that (i) the intersection area of any $t \geq 2$ ranges $\mathcal{R}_1, \mathcal{R}_2, \cdots \mathcal{R}_t \in \mathcal{R}$ is bounded by $O\left(n/2^{\sqrt{\log n}}\right)$; (ii) the total number of intersections is bounded by $O\left(n^{2k}\right)$ for $k \geq 1$. Then with probability $> \frac{1}{2}$, for all distinct ranges $\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_t \in \mathcal{R}$, $|\mathcal{R}_1 \cap \mathcal{R}_2 \cap \cdots \mathcal{R}_t \cap \mathcal{P}| < 3k\sqrt{\log n}$.*

▶ **Lemma 5.** *Let $\mathcal{P}$ be a set of n points chosen uniformly at random in a square S of side length n in $\mathbb{R}^2$. Let $\mathcal{R}$ be a set of ranges in S such that (i) the intersection area of any range $\mathcal{R} \in \mathcal{R}$ and S is at least cnt for some constant $c \geq 4k$ and a parameter $t \geq \log n$, where $k \geq 2$; (ii) the total number of ranges is bounded by $O\left(n^{k+1}\right)$. Then with probability $> \frac{1}{2}$, for every range $\mathcal{R} \in \mathcal{R}$, $|\mathcal{R} \cap \mathcal{P}| \geq t$.*

Given a univariate polynomial $P(x)$, the following simple lemma establishes the relationship between the coefficient of the maximum degree term and the maximum range within which its value is bounded. This lemma will be used to upper bound the intersection area of two polynomial slabs. For the proof of this lemma, see the full version of the paper.

▶ **Lemma 6.** *Let $P(x) = \sum_{i=0}^{\Delta} a_i x^i$ be a degree Δ univariate polynomial where $\Delta > 0$ and $|a_\Delta| \geq d$ for some positive d. Let w be any positive value and x_l be a parameter. If $|P(x)| \leq w$ for all $x \in [x_l, x_l + t]$, then $t \leq (\Delta+1)^3 \left(\frac{w}{d}\right)^{\frac{1}{\Delta}}$.*

With Lemma 6 at hand, we now show a lower bound for polynomial slab reporting.

▶ **Theorem 7.** *Let $\mathcal{P}$ be a set of n points in $\mathbb{R}^2$. Let $\mathcal{R}$ be the set of all 2D polynomial slabs $\{(P(x), w) : \deg(P) = \Delta \geq 2, w > 0\}$. Then any data structure for $\mathcal{P}$ that solves polynomial slab reporting for queries from $\mathcal{R}$ with query time $Q(n) + O(k)$, where k is the output size, uses $S(n) = \overset{o}{\Omega}\left(n^{\Delta+1}/Q(n)^{\Delta^2+\Delta}\right)$ space.*

Proof. We use Chazelle's framework to prove this theorem. To this end, we will need to show the existence of a hard input instance. We do this as follows. In a square S, we construct a set of special polynomial slabs with the following properties: (i) The intersection area of any two slabs is small; and (ii) The area of each slab inside S is relatively large. Intuitively and consequently, if we sample n points uniformly at random in S, in expectation, few points will be in the intersection of two slabs, and many points will be in each slab. Intuitively, this satisfies the two conditions of Theorem 1. By picking parameters carefully and a derandomization process, we get our theorem. Next, we describe the details.

Consider a square $S = [0, n] \times [0, n]$. Let d, w be some parameters to be specified later. We generate a set of $\Theta\left(\left(\frac{n}{2d}\right)^{\Delta} \cdot \frac{n}{w}\right)$ polynomial slabs $(P(x), w)$ with

$$P(x) = \left(\sum_{i=1}^{\Delta} \frac{j_i d x^i}{n^i}\right) + kw$$

where $j_i = \lfloor \frac{n}{2d} \rfloor, \lfloor \frac{n}{2d} \rfloor + 1, \cdots, \lfloor \frac{n}{d} \rfloor$ for $1 \leq i \leq \Delta$ and $k = \lfloor \frac{n}{4w} \rfloor, \lfloor \frac{n}{4w} \rfloor + 1, \cdots, \lfloor \frac{n}{2w} \rfloor$. Note that we normalize the coefficients such that for any polynomial slab in range $x \in [0, n]$, a quarter of this slab is contained in S if $w < n/6$. To show this, it is sufficient to show that every polynomial is inside S, for every $x \in [0, n/4]$. As all the coefficients of the polynomials are positive, it is sufficient to upper bound $P(n/4)$, among all the polynomials $P(x)$ that we have generated. Similarly, this maximum is attained when all the coefficients are set to their maximum value, i.e., when $j_i = n/d$ and $k = n/(2w)$, resulting in the polynomial $P_u(x) = \left(\sum_{i=1}^{\Delta} x^i/n^{i-1}\right) + \frac{n}{2}$. Now it easily follows that $P_u(n/4) < 5n/6$. Then, the claim follows from the following simple observation.

▶ **Observation 8.** *The area of a polynomial slab $\{P(x), w\}$ for when $a \leq x \leq b$ is $(b-a)w$.*

Proof. The claimed area is $(\int_a^b (P(x) + w)dx) - (\int_a^b P(x)dx) = \int_a^b w dx = (b-a)w$. ◀

Next, we bound the area of the intersection of two polynomial slabs. Consider two distinct slabs $\mathcal{R}_p = (P(x), w)$ and $\mathcal{R}_q = (Q(x), w)$. Observe that by our construction, if $P(x)$ and $Q(x)$ only differ in their constant terms, their intersection is empty. So we only consider the case that there exists some $0 < i \leq \Delta$, such that the coefficients for x^i are different in $P(x)$ and $Q(x)$. As each slab is created using two polynomials of degree Δ, $\mathcal{R}_q \cap \mathcal{R}_p$ can have at most $O(\Delta)$ connected regions. Consider one connected region $\mathfrak{R}$ and let the interval $\eta = [x_1, x_2] \subset [0, n]$, be the projection of $\mathfrak{R}$ onto the X-axis. Define the polynomial $R(x) = P(x) - Q(x)$ and observe that we must have $|R(x)| \le w$ for all $x \in [x_1, x_2]$. We now consider the coefficient of the highest degree term of $R(x)$. Let $j_i d/n^i$ (resp. $j'_i d/n^i$) be the coefficient of the degree i term in $P(x)$ (resp. $Q(x)$). Clearly, if $j_i = j'_i$, then the coefficient of x^i in $R(x)$ will be zero. Thus, to find the highest degree term in $R(x)$, we need to consider the largest index i such that $j_i \neq j'_i$; in this case, $R(x)$ will have degree i and coefficient of x_i will have absolute value $\left|(j_i - j'_i)d/n^i\right| \ge d/n^i$. When $w \le d$, by Lemma 6, $x_2 - x_1 \le O(\Delta^3)\left(\frac{wn^i}{d}\right)^{1/i} \le O(\Delta^3)n\left(\frac{w}{d}\right)^{1/\Delta}$. Next, by Observation 8, the area of the intersection of $\mathcal{R}_q$ and $\mathcal{R}_p$ is $O(\Delta^3)nw\left(\frac{w}{d}\right)^{1/\Delta}$.

We pick $d = c\Delta^{3\Delta}w^{\Delta+1}2^{\Delta\sqrt{\log n}}$ and $w = 16\Delta Q(n)$, for a large enough constant c. Then, the intersection area of any two polynomial slabs is bounded by $n/2^{\sqrt{\log n}}$. Since in total we have generated $O(n^{\Delta+1})$ slabs, the total number of intersections they can form is bounded by $O(n^{2(\Delta+1)})$. By Lemma 4, with probability $> \frac{1}{2}$, the number of points of $\mathcal{P}$ in any intersection of two polynomial slabs is at most $3(\Delta+1)\sqrt{\log n}$. Also, as we have shown that the intersection area of every slab with S is at least $nw/4 = 4\Delta nQ(n)$, by Lemma 5, with probability more than $\frac{1}{2}$, each polynomial slab has at least $Q(n)$ points of $\mathcal{P}$.

It thus follows that with positive probability, both conditions of Theorem 1 are satisfied, and consequently, we obtain the lower bound of

$$S(n) = \Omega\left(\frac{Q(n)\cdot\left(\frac{n}{2d}\right)^{\Delta}\cdot\frac{n}{w}}{2^{3(\Delta+1)\sqrt{\log n}}}\right) = \overset{o}{\Omega}\left(\frac{n^{\Delta+1}}{Q(n)^{\Delta^2+\Delta}}\right). \qquad ◀$$

So for the "fast query" case data structure, by picking $Q(n) = O(\log^{O(1)} n)$, we obtain a space lower bound of $S(n) = \overset{o}{\Omega}(n^{\Delta+1})$.

3.2 2D Polynomial Slab Stabbing

By small modifications, our construction can also be applied to obtain a lower bound for (the reporting version of) polynomial slab stabbing problems using Theorem 2.

One modification is that we need to generate the slabs in such a way that they cover the entire square S. The framework provided through Theorem 2 is more stream-lined and derandomization is not needed and we can directly apply the "volume upper bound" obtained through Lemma 6. There is also no $n^{o(1)}$ factor loss (our lower bound actually uses $\Omega(\cdot)$ notation). The major change is that we need to use different parameters since we need to create n polygonal slabs, as now they are the input. For the details refer to the full version of the paper.

▶ **Theorem 9.** *Give a set $\mathcal{R}$ of n 2D polynomial slabs $\{(P(x), w) : deg(P) = \Delta \ge 2, w > 0\}$, any data structure for $\mathcal{R}$ solving the 2D polynomial slab stabbing problem with query time $Q(n) + O(k)$ uses $S(n) = \Omega\left(\frac{n^{1+\frac{1}{\Delta}}}{Q(n)^{\frac{\Delta+1}{\Delta^2}}}\right)$ space, where k is the output size.*

So for any data structure that solves the 2D polynomial slab stabbing problem using $S(n) = O(n)$ space, Theorem 9 implies that its query time must be $Q(n) = \Omega(n^{1-1/(\Delta+1)})$.

4 2D Annulus Reporting and Stabbing

4.1 2D Annulus Reporting

In this section, we show that any data structure that solves 2D annulus reporting with $O(\log^{O(1)} n)$ query time must use $\mathring{\Omega}(n^3)$ space. Recall that a annulus is the region between two concentric circles and the *width* of the annulus is the difference between the radii of the two circles. In general, we show that if the query time is $Q(n) + O(k)$, then the data structure must use $\mathring{\Omega}(n^3/Q(n)^5)$ space. Note that this is also a better trade-off curve than what we obtained for the polynomial slab reporting problem when $\Delta = 2$. We will still use Chazelle's framework.

We first present a technical geometric lemma which upper bounds the intersection area of two 2D annuli. We will later use this lemma to show that with probability more than 1/2, a random point set satisfies the first condition of Theorem 1.

▶ **Lemma 10.** *Consider two annuli of width w with inner radii of r_1, r_2, where $r_1 + w \le r_2, w < r_1$, and $r_1, r_2 = \Theta(n)$. Let d be the distance between the centers of two annuli. When $w \le d < r_2$, the intersection area of two annuli is bounded by $O\left(wn\sqrt{\frac{w^2}{(g+w)d}}\right)$, where $g = \max\{r_1 - r_2 + d, 0\}$.*

The proof sketch. For the complete proof see the full version of the paper. When $w \le d \le r_2 - r_1 + 2w$, the intersection region consists of two triangle-like regions. We only bound the triangle-like region $\tilde{\triangle}PQR$ in the upper half annuli as shown in Figure 1. We can show that its area is asymptotically upper bounded by the product of its base length $|QR| = O(w)$ and its height h. We bound h by observing that $\frac{hd}{2}$ is the area of triangle $\triangle PO_1O_2$ but we can also obtain its area of using Heron's formula, given its three side lengths. This gives $h = O(n\sqrt{w/d})$. Since in this case $g \le 2w$, the intersection area is upper bounded by $O\left(wn\sqrt{\frac{w^2}{(g+w)d}}\right)$ as claimed.

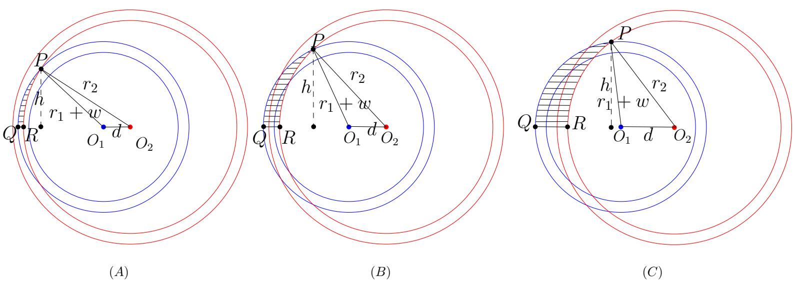

Figure 1 Intersections When d is Small.

When $r_2 - r_1 + 2w \le d \le r_2$, the intersection region consists of two quadrilateral-like regions. Again we only consider $\tilde{\square}ABCD$ in the upper half of the annuli, which is contained in a partial annulus, $\tilde{\mathcal{R}}_{EFHD}$, as shown in Figure 2a. We show the area of $\tilde{\mathcal{R}}_{EFHD}$ is asymptotically bounded by $|BH| \cdot w$, where $|BH|$ is the distance between the two endpoints of the inner arc. We upper bound $|BH|$ by $|BD|$. We use the algebraic representation of the two annuli, to bound the length of the projection of BD on the X-axis by $\Theta\left(\frac{wn}{d}\right)$; See Figure 2b. We use Heron's formula to bound the length of the projection of BD on the Y-axis by $O\left(n\sqrt{w^2/dg}\right)$. The maximum of the length of the two projections yields the claimed bound. ◀

(a) Cover an Intersection by A Partial Annulus

(b) Bound the Length of $|BD|$

Figure 2 Cover a Quadrilateral-like Region by a Partial Annulus.

We use Chazelle's framework to obtain a lower bound for 2D annulus reporting. Let S_1 and S_2 be two squares of side length n that are placed $10n$ distance apart and S_2 is directly to the left of S_1. We generate the annuli as follows. We divide S_1 into a $\frac{n}{T} \times \frac{n}{T}$ grid where each cell is a square of side length T. For each grid point, we construct a series of circles as follows. Let O be a grid point. The first circle generated for O must pass through a corner of S_2 and not intersect the right side of S_2, as shown in Figure 3. Then we create a series of circles centered at O by increasing the radius by increments of w, for some $w < T$, as long as it does not intersect the left side of S_2. Every consecutive two circles defines a annulus centered on O. We repeat this for every grid cell in S_1 and this makes up our set of queries. The input points are placed uniformly randomly inside S_2.

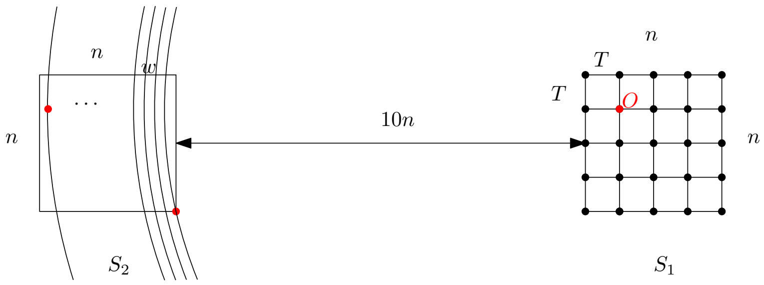

Figure 3 Generate a Family of Annuli at Point O.

We now show that for the annuli we constructed, the intersection of ℓ annuli is not too large, for some ℓ we specify later. More precisely we prove the following.

▶ **Lemma 11.** *There exists a large enough constant c such that in any subset of $\ell = cw^2/\sqrt{T}$ annuli, we can find two annuli such that their intersection has area $O\left(nw\sqrt{\frac{1}{T}}\right)$.*

The proof sketch. For the complete proof see the full version of the paper. Let $\mathcal{S}$ be a set of $\ell = cw^2/\sqrt{T}$ annuli. Suppose for the sake of contradiction that we cannot find two annuli in $\mathcal{S}$ whose intersection area is $O\left(nw\sqrt{\frac{1}{T}}\right)$. Since by Lemma 10, the intersection area of any two annuli in our construction with distance $\Omega(wT)$ is $O\left(nw\sqrt{\frac{1}{T}}\right)$. The maximum distance between any two annuli in $\mathcal{S}$ must be $o(wT)$.

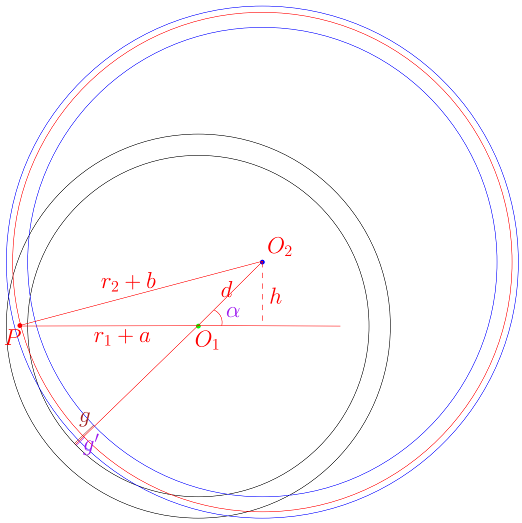

Figure 4 Intersection of Two Annuli.

Let P be a point in the intersection of annuli in $\mathcal{S}$. Consider an arbitrary annulus $\mathcal{R}_1 \in \mathcal{S}$ centered at O_1 and another annulus $\mathcal{R}_2 \in \mathcal{S}$ centered at O_2 for some $O_2 \notin PO_1$. For $\mathcal{R}_1, \mathcal{R}_2$ to contain P, we must have $|PO_1| = r_1 + a, |PO_2| = r_2 + b$ for $0 \leq a, b \leq w$. See Figure 4 for an example. Also $|O_1O_2| = d$, by exploiting the shape of $\triangle PO_1O_2$ and applying Lemma 10, we can compute an upper bound for the distance between O_2 and PO_1, namely, $h = d \sin \alpha = o(w\sqrt{T})$, where α is the angle between O_1O_2 and PO_1. This implies that $\mathcal{S}$ must fit in a rectangle of size $o(wT) \times o(w\sqrt{T})$. Since the gird cell size is $T \times T$, only $o(w^2/\sqrt{T})$ annuli are contained in such a rectangle, a contradiction. ◀

We are now ready to plug in some parameters in our construction. We set $T = w^2 2^{2\sqrt{\log n}}$. First, we claim that from each grid cell O, we can draw $\Theta(n/w)$ circles; Let C_1, C_2, C_3, and C_4 be the corners of S_2 sorted increasingly according to their distance to O. As S_1 and S_2 are placed $10n$ distance apart, an elementary geometric calculation reveals that C_1 and C_2 are vertices of the right edge of S_2, meaning, the smallest circle that we draw from O passes through C_2 and we keep drawing circles, by incrementing their radii by w until we are about to draw a circle that is about to contain C_3. We can see that $|OC_3| - |OC_2| = \Theta(n)$ and thus we draw $\Theta(n/w)$ circles from O. As we have $\Theta((n/T)^2)$ grid cells, it thus follows that we have $\Theta(n^3/(T^2w))$ annuli in our construction.

Also by our construction, the area of each annulus within S_2 is $\Theta(wn)$. To see this, let P be an arbitrary point in S_1, let A, B be the intersections of some circle centered at P as in Figure 5.

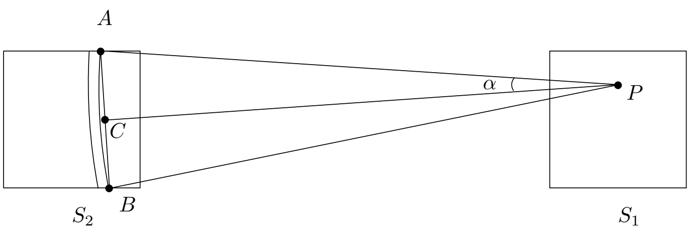

Figure 5 The Angle of a Annulus.

We connect AB and let C be the center of AB. Let $\alpha = \angle APC$. In the triangle $\triangle ABP$, all the sides are within constant factors of each other and thus $\alpha = \Theta(1)$ and so the area of the annulus inside S_2 is at least a constant fraction of the area of the entire annulus.

Suppose we have a data structure that answers 2D annulus reporting queries in $Q(n)+O(k)$ time. We set $w = c'Q(n)$ for a large enough constant c' such that the area of each annulus within S_2 is at least $\Theta(wn) > 8nQ(n)$. By Lemma 5, if we sample n points uniformly at random in S_2, then with probability more than $1/2$, each annulus contains at least $Q(n)$ points.

Also by our construction, the total number of intersections of two annuli is bounded by $O(n^6)$ and by our choice of T, $O\left(nw\sqrt{\frac{1}{T}}\right) = O\left(\frac{n}{2^{\sqrt{\log n}}}\right)$. Then by Lemma 4 and Lemma 11, with probability $> \frac{1}{2}$, a point set of size n picked uniformly at random in S_2 satisfies that the number of points in any of the intersection of $cw^2/\sqrt{T}$ annuli is no more than $9\sqrt{\log n}$.

Now by union bound, there exist $\Theta\left(\frac{n^3}{wT^2}\right)$ point sets such that each set is the output of some 2D annulus query and each set contains at least $Q(n)$ points. Furthermore, the intersection of any $cw^2/\sqrt{T}$ sets is bounded by $9\sqrt{\log n}$. Then by Theorem 1, we obtain a lower bound of

$$S(n) = \Omega\left(\frac{Q(n)n^3\sqrt{T}}{wT^2w^2 2^{O\left(\sqrt{\log n}\right)}}\right) = \overset{o}{\Omega}\left(\frac{n^3}{Q(n)^5}\right).$$

This proves the following theorem about 2D annulus reporting.

▶ **Theorem 12.** *Any data structure that solves 2D annulus reporting on point set of size n with query time $Q(n) + O(k)$, where k is the output size, must use $\overset{o}{\Omega}\left(n^3/Q(n)^5\right)$ space.*

So for any data structure that solves 2D annulus reporting in time $Q(n) = O(\log^{O(1)} n)$, Theorem 12 implies that $\overset{o}{\Omega}\left(n^3\right)$ space must be used.

4.2 2D Annulus Stabbing

Modifications similar to those done in Subsection 3.2 can be used to obtain the following lower bound. See the full version of the paper for details.

▶ **Theorem 13.** *Any data structure that solves the 2D annulus stabbing problem with query time $Q(n) + O(k)$, where k is the output size, must use $S(n) = \Omega(n^{3/2}/Q(n)^{3/4})$ space.*

So for any data structure that solves the 2D annulus stabbing problem using $O(n)$ space, Theorem 13 implies that its query time must be $Q(n) = \Omega(n^{2/3})$.

5 Conclusion and Open Problems

We investigated lower bounds for range searching with polynomial slabs and annuli in $\mathbb{R}^2$. We showed space-time tradeoff bounds of $S(n) = \overset{o}{\Omega}(n^{\Delta+1}/Q(n)^{\Delta^2+\Delta})$ and $S(n) = \overset{o}{\Omega}(n^3/Q(n)^5)$ for them respectively. Both of these bounds are almost tight in the "fast query" case, i.e., when $Q(n) = O(\log^{O(1)} n)$ (up to a $n^{o(1)}$ factor). This refutes the conjecture of the existence of data structure that can solve semialgebraic range searching in $\mathbb{R}^d$ using $\overset{o}{O}(n^d)$ space and $O(\log^{O(1)} n)$ query time. We also studied the "dual" polynomial slab stabbing and annulus stabbing problems. For these two problems, we obtained lower bounds $S(n) = \Omega(n^{1+1/\Delta}/Q(n)^{(\Delta+1)/\Delta^2})$ and $S(n) = \Omega(n^{3/2}/Q(n)^{3/4})$ respectively. These bounds are tight when $S(n) = O(n)$. Our work, however, brings out some very interesting open problems.

To get the lower bounds for the polynomial slabs, we only considered univariate polynomials of degree Δ. In this setting, the number of coefficients is at most $\Delta + 1$, and we have also assumed they are all independent. It would be interesting to see if similar lower bounds can be obtained under more general settings. In particular, as the maximum number of coefficients of a bivaraite polynomial of degree Δ is $\binom{\Delta+2}{2}$, it would interesting to see if a $\overset{o}{\Omega}(n^{\binom{\Delta+2}{2}-1})$ space lower bound can be obtained for the "fast query" case.

It would also be interesting to consider space-time trade-offs. For instance, by combining the known "fast query" and "low space" solutions for 2D annulus reporting, one can obtain data structures with trade-off curve $S(n) = \tilde{O}(n^3/Q(n)^4)$, however, our lower bound is $S(n) = \overset{o}{\Omega}(n^3/Q(n)^5)$ and it is not clear which of these bounds is closer to truth. For the annulus searching problem in $\mathbb{R}^2$, in our lower bound proof, we considered a random input point set, since in most cases a random point set is the hardest input instance and our analysis seems to be tight, we therefore conjecture that our lower bound could be tight, at least when $Q(n)$ is small enough. We believe that it should be possible to obtain the trade-off curve of $S(n) = \tilde{O}(n^3/Q(n)^5)$ when the input points are uniformly random in the unit square and $Q(n)$ is not too big.

Finally, another interesting direction is to study the lower bound for the counting variant of semialgebraic range searching.

References

1. Peyman Afshani. Improved pointer machine and I/O lower bounds for simplex range reporting and related problems. In *Proceedings of the Twenty-Eighth Annual Symposium on Computational Geometry*, SoCG '12, page 339–346, New York, NY, USA, 2012. Association for Computing Machinery. `doi:10.1145/2261250.2261301`.
2. Peyman Afshani and Anne Driemel. On the complexity of range searching among curves. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 898–917. SIAM, Philadelphia, PA, 2018. `doi:10.1137/1.9781611975031.58`.
3. Pankaj K. Agarwal. *Simplex Range Searching and Its Variants: A Review*, pages 1–30. Springer International Publishing, Cham, 2017. `doi:10.1007/978-3-319-44479-6_1`.
4. Pankaj K. Agarwal, Boris Aronov, Esther Ezra, and Joshua Zahl. An efficient algorithm for generalized polynomial partitioning and its applications. In *35th International Symposium on Computational Geometry*, volume 129 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 5, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.
5. Pankaj K. Agarwal and Jiří Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11(4):393–418, 1994. `doi:10.1007/BF02574015`.
6. Pankaj K. Agarwal, Jiří Matoušek, and Micha Sharir. On range searching with semialgebraic sets. II. *SIAM J. Comput.*, 42(6):2039–2062, 2013. `doi:10.1137/120890855`.
7. Timothy M. Chan. Optimal partition trees. *Discrete Comput. Geom.*, 47(4):661–690, 2012. `doi:10.1007/s00454-012-9410-z`.
8. B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3):229–249, 1990. `doi:10.1007/BF02122778`.
9. Bernard Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2(4):637–666, 1989. `doi:10.2307/1990891`.
10. Bernard Chazelle. Lower bounds for orthogonal range searching. I. The reporting case. *J. Assoc. Comput. Mach.*, 37(2):200–212, 1990. `doi:10.1145/77600.77614`.
11. Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, 1993. `doi:10.1007/BF02189314`.
12. Bernard Chazelle. Cuttings. In Dinesh P. Mehta and Sartaj Sahni, editors, *Handbook of Data Structures and Applications*. Chapman and Hall/CRC, 2018. Second edition. `doi:10.1201/9781315119335`.

13 Bernard Chazelle and Burton Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom.*, 5(5):237–247, 1996. `doi:10.1016/0925-7721(95)00002-X`.

14 Bernard Chazelle and Emo Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4(5):467–489, 1989. `doi:10.1007/BF02187743`.

15 Kenneth L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2(2):195–222, 1987. `doi:10.1007/BF02187879`.

16 Zeev Dvir. On the size of Kakeya sets in finite fields. *J. Amer. Math. Soc.*, 22(4):1093–1097, 2009. `doi:10.1090/S0894-0347-08-00607-3`.

17 Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors. *Handbook of discrete and computational geometry*. Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL, 2018. Third edition. `doi:10.1201/9781315119601`.

18 Larry Guth and Nets Hawk Katz. On the Erdős distinct distances problem in the plane. *Ann. of Math. (2)*, 181(1):155–190, 2015. `doi:10.4007/annals.2015.181.1.2`.

19 D Haussler and E Welzl. Epsilon-nets and simplex range queries. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG '86, page 61–71, New York, NY, USA, 1986. Association for Computing Machinery. `doi:10.1145/10515.10522`.

20 Jiří Matoušek. Cutting hyperplane arrangements. *Discrete Comput. Geom.*, 6(5):385–406, 1991. `doi:10.1007/BF02574697`.

21 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993. `doi:10.1007/BF02573972`.

22 Jiří Matoušek and Zuzana Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54(1):22–41, 2015. `doi:10.1007/s00454-015-9701-2`.

23 Emo Welzl. Partition trees for triangle counting and other range searching problems. In *Proceedings of the Fourth Annual Symposium on Computational Geometry (Urbana, IL, 1988)*, pages 23–33. ACM, New York, 1988. `doi:10.1145/73393.73397`.

24 Dan E. Willard. Polygon retrieval. *SIAM J. Comput.*, 11(1):149–165, 1982. `doi:10.1137/0211012`.

25 A C Yao and F F Yao. A general approach to d-dimensional geometric queries. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 163–168, New York, NY, USA, 1985. Association for Computing Machinery. `doi:10.1145/22145.22163`.