On the Importance of Idempotence

Sunil Arya[*]
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
arya@cs.ust.hk

Theocharis Malamatos
Max-Planck-Institut für Informatik
Im Stadtwald, D-66123
Saarbrücken, Germany
tmalamat@mpi-inf.mpg.de

David M. Mount[†]
Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742
mount@cs.umd.edu

ABSTRACT

Range searching is among the most fundamental problems in computational geometry. An n-element point set in $\mathbb{R}^d$ is given along with an assignment of weights to these points from some commutative semigroup. Subject to a fixed space of possible range shapes, the problem is to preprocess the points so that the total semigroup sum of the points lying within a given query range η can be determined quickly. In the approximate version of the problem we assume that η is bounded, and we are given an approximation parameter $\varepsilon > 0$. We are to determine the semigroup sum of all the points contained within η and may additionally include any of the points lying within distance $\varepsilon \cdot diam(\eta)$ of η's boundary.

In this paper we contrast the complexity of range searching based on semigroup properties. A semigroup $(S, +)$ is *idempotent* if $x + x = x$ for all $x \in S$, and it is *integral* if for all $k \geq 2$, the k-fold sum $x + \cdots + x$ is not equal to x. For example, $(\mathbb{R}, \min)$ and $(\{0, 1\}, \vee)$ are both idempotent, and $(\mathbb{N}, +)$ is integral. To date, all upper and lower bounds hold irrespective of the semigroup. We show that semigroup properties do indeed make a difference for both exact and approximate range searching, and in the case of approximate range searching the differences are dramatic.

First, we consider exact halfspace range searching. The assumption that the semigroup is integral allows us to improve the best lower bounds in the semigroup arithmetic model. For example, assuming $O(n)$ storage in the plane and ignoring polylog factors, we provide an $\Omega^*(n^{2/5})$ lower bound for integral semigroups, improving upon the best lower bound of $\Omega^*(n^{1/3})$, thus closing the gap with the $O(n^{1/2})$ upper bound.

We also consider approximate range searching for Euclidean ball ranges. We present lower bounds and nearly matching upper bounds for idempotent semigroups. We also present lower bounds for range searching for integral semigroups, which nearly match existing upper bounds. These bounds show that the advantages afforded by idempotency can result in major improvements. In particular, assuming roughly linear space, the exponent in the ε-dependencies is smaller by a factor of nearly 1/2. All our results are presented in terms of space-time tradeoffs, and our lower and upper bounds match closely throughout the entire spectrum.

To our knowledge, our results provide the first proof that semigroup properties affect the computational complexity of range searching in the semigroup arithmetic model. These are the first lower bound results for any approximate geometric retrieval problems. The existence of nearly matching upper bounds, throughout the range of space-time tradeoffs, suggests that we are close to resolving the computational complexity of both idempotent and integral approximate spherical range searching in the semigroup arithmetic model.

Categories and Subject Descriptors: F.2 [Theory of Computation]: ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY

General Terms: Algorithms

Keywords: Range searching, approximation algorithms, idempotence.

1. INTRODUCTION

Answering range queries is a problem of fundamental importance in spatial information retrieval and computational geometry. The objective is to store a set of n points P in $\mathbb{R}^d$, each associated with a weight, so that it is possible to count (or more generally to compute some function of the weights of) the points lying inside a given query range. Range searching is a well studied problem, and many search structures have been proposed and analyzed [1, 15]. At an abstract level the problem has been one of the success stories of theoretical computational geometry, where years of study by numerous researchers have resulted in nearly matching asymptotic upper and lower bounds for many formulations of the problem. There is a spectrum of space-time tradeoffs. The most relevant work to ours involves halfspace range counting queries, which Matoušek [14] has shown can be answered in $O(n/m^{1/d})$ time from a data structure of space $O(m)$. Nearly matching lower bounds were given by

[*]This author's work was supported in part by the Research Grants Council, Hong Kong, China (HKUST6184/04E).
[†]This author's work was supported in part by the National Science Foundation under grant CCR-0098151.

STOC'06, May21–23, 2006, Seattle, Washington, USA.

Chazelle [8] for the more general problem of simplex range searching, and these were later refined to halfspace range searching by Brönnimann, Chazelle and Pach [7]. Throughout, we refer to this paper as *BCP*.

Given the relatively high complexity of range searching, it is natural to consider the problem in the context of approximation. We are given an approximation parameter $\varepsilon > 0$ and assume that ranges are bounded. Let η denote a range, and let $diam(\eta)$ denote its diameter. All the points that lie in the range must be counted, and any of the points that lie within distance $\varepsilon \cdot diam(\eta)$ of the range's boundary may be counted as well. Arya and Mount [5] showed that in any fixed dimension d with $O(n \log n)$ preprocessing time and $O(n)$ space, ε-approximate range queries for any bounded convex range can be answered in time $O(\log n + 1/\varepsilon^{d-1})$ [5]. Later, Chazelle, Liu, and Magen [9] considered approximate halfspace range and Euclidean ball searching in the high dimensional setting. Ignoring polylogarithmic factors, they showed that is possible to answer queries in $O(d/\varepsilon^2)$ time with $O(dn^{O(1/\varepsilon^2)})$ space.

In fixed dimensional spaces a natural goal is to achieve query times that are polylogarithmic in n while using space that is roughly linear in n. If we were to focus exclusively on n and assume that ε is a fixed constant, then the results of [5] would seem to be the end of the story. However, the additive term that depends on ε grows rapidly, and in practice these ε-dependencies dominate the query time. Throughout, we treat both n and ε as asymptotic quantities, and assume that $n \gg \varepsilon^{-1}$.

We are concerned with the following very broad question: *What is the computational complexity of approximate range searching in spaces of constant dimension?* This line of thought raises a number of questions. What are the best ε-dependencies that can be achieved? How do various aspects of the problem formulation affect these dependencies? What sorts of models, tools, and structures need to be developed to provide meaningful lower and upper bounds? As mentioned above, in range searching we are computing some function of the weights of the points lying within a range. Such a function is commonly assumed to arise from a commutative *faithful semigroup* over the domain of weights. We consider how semigroup properties affect the complexity of approximate (and exact) range searching.

A key semigroup property is idempotence. A semigroup is said to be *idempotent* if $x + x = x$ for all semigroup elements x. For example, $(\mathbb{R}, \min)$ and $(\{0,1\}, \vee)$ are both idempotent. The first is useful for reporting the smallest weight of any point in the range. The latter is useful for range emptiness queries, which is closely related to nearest neighbor queries. In contrast, if for all nonzero semigroup elements x and all natural numbers $k \geq 2$ the k-fold sum $x + \cdots + x$ is not equal to x, the semigroup is said to be *integral* [11]. For example, $(\mathbb{N}, +)$ is integral. It is useful for traditional counting queries.

To see the relevance of idempotence, consider how range searching usually works. At preprocessing time the algorithm implicitly computes the semigroup sum of a number of suitably chosen subsets of P, called *generators*.[1] To answer a query η, the algorithm determines an (ideally small) set of generators whose union covers $P \cap \eta$, and then returns their total sum. If the semigroup is idempotent, these generators may overlap, but for integral semigroups they must be disjoint. Because of the constraint of disjointness, one would expect that range searching over integral semigroups should be harder than for idempotent semigroups. It is remarkable, however, that for virtually all formulations of range searching, idempotence seems to be of no advantage. In their survey Agarwal and Erickson state, "Although in principle, storage schemes can exploit special properties of the semigroup, in practice, they never do. All known upper and lower bounds in the semigroup arithmetic model hold for all faithful semigroups." [1].

Our main result is that semigroup properties, idempotence in particular, do indeed make a difference in the complexity of both exact and approximate range searching. We show that for exact halfspace range searching, the assumption that the semigroup is integral allows us to prove a stronger lower bound in the semigroup arithmetic model than the one proved in BCP. For example, assuming $O(n)$ storage in the plane and ignoring polylog factors, we provide an $\Omega^*(n^{2/5})$ lower bound for integral semigroups, improving upon the BCP lower bound of $\Omega^*(n^{1/3})$, which holds for arbitrary semigroups, thus closing the gap with the $O(n^{1/2})$ upper bound [14]. Our proof requires the assumption of *convex generators*, which states that the convex hull of each generator subset contains no other points of P (see Section 3). We conjecture that our bounds hold even without this assumption.

Given the lengthy history of range searching, it is surprising that this fact has escaped notice until now. This may be because for higher dimensions the lower bounds for exact integral range searching are only marginally better than the BCP bounds for arbitrary semigroups. We show, however, that the story is dramatically different for approximate range searching. We present lower bounds for approximate range searching for both types of semigroups. We also present nearly matching upper bounds for idempotent semigroups (and upper bounds for integral semigroups were given in [4]). These bounds show that in the idempotent case, given $O(n/\varepsilon)$ space, the exponent in the ε-dependencies is smaller by a factor of nearly $1/2$.

We consider space-time tradeoffs for this problem. Rather than expressing our space and time tradeoffs in the conventional manner (query time as a function of space and data size) we adopt a notation that more clearly illustrates the incremental benefits of increased space. Recall that n denotes the data size, and let m denote the space of the data structure. Let $\rho = m/n$ denote the *expansion ratio* of the data structure size over data size. Clearly $m \geq n$, and so $\rho \geq 1$. We express query times as a fraction whose numerator gives the running time assuming the smallest amount of space supported by the data structure (which is typically $O(n)$), and the denominator gives the *tradeoff rate*, that is, the rate with which query time decreases as a function of a multiplicative increase in space. For example, for exact halfspace range queries, the conventionally expressed query time of $n/m^{1/d}$ would instead be expressed as $n^{1-1/d} \big/ \rho^{1/d}$. In this form it is readily seen from the numerator that the problem can be solved in $O(n^{1-1/d})$ time given linear space, and that by doubling space, the query time is decreased by

[1] Our use of the term *generator* is nonstandard. It is more commonly used to refer to a linear form involving subsets. We use it to refer to the subsets themselves, but since we only charge one unit of storage per generator, the computational model is identical.

		Idempotent		Integral	
Exact (Halfspaces)	Lower Bound	$n^{1-\frac{2}{d}+O\left(\frac{1}{d^2}\right)}$	[7]	$n^{1-\frac{1}{d}-O\left(\frac{1}{d^2}\right)}$	(new)
	Upper Bound	$n^{1-\frac{1}{d}}$	[14]	$n^{1-\frac{1}{d}}$	[14]
Approximate (Balls)	Lower Bound	$\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-O(1)}$	(new)	$\left(\frac{1}{\varepsilon}\right)^{d-O(1)}$	(new)
	Upper Bound	$\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-O(1)}$	(new)	$\left(\frac{1}{\varepsilon}\right)^{d-O(1)}$	[4]

Table 1: Query times (ignoring logarithmic factors) for n points and $O(n)$ space, except for the upper bound for approximate idempotent case, which requires $O(n/\varepsilon)$ space.

a factor of $2^{1/d}$. For one of our results (the upper bound for approximate idempotent queries) the minimum allowable space is $O(n/\epsilon)$. To indicate this we specify that ρ is at least $\Omega(1/\varepsilon)$.

Here is a summary of our results. (Also see Table 1 for a somewhat simpler presentation.) We use the notation O^* and Ω^* to indicate the omission of polylogarithmic factors. Our lower bound results are for worst-case query time in the semigroup arithmetic model assuming a faithful semigroup.

- We present a lower bound for exact halfspace range searching over integral semigroups. Assuming convex generators, we show that the query time is at least $\Omega^*\left(n^{1-\frac{1}{d}-O\left(\frac{1}{d^2}\right)} \Big/ \rho^{\frac{1}{d}+\frac{1}{d^2}}\right)$. By contrast, the BCP lower bound is $\Omega^*\left(n^{1-\frac{2}{d}+O\left(\frac{1}{d^2}\right)} \Big/ \rho^{\frac{1}{d}}\right)$. See Theorem 1 for details.
- We present a lower bound for answering ε-approximate range queries for Euclidean balls over arbitrary semigroups (and hence for idempotent semigroups). We show that the query time is $\Omega^*\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-1} \Big/ \rho^{\frac{1}{2}-\frac{1}{2(d+1)}}\right)$. See Theorem 2(i) for details.
- We present a lower bound for answering ε-approximate range queries for Euclidean balls over integral semigroups. Assuming convex generators, we show that the query time is at least $\Omega^*\left(\left(\frac{1}{\varepsilon}\right)^{d-5} \Big/ \rho^{1-\frac{4}{d}}\right)$. See Theorem 2(ii) for details.
- We present a data structure for answering ε-approximate range queries for Euclidean balls over idempotent semigroups. We show that if ρ is at least $\Omega(1/\varepsilon)$, queries can be answered in $O^*\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-\frac{1}{2d}} \Big/ \rho^{\frac{1}{2}-\frac{1}{2d}}\right)$ time. See Theorem 3 for details.

In [4] we showed that Euclidean ball range queries over integral semigroups can be answered with a space-time tradeoff of $O^*\left((1/\varepsilon)^{d-1} \big/ \rho^{1-\frac{1}{d}}\right)$. Our results show that the restriction to idempotent semigroups can be of great benefit for approximate range searching. Although both data structures are based on the general concept of an AVD (or *approximate Voronoi diagram*) [2,3,13], the methods used for constructing generators in this paper are considerably different. Table 1 summarizes our results on exact halfspace range searching and approximate range searching for Euclidean balls in dimension d, assuming n points and $O(n)$ space, and ignoring logarithmic factors.

These are the first results we know of that exhibit the impact of semigroup properties on the complexity of range searching. These are the first lower bound results for *any* approximate geometric retrieval problems. The existence of nearly matching upper bounds, throughout the range of space-time tradeoffs, suggests that we are close to resolving the computational complexity of both idempotent and integral approximate spherical range searching in the semigroup arithmetic model.

2. PRELIMINARIES

Before presenting our results we make some preliminary remarks about the computational model. Throughout we assume that the dimension d is a fixed constant and treat n and ε as asymptotic quantities. Unless otherwise stated, we will use the term "constant" to refer to any fixed quantity that may depend on d but not on n or ε. Throughout, let $\mathbb{U}^d$ denote the unit hypercube in $\mathbb{R}^d$, and given a body C in $\mathbb{R}^d$, let $\mu(C)$ denote its Lebesgue measure.

Let $(S,+)$ be a commutative semigroup. We will assume that each element in S can be stored in unit space, and that for any two elements $x, y \in S$, their semigroup sum $x+y$ can be computed in constant time. Let P be a set of n points in $\mathbb{R}^d$ and let $w : P \to S$ be a function that assigns a semigroup value in S to each point in P. For any subset G of P, we define its weight $w(G) = \sum_{p\in G} w(p)$, where the summation is taken over the semigroup. Let $\mathcal{Q}$ denote the set of query ranges being considered. Recall that in the exact range searching problem, we are required to preprocess $\mathcal{Q}$ so that for any query range $\eta \in \mathcal{Q}$, we can efficiently compute $w(P \cap \eta)$. In the approximate range searching problem, instead of computing $w(P \cap \eta)$, it suffices to compute $w(X)$, where X is any subset of P satisfying $P \cap \eta \subseteq X \subseteq P \cap \eta^+$. Here η^+ denotes the expanded range, which consists of all points that lie within distance $\varepsilon \cdot diam(\eta)$ of η.

Our lower bound proofs are in the *semigroup arithmetic model* [7, 8, 12, 16]. Due to space limitations, we omit its technical definition. Lower bound proofs in this model assume that the semigroup is *faithful*, meaning that any two identically equal linear forms have the same set of variables. For example, $(\mathbb{N},+)$, $(\mathbb{R},\min)$, and $(\{0,1\},\vee)$ are faithful, but $(\{0,1\}, + \bmod 2)$ is not. Let $\mathcal{G}$ be any set of m generators. For any range $\eta \in \mathcal{Q}$, define $A_\eta \subseteq \mathcal{G}$ to be the smallest set such that $\bigcup_{G\in A_\eta} G = P \cap \eta$. Define $T(P,\mathcal{Q},m)$ to be minimum, over all sets $\mathcal{G}$ consisting of m generators, of $\max_{\eta\in\mathcal{Q}} |A_\eta|$. Then in the semigroup arithmetic model, the following holds: Given m units of storage, for any commutative faithful semigroup, the worst-case query time for range searching is at least equal to $T(P,\mathcal{Q},m)$ [8]. The only modification necessary for approximate range searching is to define $A_\eta \subseteq \mathcal{G}$ to be the smallest set such that the union of the corresponding generators, $\bigcup_{G\in A_\eta} G$, contains all the points of P lying within η but none of the points lying outside of η^+.

3. EXACT HALFSPACE RANGE SEARCH

In BCP a lower bound of $\Omega\left((n/\log n)^{1-\frac{d-1}{d(d+1)}}/m^{1/d}\right)$ is derived for exact halfspace range searching for arbitrary semigroups in the semigroup arithmetic model. In this section we present an improved lower bound for the case of integral semigroups. We need an additional assumption. Let P denote the point set and $\mathcal{G}$ be the set of generators. We say that $\mathcal{G}$ satisfies the *convex generator assumption* if for all $G \in \mathcal{G}$, we have $G = P \cap conv(G)$, where $conv(G)$ denotes the convex hull of G. This assumption does not seem to be very restrictive. Although it does not apply to the most efficient data structure for halfspace range searching [14], after minor modifications it does apply to the quasi-optimal data structure of Chazelle, Sharir, and Welzl [10].

Our proof is based on a framework similar to that devised in BCP. In their proof the generators used to answer a query are allowed to overlap. A key contribution of our paper is to show how to enhance their proof, so it yields a better lower bound for the integral case, by exploiting the fact that the generators used to answer a query must be disjoint. Our new idea leads to only a marginal improvement for exact halfspace range searching, but in Section 4 we shall see that this leads to a much better lower bound for approximate range searching for Euclidean balls.

We assume some prior familiarity with the BCP proof [7]. Here we restrict ourselves to a brief description of it, which will allow us to illustrate the intuition underlying our improvements for the integral case. The lower bound example consists of a *scattered* set P of n data points in the unit hypercube $\mathbb{U}^d$. Roughly speaking, scattered here means that the number of data points in any suitably large convex body $C \subseteq \mathbb{U}^d$ is $\Theta(n \cdot \mu(C))$. (A random set of n points sampled uniformly and independently in $\mathbb{U}^d$ is scattered with high probability.) Let $\mathcal{G}$ denote a set of generators that allow for any halfspace query to be answered in time t in the semigroup arithmetic model. A suitable set $\mathcal{H}_0^+$ of query halfspaces is defined along with an appropriate differential element dH for halfspaces. In BCP it is observed that the complexity of halfspace range searching stems from the difficulty of covering points that lie close to the boundary of the halfspace. To formalize this idea, for each halfspace $H \in \mathcal{H}_0^+$, define a *region of interest* $R_H \subseteq H$ to be a slab of thickness $(ct \log n)/n$ (for a suitable constant c) adjacent to the bounding hyperplane of H. A lower bound on the query time t is obtained by computing lower and upper bounds on the quantity $\Phi = \int_{\mathcal{H}_0^+} |P \cap R_H| \, dH$.

For any halfspace $H \in \mathcal{H}_0^+$, the BCP proof shows that $|P \cap R_H|$ is at least $3at \log n$, where a is a suitable constant. In order to compute an upper bound on Φ, let us say that a generator G is *absolutely fat* with respect to an $H \in \mathcal{H}_0^+$ if $|G \cap R_H| > 2a \log n$ and $G \subseteq H$. The BCP proof makes crucial use of the following simple fact.

LEMMA 1. *For any $H \in \mathcal{H}_0^+$, a constant fraction of the points of P lying within R_H are covered by generators in $\mathcal{G}$ that are absolutely fat with respect to H.*

Using the lemma, it follows that, up to constant factors, $\Phi \le \sum_{G \in \mathcal{G}} \int_{\Delta_G} |G \cap R_H| \, dH$, where Δ_G is the set of halfspaces in $\mathcal{H}_0^+$ for which G is absolutely fat. Intuitively, $\int_{\Delta_G} |G \cap R_H| \, dH$ serves as a measure of the *usefulness* of generator G. Note that a generator is viewed as useful for only those query halfspaces H for which it is absolutely fat, and its contribution in this case is the number of data points it covers in R_H. Naturally, for a fixed number of generators, an upper bound on the usefulness of any generator would imply a lower bound on the query time t. It is not easy to bound the usefulness of a generator, and in BCP some beautiful techniques are devised to compute it. We can present the intuition behind our improvement without these details. In the integral case, since the generators used to answer a query are required to be disjoint, intuitively it seems that this must reduce the usefulness of a generator. To formalize this intuition, we devise a different notion of a generator's usefulness, more suited to the integral case. Let us say that a generator G is *relatively fat* with respect to an $H \in \mathcal{H}_0^+$ if $|G \cap R_H| > (at \log n/2n)|G|$ and $G \subseteq H$. We can now easily establish the following lemma in the integral case.

LEMMA 2. *For any $H \in \mathcal{H}_0^+$, a constant fraction of the points of P lying within R_H are covered by generators in $\mathcal{G}$ that are both absolutely and relatively fat with respect to H.*

The proof of the lemma is omitted due to space limitations. The lemma implies that

$$\Phi \;\le\; \sum_{G \in \mathcal{G}} \int_{\Delta'_G} |G \cap R_H| \, dH,$$

where Δ'_G is the set of halfspaces in $\mathcal{H}_0^+$ for which G is both absolutely and relatively fat. In the integral case, $\int_{\Delta'_G} |G \cap R_H| \, dH$ measures the *usefulness* of generator G. In contrast to the idempotent case, a generator is now viewed as useful for only those query halfspaces H for which it is both absolutely and relatively fat. Intuitively, to see that this is a considerably stronger condition, let us first consider the case of a large generator that covers a constant fraction of the data points in $\mathbb{U}^d$. To be absolutely fat with respect to a halfspace H, it would only need to cover $2a \log n$ points in R_H, but to be relatively fat, it would need to cover roughly t times as many points in R_H (neglecting constant factors). It follows that, by our new criterion, a large generator has smaller usefulness than it would possess by virtue of the BCP criterion. On the other hand, if a generator is small, then in any case its usefulness would be small as fewer halfspaces in $\mathcal{H}_0^+$ would find such a generator close to their boundary. Considerations such as these suggest that our new criterion should lead to a smaller estimate for the maximum possible usefulness of a generator. Indeed, subject to the convex generator assumption, it is not too hard to adapt the proof in BCP and confirm the accuracy of this basic intuition. We omit these details due to space limitations. (But see Section 4 for calculations of a similar flavor in the context of approximate range searching.) We conclude with the main result of this section.

THEOREM 1. *Let $d > 1$ be a fixed dimension. Consider a range space consisting of all halfspaces and a weight function over any integral semigroup. Then for all sufficiently large n, the worst-case query time in the semigroup arithmetic model, for exact range searching among n points, using $m \ge n$ units of storage, is at least*

$$\Omega\left(\frac{n}{m^{\frac{d+1}{d^2+1}} \log n}\right) \;=\; \Omega\left(n^{1-\frac{1}{d}-O\left(\frac{1}{d^2}\right)} \Big/ \rho^{\frac{1}{d}+\frac{1}{d^2}}\right),$$

where $\rho = m/n$ is the expansion ratio.

4. EUCLIDEAN BALLS: LOWER BOUNDS

Here are the main results of this section. The first result applies to arbitrary (faithful) semigroups, and so applies to idempotent semigroups, and the second applies to integral semigroups. This theorem implies that the upper bounds on query times presented in Section 5 for idempotent semigroups and in [4] for arbitrary semigroups are both nearly optimal.

THEOREM 2. *Let $d > 1$ be a fixed dimension. Consider ε-approximate range searching for n points over the range space of Euclidean balls for a weight function over a faithful semigroup. Then for all sufficiently small ε and sufficiently large n, and for $m \geq n$ units of storage (that is, $\rho = m/n$),*

(*i*) *The worst-case query time in the semigroup arithmetic model is at least*

$$\Omega\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-1}\left(\frac{n}{m\log\frac{1}{\varepsilon}}\right)^{\frac{1}{2}-\frac{1}{2(d+1)}}\right) = \Omega^*\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-1} \bigg/ \rho^{\frac{1}{2}-\frac{1}{2(d+1)}}\right).$$

(*ii*) *Further, if the semigroup is integral it is at least*

$$\Omega\left(\left(\frac{1}{\varepsilon}\right)^{d-5}\left(\frac{n}{m}\right)^{1-\frac{4}{d}}\right) = \Omega\left(\left(\frac{1}{\varepsilon}\right)^{d-5} \bigg/ \rho^{1-\frac{4}{d}}\right).$$

We begin with the proof of Theorem 2(i). Our proof is structurally similar to the proof in BCP for establishing the hardness of exact halfspace range searching. However, a considerable number of nontrivial modifications are needed in order to adapt their framework to our context. We begin with a high-level description of these modifications.

As with exact halfspace range searching, the complexity of approximate spherical range searching arises from the difficulty of covering points close to the boundary of the query range. Recall from Section 3 that this idea is formalized in BCP for the exact case by defining a thin slab R_H close to the boundary of each query halfspace H, and showing that if any generator covers a lot of points in this slab, then it cannot be useful in this manner for many queries. More precisely, the *usefulness* of a generator G is measured by the quantity $\int_{\Delta_G} |G \cap R_H|\, dH$, where Δ_G is the set of query halfspaces for which G is *absolutely fat*. In BCP a lower bound on the query time is derived by estimating the maximum possible usefulness of any generator.

The notion of a slab for halfspace range queries naturally corresponds in our context to the concept of a *generalized annulus*, that is, the difference of two concentric Euclidean balls. With both exact and approximate range searching, reducing the thickness of the slab or annulus (assuming a certain given radius) leads in general to better lower bounds. However, if the slab or annulus is too thin, then it would contain too few points, and such a region can be easily covered using generators consisting of single points. With exact range searching, this is a key factor limiting how thin to make the slabs. In approximate range searching, however, there is an additional very important factor to consider. Let r denote the radius of the range. Using an annulus of width smaller than $o(\varepsilon r)$ is counterproductive because the algorithm is at liberty to include points that lie outside the range within distance $2\varepsilon r$ of the boundary and so effectively increasing the annulus width to $\Theta(\varepsilon r)$. This suggests that we should set the annulus width to εr. But note that we can now adjust the parameter r. To obtain the best lower bound, we want to make r as small as we can, subject to the above constraint on the minimum number of points the annulus must contain. Our approach, which is essentially equivalent, is to use query balls whose radii are allowed to range over some constant interval, create many separate instances of the same point distribution, called *replicants*, and apply the lower bound argument separately to each one.

The second important issue involves the technique used to compute an upper bound on the usefulness of any generator. In the exact case, this is computed with the help of a sophisticated decomposition of the convex hull of a generator into convex bodies called *Macbeath regions*. (See Lemma 4.) Unfortunately, this machinery seems to break down utterly on replacing hyperplane slabs by our "annulus slabs." Not only are annulus slabs not convex, the intersection of the convex hull of a generator and an annulus may even consist of many connected components.

An obvious approach for overcoming this difficulty is to use the standard transformation of lifting to the paraboloid. This reduces problems involving Euclidean balls in $\mathbb{R}^d$ to problems involving halfspaces in $\mathbb{R}^{d+1}$. The problem here is that we lose the uniformity of the point distribution (a central element to the BCP proof) since the lifted points are constrained to lie on the paraboloid. Thus, we cannot directly relate the volume of a convex body in $\mathbb{R}^{d+1}$ to the number of points that are expected to lie within the body. To handle this problem we establish a type of isoperimetric inequality that relates the surface area of a convex body in $\mathbb{R}^d$ to the volume in $\mathbb{R}^{d+1}$ of the convex hull of the lifted body on the paraboloid. With its aid, we can determine the relationship between the volume of the convex bodies in $\mathbb{R}^{d+1}$ that arise in our proof and the number of lifted points inside it. (See Property 3 of the scattered set properties given below). Furthermore, we show that the Macbeath-region machinery developed in the BCP proof can now be applied in our context.

Next we present some of the tools from [7, 8] that will be needed in our analysis and show how these are modified and generalized to the context of range queries with Euclidean balls. Our proof will make use of the standard *lifting map*. Given a point $p \in \mathbb{R}^d$, let $p^{\uparrow}$ denote its projection onto the paraboloid $x_{d+1} = \sum_{i=1}^{d} x_i^2$ in $\mathbb{R}^{d+1}$. Given a ball B in $\mathbb{R}^d$ of radius r centered at some point q, let $h(B)$ denote the d-dimensional hyperplane in $\mathbb{R}^{d+1}$ passing through the lift of all the points on the boundary of B. It is well known that p lies inside/on/outside B if and only if $p^{\uparrow}$ lies respectively below/on/above $h(B)$ [6]. Let $h^-(B)$ denote the lower halfspace containing lifted points that lie within B, and define $h^+(B)$ analogously for points outside of B.

Given a set $X \subset \mathbb{R}^d$, let $X^{\uparrow}$ denote the image of X under the lifting transformation. Thus, if X is some connected region of $\mathbb{R}^d$, the lifted set $X^{\uparrow}$ will form a connected surface patch on the paraboloid. After lifting, we will be interested in the $(d+1)$-dimensional volume of the convex hull of the lifted set, which will generally lie both on and above the surface of the paraboloid. Let $\mu_C(X^{\uparrow})$ denote this volume of the convex hull of the lifted set, or more formally, $\mu_C(X^{\uparrow}) = \mu(conv(X^{\uparrow}))$.

Let P be a set of n points in the unit hypercube $\mathbb{U}^d =$

$[0,1]^d$. We say that P is *scattered* if the following three properties hold for some constant $a > 1$ depending only on dimension. The first property provides a lower bound on the number of points in a convex body in terms of its volume. The second provides a lower bound on the number of points in a generalized annulus in terms of the volume of the annulus. The third provides an upper bound on the number of points lying in the difference of a convex body and a ball. Such a difference need not be convex or even connected, but as mentioned above we can relate it to the volume of an appropriate convex set in the lifted space, denoted $L(K, B)$, which consists of taking the convex hull of $K^\uparrow$ and intersecting it with $h^+(B)$.

Property 1: Let K be any convex body contained in $\mathbb{U}^d$, and let $k = |P \cap K|$. If $k \geq \log n$, then $k \geq (n/a)\mu(K)$.

Property 2: Let B_1 and B_2 be any two concentric balls contained in $\mathbb{U}^d$ such that the radii of both balls exceeds some fixed constant, and let $k = |P \cap (B_1 \setminus B_2)|$. If $\mu(B_1 \setminus B_2) \geq (a \log n)/n$, then $k \geq (n/a)\mu(B_1 \setminus B_2)$.

Property 3: Let K be any convex body and B be any ball, both contained in $\mathbb{U}^d$. Define $L(K, B) = conv(K^\uparrow) \cap h^+(B)$, and let $k = |P \cap (K \setminus B)|$. If $k \geq \log n$, then $k \leq an(n/\log n)^{2/d}\mu(L(K, B))$.

The following lemma asserts the existence of sets satisfying these properties. The proof of the lemma is omitted due to space limitations.

Lemma 3. *A random set of n points chosen uniformly and independently in a unit hypercube U is scattered with probability $1 - o(1)$.*

Following BCP, our lower bound proof requires us to decompose generators in order to analyze their impact on covering points close to the boundary of the query range. The following tool, which involves important convex bodies called *Macbeath regions*, will help us in that task. We state this and the following lemma in dimension $d + 1$ since we will apply them to lifted sets.

Lemma 4. *(Brönnimann, Chazelle and Pach [7]) Given a compact convex body $K \subset \mathbb{R}^{d+1}$ of unit volume and $0 < \beta < 1$, there exists a collection of $O((1/\beta)^{1-2/(d+2)})$ convex bodies, $K_1, K_2, \ldots \subseteq K$, satisfying the following condition: For any halfspace τ with $\mu(K \cap \tau) \geq \beta$, there exists $K_i \subseteq K \cap \tau$ such that $\mu(K_i) = \Omega(\mu(K \cap \tau))$.*

The third tool we will need is an isoperimetric inequality proved by Chazelle [8] that bounds the probability that a "random" slab encloses a given convex body. For any real $\alpha > 0$ and any hyperplane H in $\mathbb{R}^{d+1}$, let $S^\alpha(H)$ denote the slab consisting of points in $\mathbb{R}^{d+1}$ whose distance from H is at most α. Let O denote the origin of a coordinate frame of reference that we associate with $\mathbb{R}^{d+1}$. For any point $q \in \mathbb{R}^{d+1} \setminus \{O\}$, let H_q denote the hyperplane that passes through q and is orthogonal to the segment $\overline{Oq}$. Define the measure λ of any set $\mathcal{X}$ of hyperplanes as follows:

$$\lambda(\mathcal{X}) = \int_{\mathcal{X}} dH \ = \ \int_{H_q \in \mathcal{X}} \frac{dx_1 \wedge \ldots \wedge dx_{d+1}}{\|q\|^d},$$

where $q = (x_1, \ldots, x_{d+1})$. The choice of this measure is governed by the fact that it is invariant under rigid motions [8].

Lemma 5. *(Chazelle [8]) Given any compact convex body $K \subseteq \mathbb{U}^d$, we have $\mu(K) \cdot \int_{S^\alpha(H) \supseteq K} dH = O(\alpha^{d+2})$.*

We are now ready to prove Theorem 2(i). Let ε denote the approximation error. Throughout we assume that ε is a sufficiently small real number between 0 and 1. Let $\mathcal{Q}$ denote the set of all Euclidean balls. Let n be the number of points in the data set. (We assume that n is sufficiently large). Let $m \geq n$ denote the number of generators, and let $t = t(n, m)$ denote the worst-case query time in the arithmetic model over all the ranges in $\mathcal{Q}$.

We construct a set P of n data points for which we will argue that the query time must be sufficiently large for some range in $\mathcal{Q}$. As mentioned above in the overview of the proof, we will assume that P is composed of a collection of identical subsets, called *replicants*. Towards this end, let

$$n' \ = \ \frac{t}{\varepsilon} \log \frac{t}{\varepsilon}.$$

For simplicity we will assume that n is a multiple of n'. Consider any collection $\mathcal{U}$ of n/n' interior-disjoint unit hypercubes. Our set P consists of a scattered set of n' points placed in each of these hypercubes.

Let $\mathcal{G}$ denote any set of m generators for P. For each hypercube in $\mathcal{U}$, consider the subset of generators that contains no point outside this hypercube. Let U' denote the hypercube that has the smallest such subset of generators. Let $\mathcal{G}'$ denote this subset of generators. Clearly $|\mathcal{G}'| \leq m'$, where $m' = mn'/n$. Let us restrict ourselves henceforth to the subset of n' points $P' = P \cap U'$. Without loss of generality, we may take U' to be the unit hypercube $\mathbb{U}^d$. The remainder of the proof consists of placing a lower bound on the number of generators needed to cover some ball of $\mathcal{Q}$ that is contained within $\mathbb{U}^d$, as a function of n', m', and ε. This will provide the desired lower bound. To complete the proof, this bound will then be cast in terms of our original parameters n and m.

Let $\mathcal{Q}'$ denote the set of balls b of radius between $1/4$ and $1/2$ such that the corresponding ε-expanded ball, b^+, lies entirely within $\mathbb{U}^d$. Let $\mathcal{H}$ denote the set of all hyperplanes in $d{+}1$ dimensions obtained by applying the map h to each of the balls in $\mathcal{Q}'$. To simplify the notation, for any hyperplane $H \in \mathcal{H}$, we let O_H denote the corresponding ball $h^{-1}(H)$.

For any $H \in \mathcal{H}$, let $A_H \subseteq \mathcal{G}$ denote the smallest set of generators that provides a valid answer for the query O_H, that is,

$$P \cap O_H \ \subseteq \ \bigcup_{G \in A_H} G \ \subseteq \ P \cap O_H^+. \qquad (1)$$

Clearly $t \geq |A_H|$. Since the hypercubes of $\mathcal{U}$ have disjoint interiors and $O_H^+ \subseteq \mathbb{U}^d$, it follows that the above inequality holds if P is replaced with P' and A_H is restricted to a subset of $\mathcal{G}'$.

So far, we have limited consideration to a subset of generators $\mathcal{G}'$ of size at most m' that lie entirely within the unit hypercube $\mathbb{U}^d$, and to the subset of points P' of size n' that lie within $\mathbb{U}^d$ and a subset of ball ranges $\mathcal{Q}'$ whose ε-expansions lie within $\mathbb{U}^d$. In the first part of the proof, we will consider the problem only within this limited context.

Recall that we have a hyperplane $H \in \mathcal{H}$ and its associated ball O_H. We define a region of interest for H as follows. Let c_1 be a positive constant, whose value will be set later, and let $\alpha = c_1\varepsilon$. Let $O_H(\alpha)$ denote the ball concentric with

O_H whose radius is smaller than that of O_H by α. The *region of interest* for H, denoted R_H, is defined to be

$$R_H \;=\; O_H \setminus O_H(\alpha).$$

Later in the proof, we will also make use of the following outer region, which will be convenient to define now. Let

$$R_H^{\geq} \;=\; \overline{O_H(\alpha)} \qquad (2)$$

Clearly $R_H \subset R_H^{\geq}$.

As observed in BCP, the complexity of (exact) halfspace range searching arises from the difficulty of covering points inside the range that lie close to its boundary. This factor is also responsible for the complexity of approximate range searching. In order to make this more precise, we introduce a quantity, which corresponds roughly to the number of points lying within the region of interest for an average query. Consider the quantity

$$\Phi \;=\; \int_{\mathcal{H}} |P' \cap R_H|\, dH, \qquad (3)$$

where dH is the differential element defined earlier for hyperplanes. We will compute lower and upper bounds on Φ, which together will provide the desired lower bound on the worst-case query time t. The intuition behind our proof is that if a generator covers a large number of points in R_H, then it cannot be useful in this manner for many queries.

For all sufficiently small ε, one can easily verify that for any $H \in \mathcal{H}$, the volume of the region of interest satisfies

$$\mu(R_H) \;\geq\; c_2\alpha,$$

where c_2 is some suitable constant. Now, by setting $c_1 = 8a/c_2$, we have

$$\mu(R_H) \;\geq\; 8a\varepsilon.$$

It is easy to verify that this exceeds $(a \log n')/n'$, and so, by Property 2 of scattered point sets, it follows that the number of points of P' in R_H is at least $8t\log(t/\varepsilon)$. Clearly the measure of $\mathcal{H}$ is at least some constant, and so we have the following lower bound on Φ.

$$\Phi \;=\; \Omega\left(t \log \frac{t}{\varepsilon}\right). \qquad (4)$$

Next we compute an upper bound on Φ. Towards this end, it is helpful to concentrate on those generators that are most efficient in covering the region of interest R_H. We say that a generator $G \in \mathcal{G}'$ is *absolutely fat* with respect to a hyperplane $H \in \mathcal{H}$ if $|G \cap R_H| > 4\log(t/\varepsilon)$ and $G \subseteq O_H^+$. We have the following lemma. The proof of the lemma is similar to that of Lemma 1, and is omitted.

LEMMA 6. *For any hyperplane $H \in \mathcal{H}$, a constant fraction of the points of P' lying within R_H are covered by generators in $\mathcal{G}'$ that are absolutely fat with respect to H.*

Recalling that $R_H \subset R_H^{\geq}$, the above lemma implies that a constant fraction of the points of P' in R_H are covered by generators $G \in \mathcal{G}'$ that satisfy $|G \cap R_H^{\geq}| > 4\log(t/\varepsilon)$ and $G \subseteq O_H^+$. Thus, it follows from the definition of Φ in Eq. (3) that, up to constant factors,

$$\Phi \;\leq\; \sum_{G \in \mathcal{G}'} \int_{\Delta_G} |G \cap R_H|\, dH, \qquad (5)$$

where

$$\Delta_G \;=\; \left\{ H \in \mathcal{H} \;:\; |G \cap R_H^{\geq}| > 4\log\frac{t}{\varepsilon} \;\text{ and }\; G \subseteq O_H^+ \right\}.$$

We will refer to the quantity $\int_{\Delta_G} |G \cap R_H|\, dH$ as the *usefulness* of generator G, denoted $u(G)$. Our goal is to compute an upper bound on the maximum possible usefulness of any generator $G \in \mathcal{G}'$, which in turn will help us derive a lower bound on t.

Since $R_H \subset R_H^{\geq}$ we have

$$u(G) \;\leq\; \int_{\Delta_G} |G \cap R_H^{\geq}|\, dH.$$

Let H be a hyperplane in Δ_G. Because generator G contributes a sufficiently large number of points within $R_H^{\geq}$, we may apply Property 3 of scattered points to bound the volume of the convex set $conv(conv(G)^{\uparrow}) \cap h^+(O_H(\alpha))$ as follows. Since $G \subseteq P' \cap conv(G)$, we have

$$|G \cap R_H^{\geq}| \;\leq\; |P' \cap conv(G) \cap R_H^{\geq}|. \qquad (6)$$

By definition of Δ_G, $|G \cap R_H^{\geq}| > 4\log\frac{t}{\varepsilon}$, and so $|P' \cap conv(G) \cap R_H^{\geq}| > 4\log\frac{t}{\varepsilon} \geq \log n'$ (recall that we are dealing with a single replicant). By applying Property 3 (where the convex body is $conv(G)$ and the ball is $O_H(\alpha)$) we have

$$\begin{aligned} |P' \cap conv(G) \cap R_H^{\geq}| \;&\leq\; a\,n' \left(\frac{n'}{\log n'}\right)^{\frac{2}{d}} \mu(L'(G,H)) \\ &\leq\; a\,n' \left(\frac{t}{\varepsilon}\right)^{\frac{2}{d}} \mu(L'(G,H)), \qquad (7) \end{aligned}$$

where $L'(G,H) = L(conv(G), O_H(\alpha)) = conv(conv(G)^{\uparrow}) \cap h^+(O_H(\alpha))$. Thus, by combining Eqs. (6) and (7) we obtain, up to constant factors,

$$u(G) \;\leq\; n' \left(\frac{t}{\varepsilon}\right)^{\frac{2}{d}} \int_{\Delta_G} \mu(L'(G,H))\, dH. \qquad (8)$$

In order to bound this integral, first observe that for all G, in the integration domain, by Eq. (7) we have

$$\mu(L'(G,H)) \;\geq\; \frac{1}{an'} \left(\frac{\varepsilon}{t}\right)^{\frac{2}{d}} |P' \cap conv(G) \cap R_H^{\geq}|. \qquad (9)$$

Recalling that $|P' \cap conv(G) \cap R_H^{\geq}| \geq 4\log(t/\varepsilon)$, and substituting the definition of n' for a single replicant, it follows that $\mu(L'(G,H))$ is at least $(4/a)(\varepsilon/t)^{1+2/d}$. Since $conv(G) \subseteq \mathbb{U}^d$ and, by definition of the lifting transformation, the $(d+1)$-th coordinate after lifting of any point in $\mathbb{U}^d$ is at most d, it follows that $\mu(conv(conv(G)^{\uparrow}))$ is at most d. Therefore,

$$\frac{\mu(L'(G,H))}{\mu(conv(conv(G)^{\uparrow}))} \;\geq\; \frac{4}{ad}\left(\frac{\varepsilon}{t}\right)^{1+\frac{2}{d}}.$$

Now, by setting β equal to the right hand side of the above equation, we may apply Lemma 4 to obtain a collection of $O\left((1/\beta)^{1-2/(d+2)}\right)$ convex bodies $K_1, K_2, \ldots$, each contained within $conv(conv(G)^{\uparrow})$, such that for some K_j

(i) $K_j \subseteq L'(G,H)$, and
(ii) $\mu(K_j) = \Omega(\mu(L'(G,H)))$.

For some constant c, we assert that condition (i) implies that K_j is contained within the slab $S^{c\varepsilon}(H)$. To see this, recall that $G \subseteq O_H^+$. It follows that $conv(conv(G)^{\uparrow})$ lies entirely below the hyperplane $h(O_H^+)$. Therefore $L'(G,H) = conv(conv(G)^{\uparrow}) \cap h^+(O_H(\alpha))$ lies between the hyperplanes $h(O_H(\alpha))$ and $h(O_H^+)$. Since $\alpha = \Theta(\varepsilon)$, by basic properties of the lifting transformation, it follows that $L'(G,H) \subseteq S^{c\varepsilon}(H)$, for a suitable constant c. The desired assertion now follows from condition (i).

Thus, up to constant factors,

$$\int_{\Delta_G} \mu(L'(G,H))\, dH \;\le\; \sum_j \mu(K_j) \int_{\Delta'_{K_j}} dH,$$

where

$$\Delta'_{K_j} \;=\; \{H \in \mathcal{H} \;:\; S^{c\varepsilon}(H) \supseteq K_j\}.$$

By the isoperimetric inequality given in Lemma 5, we obtain

$$\mu(K_j) \int_{\Delta'_{K_j}} dH \;=\; O(\varepsilon^{d+2}).$$

Thus,

$$\int_{\Delta_G} \mu(L'(G,H))\, dH \;=\; \sum_j O(\varepsilon^{d+2}).$$

From our bound on the number of convex bodies K_j and by the definition of β we have

$$\begin{aligned}\int_{\Delta_G} \mu(L'(G,H))\, dH &= O\left(\left(\frac{1}{\beta}\right)^{1-\frac{2}{d+2}} \varepsilon^{d+2}\right)\\ &= O\left(\left(\frac{t}{\varepsilon}\right)^{\left(1+\frac{2}{d}\right)\left(1-\frac{2}{d+2}\right)} \varepsilon^{d+2}\right)\\ &= O\left(t\varepsilon^{d+1}\right).\end{aligned}$$

Substituting this into Eq. (8), we obtain

$$u(G) \;=\; O\left(n'\, t^{1+\frac{2}{d}}\, \varepsilon^{d+1-\frac{2}{d}}\right).$$

Recalling that $|\mathcal{G}'| \le m'$, we obtain the following upper bound on Φ.

$$\Phi \;=\; O\left(n'\, m'\, t^{1+\frac{2}{d}}\, \varepsilon^{d+1-\frac{2}{d}}\right).$$

Now, by combining this with the lower bound on Φ given in Eq. (4) we obtain the following

$$n'\, m'\, t^{1+\frac{2}{d}}\, \varepsilon^{d+1-\frac{2}{d}} \;=\; \Omega\left(t \log \frac{t}{\varepsilon}\right). \tag{10}$$

We are now able to incorporate the above single-replicant results into the overall analysis. Recall that $m' = mn'/n$ and $n' = (t/\varepsilon)\log(t/\varepsilon)$. Note that we may assume that $t < 1/\varepsilon^{d/2-1}$, since otherwise Theorem 2(i) holds trivially. It follows that $\log t = O(\log(1/\varepsilon))$. Substituting these values of m' and n' into Eq. (10) and then simplifying yields Theorem 2(i).

Next we consider the proof of Theorem 2(ii) (for integral semigroups). The general structure of the proof is the same as that of part (i) for arbitrary semigroups. However, when selecting the subset of generators to be used in the analysis, we apply a similar criterion as the one introduced in Section 3. In particular, we require that the generators used in the analysis are both absolutely and relatively fat. As in Section 3 it can be shown that, by this new criterion, a large generator has smaller usefulness than it would possess by virtue of the criterion used in the proof of part (i). This leads to a significantly better lower bound.

As in part (i), we begin by decomposing the point set P into replicants, and then analyzing each replicant separately. The decomposition and the analysis of the single replicant until Lemma 6 is carried out in exactly the same way as in part (i). We next show that, by exploiting the fact that the generators used to answer a query must be disjoint, we can significantly strengthen Lemma 6. Towards this end, let us say that a generator $G \in \mathcal{G}'$ is *relatively fat* with respect to a hyperplane $H \in \mathcal{H}$ if $|G \cap R_H| > \varepsilon|G|$ and $G \subseteq O_H^+$. We have the following lemma analogous to Lemma 2. The proof of the lemma is straightforward and is omitted due to lack of space.

LEMMA 7. *For any hyperplane $H \in \mathcal{H}$, a constant fraction of the points of P' lying within R_H are covered by generators in $\mathcal{G}'$ that are both absolutely and relatively fat with respect to H.*

Arguing exactly as in part (i), but using Lemma 7 instead of Lemma 6, it is straightforward to show that Eqs. (5)–(9) all hold, with Δ_G redefined as:

$$\left\{H \in \mathcal{H} \;:\; |G \cap R_H^{\ge}| > \max\left(4 \log \frac{t}{\varepsilon}, \varepsilon|G|\right) \text{ and } G \subseteq O_H^+\right\}. \tag{11}$$

This new definition of Δ_G will enable us to obtain a much better upper bound on $u(G)$ and hence on Φ. In particular, it will allow us to decompose $conv(conv(G)^{\uparrow})$ using Lemma 4 into a much fewer number of Macbeath regions which, in turn, leads to a better upper bound on $u(G)$. We now present the details of this calculation.

By Eqs. (6) and (11), $|P' \cap conv(G) \cap R_H^{\ge}| \ge |G \cap R_H^{\ge}| > \varepsilon|G|$. Substituting this bound in Eq. (9), we obtain

$$\mu(L'(G,H)) \;\ge\; \frac{1}{an'} \left(\frac{\varepsilon}{t}\right)^{\frac{2}{d}} \varepsilon|G|. \tag{12}$$

Next we compute an upper bound on $\mu(conv(conv(G)^{\uparrow}))$. First, observe that $\mu(conv(conv(G)^{\uparrow})) \le d\mu(conv(G))$. This follows from the fact that $conv(conv(G)^{\uparrow})$ is contained inside a prism whose base is $conv(G)$ and $(d+1)$-th coordinate spans the interval $[0,d]$. (Because $conv(G) \subseteq \mathbb{U}^d$, and by definition of the lifting transformation, the $(d+1)$-th coordinate after lifting of any point in $\mathbb{U}^d$ is at most d.) In order to bound $\mu(conv(G))$, recall that $|P' \cap conv(G) \cap R_H^{\ge}| \ge \log n'$ (which we showed just after Eq. (6)). Therefore $|P' \cap conv(G)| \ge |P' \cap conv(G) \cap R_H^{\ge}| \ge \log n'$, so we can apply Property 1 of scattered point sets to obtain $\mu(conv(G)) \le (a/n')|P' \cap conv(G)|$. By the assumption of *convex generators*, $G = P' \cap conv(G)$, and so $\mu(conv(G)) \le (a/n')|G|$. Therefore

$$\mu(conv(conv(G)^{\uparrow})) \;\le\; \frac{ad}{n'}|G|. \tag{13}$$

Combining Eqs. (12) and (13), we obtain

$$\frac{\mu(L'(G,H))}{\mu(conv(conv(G)^{\uparrow}))} \;\ge\; \frac{1}{a^2 d}\left(\frac{\varepsilon}{t}\right)^{\frac{2}{d}} \varepsilon. \tag{14}$$

Now, by setting β equal to the right hand side of the above equation, we may apply Lemma 4 to obtain a collection of $O\left((1/\beta)^{1-2/(d+2)}\right)$ convex bodies $K_1, K_2, \ldots \subseteq conv(conv(G)^{\uparrow})$ such that some K_j satisfies the conditions

(i) and (ii) that were described earlier in part (i). The crucial observation is that the value of β in the integral case is larger than the value of β in part (i) by a factor of t (ignoring constant factors). The rest of the analysis is identical to part (i) and is therefore omitted. We finally obtain

$$t \geq \left(\frac{1}{\varepsilon}\right)^{\frac{d-1-\frac{2}{d}}{1+\frac{2}{d}+\frac{2}{d+2}}} \left(\frac{n}{m\log(1/\varepsilon)}\right)^{\frac{1}{1+\frac{2}{d}+\frac{2}{d+2}}}, \qquad (15)$$

which can be easily simplified to yield Theorem 2(ii).

5. EUCLIDEAN BALLS: UPPER BOUND (IDEMPOTENT CASE)

In this section we present our data structures for answering ε-approximate range queries over an idempotent semigroup. (Analogous results for arbitrary, and hence integral, semigroups were already presented in [4].) Our main result is summarized below.

THEOREM 3. *Let P be a set of n points in $\mathbb{R}^d$. Let $0 < \varepsilon \leq 1/2$ and $2 \leq \gamma \leq 1/\varepsilon$ be two real parameters. Then we can construct a data structure of $O(n\gamma^d/\varepsilon)$ space that answers ε-approximate spherical range queries over any idempotent semigroup in time $O(\log n + (1/(\varepsilon\gamma))^{(d-1)/2}\log(1/\varepsilon))$. It takes $O(n(\gamma/\varepsilon)^{(d+1)/2}\log(n/\varepsilon))$ time to construct the data structure.*

By defining the space expansion factor to be $\rho = \gamma^d/\varepsilon$ we have the following equivalent space-time tradeoff, ignoring logarithmic factors.

COROLLARY 3.1. *Given the setup of the Theorem 3, ε-approximate spherical range queries can be answered in time $O^*\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d}{2}-\frac{1}{2d}} \Big/ \rho^{\frac{1}{2}-\frac{1}{2d}}\right)$, where ρ is at least $\Omega(1/\varepsilon)$.*

The algorithm employs an adaptation of the AVD (or approximate Voronoi diagram) structure, as was presented in [4] for the case of general semigroups. This is a quadtree-like structure, in which space is subdivided hierarchically into cells until its leaf cells satisfy certain separation properties with respect to the surrounding points. Each node of the tree (both internal nodes and leaves) is responsible for handling queries whose center lies within the corresponding cell and whose radius is proportional to the size of the cell (depending on the degree of separation). Each node is associated with a set of generators. Because the semigroup is idempotent, the generators used to answer a query are allowed to overlap. This allows us to choose generators in the most economical way, as subsets of points lying within a judiciously chosen discrete set of Euclidean balls. We show that if the nodes of the decomposition satisfy certain separation properties, then it is possible to cover any range approximately with a small set of generators. Space-time tradeoffs are handled by altering the degree of separation. As the separation increases (controlled by a parameter γ), the number of nodes increases, but the number of generators needed to answer each query decreases.

We refer the reader to [4] for standard definitions of the box-decomposition of space into cells. The *size* of a cell is defined to be the size of its outer box. Throughout, for a cell u, we will use s_u to denote its size and b_u to denote the ball of radius $s_u d/2$ whose center coincides with the center of u's outer box. (Note that $u \subseteq b_u$.) Finally, for any ball b and any positive real γ, we use γb to denote the ball with the same center as b and whose radius is γ times the radius of b.

Let P be a set of n points in $\mathbb{R}^d$. Let $0 < \varepsilon \leq 1/2$ and $16 \leq \gamma \leq 1/\varepsilon$ be two real parameters. The parameter γ is used to control the space/time tradeoff. Without loss of generality, we may assume that the set of points P has been scaled and translated to lie within a ball of radius $\varepsilon/4$ placed at the center of the unit hypercube $\mathbb{U}^d$. We will assume that the query ball is contained within $\mathbb{U}^d$ since the other case can be handled easily.

The starting point for our construction is a data structure similar to that used previously for answering approximate nearest neighbor [2,3] and approximate spherical range counting queries [4]. In Lemma 8, we abstract the main features of this data structure. The structure is parameterized by two real numbers $0 < f \leq 1$ and $\gamma \geq 16$, and for our purposes it can be viewed as a collection of three types of cells (which may overlap), as described below. Cells of *type 2* and *type 3* satisfy certain separation properties (depending on f and γ) with respect to the points of P lying outside the cell, while cells of *type 1* do not. Letting u denote the cell under consideration, u has the following properties depending on its type.

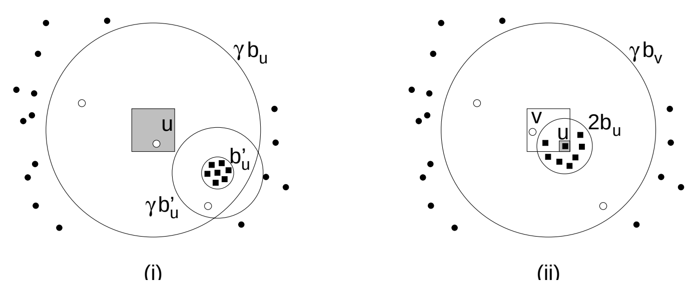

Figure 1: The separation properties for (i) type-2 cells satisfying property (a) and (ii) type-3 cells. Pollutants are indicated with hollow circles and points of the inner clusters are shown as black squares.

Type-1: u is a quadtree box.

Type-2: u is either a quadtree box or the set theoretic difference of two quadtree boxes. There exists a ball b'_u such that $|P \cap (\gamma b_u \setminus b'_u)| = O(1/f)$, and either (a) the ball $\gamma b'_u$ does not overlap u or (b) letting r_1 and r_2 denote the radii of balls b_u and b'_u, respectively, and ℓ denote the minimum distance of separation between b_u and b'_u, $\ell \geq \max(r_1, r_2)$ and $\ell/\sqrt{r_1 r_2} \geq \sqrt{\gamma}$.

Type-3: u is a quadtree box. There is an associated box v such that $u \subseteq v$ and $|P \cap (\gamma b_v \setminus 2b_u)| = O(1/f)$.

Intuitively, for a type-2 cell u, if there are many points close to it, then all but $O(1/f)$ points among them are sufficiently well clustered relative to their distance to the cell. For a type-3 cell u, if there are many points within a certain defined neighborhood, then all but $O(1/f)$ points among them are close to u. (See Fig. 1.) The point subsets of size $O(1/f)$ that do not satisfy the separation properties are called *pollutants*. These points are handled by simple brute force during the query processing. The following lemma can now be proved easily using ideas from Lemma 4 in [3] and Section 3 in [4]. Details are omitted due to space limitations.

Lemma 8. (Separation Properties) *Let P be a set of n points in $\mathbb{R}^d$, and let $\gamma \geq 16$ and $0 < f \leq 1$ be two real parameters. In $O(n\gamma^d \log(n\gamma))$ time, it is possible to construct a data structure with $O(nf\gamma^d)$ cells of type-1, type-2, and type-3, respectively, such that the following holds. For any query ball η, in $O(\log(n\gamma))$ time, we can find a cell u such that the center of η lies in u and u satisfies one of the following:*

(i) *u is of type-1 and $(\gamma/2)b_u \subseteq \eta^+ \subseteq (3\gamma)b_u$.*
(ii) *u is of type-2 and $\eta^+ \subseteq \gamma b_u$.*
(iii) *u is of type-3 and $\gamma b_u \subseteq \eta^+ \subseteq \gamma b_v$. Here v denotes the quadtree box associated with u.*

We set $f = (\varepsilon\gamma)^{(d-1)/2}$ and construct the data structure described in Lemma 8. During preprocessing, for each cell in this data structure, we compute the weight of certain generators and store them with the cell. To answer a query η, we first apply the above data structure to find a cell u satisfying one of the three properties (i)–(iii). We then use the precomputed generators stored with u to answer the query. Due to lack of space, we limit our discussion here to the case of type-1 cells (i.e., property (i) holds). This case helps to illustrate our main ideas. (The geometric technique used in the case of type-2 cells is similar and the case of type-3 cells is relatively easy.)

Recall that a cell of type 1 does not generally satisfy any separation property. If such a cell had to handle arbitrarily small or big query balls in its vicinity, it would be much too hard to answer queries efficiently. But by property (i) note that a type-1 cell only needs to handle query balls centered in it whose radius is proportional to γ times its own size. We will show that such queries can be handled efficiently using only a small set of generators. Our choice of generators relies crucially on the following geometric lemma.

Lemma 9. *Let $16 \leq \gamma \leq 1/\varepsilon$. Let u be any cell. It is possible to find a set $\mathcal{B}$ of $O((1/(\varepsilon\gamma))^{(d-1)/2} \cdot (1/\varepsilon))$ balls and store them in $O(|\mathcal{B}|)$ space, such that for any query ball η that is centered in u and satisfies $(\gamma/2)b_u \subseteq \eta^+ \subseteq (3\gamma)b_u$, the following property holds: In $O((1/(\varepsilon\gamma))^{(d-1)/2} \cdot \log(1/\varepsilon))$ time, it is possible to find a subset $\mathcal{B}_\eta \subseteq \mathcal{B}$ of $O((1/(\varepsilon\gamma))^{(d-1)/2})$ balls such that their union, $\bigcup_{b\in\mathcal{B}_\eta} b$, covers η and is contained within η^+.*

The proof of this lemma has been omitted due to space limitations. Let us see how it is applied. During preprocessing, for each type-1 cell u we obtain the set $\mathcal{B}$ of balls described in the lemma. For each ball $b \in \mathcal{B}$, we compute $w(P \cap b)$ and associate it with b. The space used for storing this information is on the order of the number of balls in $\mathcal{B}$, which is $O((1/(\varepsilon\gamma))^{(d-1)/2} \cdot (1/\varepsilon))$. Using these generators, we can determine the answer for any query ball η satisfying property (i) by first finding the set $\mathcal{B}_\eta \subseteq \mathcal{B}$ of balls described in the statement of the lemma, and then outputting $\sum_{b\in\mathcal{B}_\eta} w(P \cap b)$. The correctness of this method follows from the fact that $\eta \subseteq \bigcup_{b\in\mathcal{B}_\eta} b \subseteq \eta^+$. Summing the time it takes to find $\mathcal{B}_\eta$ with the $O(\log(n\gamma))$ time it takes to find cell u (by Lemma 8), the total query time is $O(\log n + (1/(\varepsilon\gamma))^{(d-1)/2} \log(1/\varepsilon))$.

In the full version, we show that the space and query time bounds proved above for type-1 cells also hold for type-2 and type-3 cells. Since the space used per cell is $O((1/(\varepsilon\gamma))^{(d-1)/2} \cdot (1/\varepsilon))$, and the total number of cells is $O(nf\gamma^d)$ with $f = (\varepsilon\gamma)^{(d-1)/2}$, the total space used by the data structure is $O(n\gamma^d/\varepsilon)$. Borrowing ideas from [4], we can construct a slightly modified version of this data structure in time $O(n(\gamma/\varepsilon)^{(d+1)/2} \log(n/\varepsilon))$. Details are omitted due to lack of space.

6. REFERENCES

[1] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary Mathematics*, pages 1–56. American Mathematical Society, Providence, RI, 1999.

[2] S. Arya and T. Malamatos. Linear-size approximate Voronoi diagrams. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 147–155, 2002.

[3] S. Arya, T. Malamatos, and D. M. Mount. Space-efficient approximate Voronoi diagrams. In *Proc. 34th Annual ACM Sympos. Theory Comput.*, pages 721–730, 2002.

[4] S. Arya, T. Malamatos, and D. M. Mount. Space-time tradeoffs for approximate spherical range counting. In *Proc. 16th ACM-SIAM Sympos. Discrete Algorithms*, pages 535–544, 2005.

[5] S. Arya and D. M. Mount. Approximate range searching. *Computational Geometry: Theory and Applications*, 17:135–152, 2000.

[6] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, UK, 1998. Translated by H. Brönnimann.

[7] H. Brönnimann, B. Chazelle, and J. Pach. How hard is halfspace range searching. *Discrete Comput. Geom.*, 10:143–155, 1993.

[8] B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2:637–666, 1989.

[9] B. Chazelle, D. Liu, and A. Magen. Approximate range searching in higher dimension. In *Proc. 16th Canad. Conf. Comput. Geom.*, 2004.

[10] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8:407–429, 1992.

[11] J. Erickson. Space-time tradeoffs for emptiness queries. *SIAM J. Comput.*, 29:1968–1996, 2000.

[12] M. L. Fredman. Lower bounds on the complexity of some optimal data structures. *SIAM J. Comput.*, 10:1–10, 1981.

[13] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 94–103, 2001.

[14] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993.

[15] J. Matoušek. Geometric range searching. *ACM Comput. Surv.*, 26:421–461, 1994.

[16] A. C. Yao. On the complexity of maintaining partial sums. *SIAM J. Comput.*, 14:277–288, 1985.