

Resolving SINR Queries in a Dynamic Setting

Boris Aronov¹

Department of Computer Science and Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA
boris.aronov@nyu.edu

Gali Bar-On²

Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
galibar@post.bgu.ac.il

Matthew J. Katz³

Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
matya@cs.bgu.ac.il

Abstract

We consider a set of transmitters broadcasting simultaneously on the same frequency under the SINR model. Transmission power may vary from one transmitter to another, and a signal's strength decreases (*path loss* or *path attenuation*) by some constant power α of the distance traveled. Roughly, a receiver at a given location can hear a specific transmitter only if the transmitter's signal is stronger than the signal of all other transmitters, combined. An SINR query is to determine whether a receiver at a given location can hear any transmitter, and if yes, which one.

An approximate answer to an SINR query is such that one gets a definite YES or definite NO, when the ratio between the strongest signal and all other signals combined is well above or well below the reception threshold, while the answer in the intermediate range is allowed to be either YES or NO.

We describe several compact data structures that support approximate SINR queries in the plane in a dynamic context, i.e., where both queries and updates (insertion or deletion of a transmitter) can be performed efficiently.

2012 ACM Subject Classification Theory of computation → Computational geometry, Networks → Network algorithms

Keywords and phrases Wireless networks, SINR, dynamic insertion and deletion, interference cancellation, range searching

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.145

Acknowledgements We wish to thank Pankaj K. Agarwal, Timothy Chan, Sariel Har-Peled, and Wolfgang Mulzer for discussions, hints, and outright help with some aspects of this paper. Work on this paper was initiated at the Fifth Workshop on Geometry and Graphs, Bellairs Research Institute, Barbados, 2017.

¹ B. Aronov was supported by NSF grants CCF-11-17336, CCF-12-18791, and CCF-15-40656, and by grant 2014/170 from the US-Israel Binational Science Foundation.

² G. Bar-On was supported by the Lynn and William Frankel Center for Computer Science.

³ M. Katz was supported by grant 1884/16 from the Israel Science Foundation and by grant 2014/170 from the US-Israel Binational Science Foundation.



© Boris Aronov, Gali Bar-On, and Matthew J. Katz;
licensed under Creative Commons License CC-BY

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamani, Dániel Marx, and Donald Sannella;
Article No. 145; pp. 145:1–145:13



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Table 1** Approximate SINR queries in a static setting – previous results. φ is an upper bound on the fatness parameters of the reception regions and $\varphi' \geq \varphi^2$.

	Preprocessing	Space	Query
Uniform Power [5]	$O(n^2/\varepsilon)$ ($O(n\varepsilon^{-2.5} \log^4 n \log \log n)$ [3])	$O(n/\varepsilon)$	$O(\log n)$
Non-Uniform Power [10]	$O(\frac{\varphi'}{\varepsilon^2} n^2)$	$O(\frac{\varphi'}{\varepsilon^2} n)$	$O(\frac{\varphi}{\varepsilon} \log n)$

1 Introduction

The *Signal to Interference plus Noise Ratio (SINR) model* attempts to predict whether a wireless transmission is received successfully, in a setting consisting of multiple simultaneous transmitters in the presence of background noise. Let $S = \{s_1, \dots, s_n\}$ be a set of n transmitters (points in the plane), and let p_i denote the transmission power of s_i , for $i = 1, \dots, n$. Let q be a receiver (a point in the plane). According to the SINR model, q receives s_i if and only if

$$\text{sinr}(q, s_i) = \frac{\frac{p_i}{|qs_i|^\alpha}}{\sum_{j \neq i} \frac{p_j}{|qs_j|^\alpha} + N} \geq \beta,$$

where $\alpha \geq 1$ and $\beta > 1$ are constants, N is a constant representing the background noise, and $|ab|$ is the Euclidean distance between points a and b .

Observe that, since $\beta > 1$, q may receive at most one transmitter – the one “closest” to it, namely, the one for which the value $\frac{p_i}{|qs_i|^\alpha}$ is maximum, or, equivalently, $\frac{1}{p_i^{1/\alpha}} |qs_i|$ is minimum. Thus, one can partition the plane into n not necessarily connected reception regions R_i , one per transmitter in S , plus an additional region R_\emptyset consisting of all points where none of the transmitters is received. This partition is called the *SINR diagram* of S [5].

In their seminal paper, Avin et al. [5] studied properties of SINR diagrams, focusing on the *uniform power* version where $p_1 = p_2 = \dots = p_n$. Their main result is that in this version the reception regions R_i are convex and fat. In the *non-uniform power* version, on the other hand, the reception regions are not necessarily connected, and their connected components are not necessarily convex or fat. In fact, they may contain holes [10].

An *SINR query* is: Given a receiver q , find the sole transmitter s that may be received by q and determine whether it is indeed received by q , i.e., whether or not $\text{sinr}(q, s) \geq \beta$. A natural question is how quickly can one answer an SINR query, following a preprocessing stage in which data structures of total size nearly linear in n are constructed. However, it seems unlikely that the answer is significantly sub-linear (as the degree of the polynomials describing region boundaries is high), so the research has focused on preprocessing to facilitate efficient *approximate* SINR queries.

The approach of such research has been to construct a data structure which approximates the underlying SINR diagram, and use it for answering approximate SINR queries, by performing point-location queries in this structure. That is, given a query point q , first find the sole candidate s_i that may be received at q (by searching in the appropriate Voronoi diagram), and then perform a point-location query to approximately determine whether q is in R_i or not. Two different notions of approximation have been used. In the first [5], it is guaranteed that the uncertain answer is only given infrequently, namely, the area of the uncertain region associated with R_i is at most $\varepsilon \cdot \text{area}(R_i)$, for a prespecified parameter $\varepsilon > 0$. In the second [10], it is guaranteed that for every point in the uncertain region the

SIN ratio is within an ε -neighborhood of β . See Table 1 for a summary of previous results; see also [11] for related work. In addition, Aronov and Katz [3] obtained several results for *batched* approximate SINR queries, using the latter notion of approximation; for example, one can perform n simultaneous approximate queries in a network with n transmitters at polylogarithmic amortized cost per query.

Given $\varepsilon > 0$,⁴ an *approximate SINR query* is: Given a receiver q , find the sole transmitter s that may be received by q and return a value $\tilde{\text{sinr}}(q, s)$, such that $(1 - \varepsilon)\text{sinr}(q, s) \leq \tilde{\text{sinr}}(q, s) \leq (1 + \varepsilon)\text{sinr}(q, s)$. Thus, unless $(1 - \varepsilon)\beta \leq \tilde{\text{sinr}}(q, s) < (1 + \varepsilon)\beta$, the value $\tilde{\text{sinr}}(q, s)$ enables us to determine definitely whether or not s is received by q .

In this paper, we devise efficient algorithms for handling approximate SINR queries in a dynamic setting. That is, given S , α, β , and N , as above, and $\varepsilon > 0$, we describe algorithms for answering approximate SINR queries after some initial preprocessing, in a setting where transmitters may be added to or deleted from S . We analyze our algorithms by the usual measures, namely, preprocessing time, data structure size, query time, and update time.

To the best of our knowledge, these are the first data structures to support dynamic approximate SINR queries. In contrast with previous work on approximate SINR queries, our algorithms do not compute an approximation of the underlying SINR diagram. We distinguish between two main variants of the problem – the uniform power version and the non-uniform power one. The preprocessing time in both cases is $O(n \text{ polylog } n)$, while the update and query time is $O(\text{polylog } n)$ for the uniform version, and $O(\sqrt{n} \text{ polylog } n)$ for the non-uniform version. Thus, our solution for the *dynamic* uniform version is comparable to the best known solutions for the *static* uniform version. For the non-uniform version, however, our solution is the first solution with bounds that depend only on n and ε and not on other parameters of the input, both in the static and dynamic settings.

In addition to the obvious motivation for devising algorithms for approximate SINR queries in a dynamic setting, we mention another important application of our results. *Successive Interference Cancellation (SIC)* is a technique that enables (in some circumstances) a receiver q to receive a specific transmitter t , even if t cannot be received at q in SINR sense. Informally, our results support SIC; if t 's signal is the k th strongest at q , then, through a sequence of $O(k)$ queries and updates, we can determine whether q can decode t 's signal from the combined signal; see the discussion in Section 4. In contrast, Avin et al. [4] construct a uniform-power static data structure of size $O(\varepsilon^{-1}n^{10})$ which enables one to determine in $O(\log n)$ time whether t can be received by q using SIC. Their result is not directly comparable to ours, however: They guarantee logarithmic query regardless of the number of transmitters that need to be canceled before t can be heard, and their approximation model is quite different from ours.

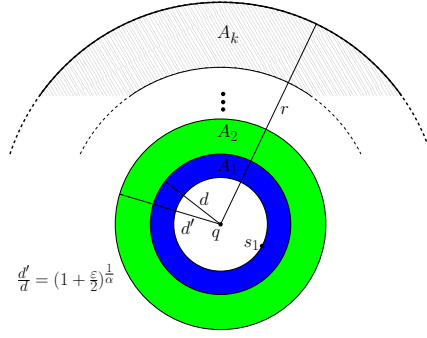
The missing proofs and some extensions can be found in the complete version of this paper [2].

2 Uniform power

Let q be a receiver and let s be the closest transmitter to q . Set $\text{intrf}(q) = \sum_{s' \in S \setminus \{s\}} \frac{1}{|qs'|^\alpha}$, then $\text{sinr}(q, s) = \frac{1}{|qs|^\alpha} \cdot \frac{1}{\text{intrf}(q)}$.⁵ When s is the transmitter closest to q , we will simply write

⁴ For simplicity of presentation, we will assume hereafter that $n > 1/\varepsilon$.

⁵ For clarity of presentation, we assume hereafter that there is no noise, i.e., $N = 0$. Our algorithms extend to the situation where noise is present in a straightforward manner.



■ **Figure 1** Partitioning $A_q(|q s_1|, r)$ into annuli.

$\text{sinr}(q)$ instead of $\text{sinr}(q, s)$. Fix $\varepsilon > 0$. We wish to compute a value $\widetilde{\text{sinr}}(q)$ such that $(1 - \varepsilon)\text{sinr}(q) \leq \widetilde{\text{sinr}}(q) \leq \text{sinr}(q)$. We show below how to compute a value $\widetilde{\text{intrf}}(q)$ such that $\text{intrf}(q) \leq \widetilde{\text{intrf}}(q) \leq (1 + \varepsilon)\text{intrf}(q)$ and then simply set $\widetilde{\text{sinr}}(q) = \frac{\frac{1}{|q s_1|^\alpha}}{\widetilde{\text{intrf}}(q)}$.

► **Claim 1.** *Under the above assumption, $(1 - \varepsilon)\text{sinr}(q) < \widetilde{\text{sinr}}(q) \leq \text{sinr}(q)$.*

We start with a slower but easier to describe solution and then refine it.

2.1 Annuli

Let $\varepsilon > 0$. Let q be a receiver and let $s \in S$ be the closest transmitter to q . Let s_1, \dots, s_{n-1} be the transmitters in $S \setminus \{s\}$, and assume without loss of generality that s_1 is the second closest transmitter to q , among all the transmitters in S . Recall that $\text{intrf}(q) = \sum_{i=1}^{n-1} \frac{1}{|q s_i|^\alpha}$ and that we wish to compute a value $\widetilde{\text{intrf}}(q)$ such that $\text{intrf}(q) \leq \widetilde{\text{intrf}}(q) \leq (1 + \varepsilon)\text{intrf}(q)$.

We will need the following simple observation.

► **Observation 2.** *$\text{intrf}(q)$ is the sum of $n - 1$ positive terms of which $\frac{1}{|q s_1|^\alpha}$ is the largest, so we have $\frac{1}{|q s_1|^\alpha} \leq \text{intrf}(q) \leq \frac{n-1}{|q s_1|^\alpha}$.*

2.1.1 Query algorithm

Let q be a query point. First, we find s and s_1 , the closest and the second closest transmitters to q , respectively. Next, we divide the transmitters in $S \setminus \{s\}$ into two subsets, S_c and S_f , where S_c consists of all transmitters that are ‘close’ to q and S_f consists of all transmitters that are ‘far’ from q . More precisely, set $r = (\frac{2n}{\varepsilon})^{1/\alpha} \cdot |q s_1|$, then S_c consists of all the transmitters in $S \setminus \{s\}$ whose distance from q is less than r , and S_f consists of all the remaining transmitters. We now approximate the contribution of each of these subsets to $\text{intrf}(q)$.

The contribution of a transmitter s_i in S_f to the sum $\text{intrf}(q)$ is

$$\frac{1}{|q s_i|^\alpha} \leq \frac{1}{r^\alpha} = \frac{\varepsilon}{2n|q s_1|^\alpha},$$

and the combined contribution of the transmitters in S_f is at most $|S_f| \cdot \frac{\varepsilon}{2n|q s_1|^\alpha} \leq \frac{\varepsilon}{2|q s_1|^\alpha}$.

We denote the annulus centered at q with inner radius r_1 and outer radius r_2 by $A_q(r_1, r_2)$. In order to approximate the overall contribution of the transmitters in S_c , we partition the annulus $A_q(|q s_1|, r)$ into k semi-open annuli, A_1, \dots, A_k , such that the ratio of the outer to the inner radius of A_j is $(1 + \frac{\varepsilon}{2})^{1/\alpha}$ (except for A_k whose ratio is at most $(1 + \frac{\varepsilon}{2})^{1/\alpha}$); see

Figure 1. By semi-open we mean that the inner circle of A_j is contained in A_j , but the outer circle is not. Now, for each A_j , we approximate the contribution of each transmitter $s_i \in S_c \cap A_j$ by $\frac{1}{d^\alpha}$, where d is the inner radius of A_j , that is, we approximate the contribution of s_i by moving it to the inner circle of A_j . We prove below (Corollary 4) that this yields a $(1 + \frac{\varepsilon}{2})$ -approximation of the overall contribution of the transmitters in S_c to $\text{intrf}(q)$.

► **Lemma 3.** *Let $s_i \in S_c$ and let $A = A_q(d, d')$ be the annulus to which s_i belongs. Then, by moving s_i to the inner circle of A , one obtains a $(1 + \frac{\varepsilon}{2})$ -approximation of the contribution of s_i to $\text{intrf}(q)$.*

Proof. Since $s_i \in A$, $d \leq s_i < d'$. Moreover, by construction, $d'/d \leq (1 + \frac{\varepsilon}{2})^{\frac{1}{\alpha}}$. So, the ratio of our approximation to the real contribution of s_i is

$$\frac{1/d^\alpha}{1/|qs_i|^\alpha} = \frac{|qs_i|^\alpha}{d^\alpha} < \left(\frac{d'}{d}\right)^\alpha \leq 1 + \frac{\varepsilon}{2} . \quad \blacktriangleleft$$

► **Corollary 4.** *By doing this for each transmitter in S_c , one obtains a $(1 + \frac{\varepsilon}{2})$ -approximation of the overall contribution of the transmitters in S_c to $\text{intrf}(q)$.*

It remains to show that $\widetilde{\text{intrf}}(q)$, which is the sum of the approximations for S_f and for S_c , satisfies the requirements, i.e., that $\text{intrf}(q) \leq \widetilde{\text{intrf}}(q) \leq (1 + \varepsilon)\text{intrf}(q)$. From the description above it is clear that $\widetilde{\text{intrf}}(q) \geq \text{intrf}(q)$, so we only need to show that $\widetilde{\text{intrf}}(q) \leq (1 + \varepsilon)\text{intrf}(q)$. Indeed,

$$\widetilde{\text{intrf}}(q) \leq \frac{\varepsilon}{2|qs_1|^\alpha} + (1 + \frac{\varepsilon}{2}) \sum_{s_i \in S_C} \frac{1}{|qs_i|^\alpha} \leq \frac{\varepsilon}{2} \text{intrf}(q) + (1 + \frac{\varepsilon}{2}) \text{intrf}(q) = (1 + \varepsilon) \text{intrf}(q) ,$$

where the second inequality is based on Observation 2.

2.1.2 Implementation

We first show that k , the number of annuli into which the annulus $A_q(|qs_1|, r)$ is partitioned, is small; for the proof, see the complete version [2].

► **Lemma 5.** $k = O(\frac{1}{\varepsilon} \log n)$.

In the preprocessing stage we compute the following data structures for the set of transmitters S .

Dynamic nearest neighbor: A data structure due to Kaplan *et al.* [12] can be used for dynamic 2D nearest-neighbor queries. A set of points in the plane can be maintained dynamically in a linear-size data structure, so as to support insertions, deletions, and nearest-neighbor queries. Each insertion takes $O(\log^3 n)$ amortized deterministic time, each deletion takes $O(\log^5 n)$ amortized deterministic time, and each query takes $O(\log^2 n)$ worst-case deterministic time, where n is the size of the set of points at the time the operation is performed; see [12] and also the randomized data structure of Chan [6] with slightly worse (expected) performance.

Dynamic disk range counting: We start with the construction of Matoušek [13]: In linear space and $O(n \log n)$ time one can preprocess a set of n points in \mathbb{R}^d to support semi-group halfspace range queries in $O(n^{1-1/d} \text{polylog } n)$ time. A point can be deleted in $O(\log n)$ amortized time and inserted in $O(\log^2 n)$ amortized time. Lifting circles to points in \mathbb{R}^3 in

the standard manner, we obtain a linear-space, $O(n \log n)$ time, $O(n^{2/3} \text{polylog } n)$ disk range counting query, $O(\log n)$ amortized delete, $O(\log^2 n)$ amortized insert data structure. We have not attempted to optimize this ingredient, as we replace this infrastructure with a more efficient one in the following section.

Given a query point q , we find s and s_1 (the closest and second closest transmitters) using the data structure for dynamic nearest neighbor; s_1 is found by deleting and reinserting s . Next, we compute the distance r and partition the annulus $A_q(|qs_1|, r)$ into k annuli, as described above. Now, we calculate $\widetilde{\text{intrf}}(q)$ as follows. We first compute the size of the set S_f by performing a disk counting query with the circle of radius r centered at q and subtracting the answer from $n - 1$; we initialize $\widetilde{\text{intrf}}(q)$ to $|S_f| \cdot \frac{\varepsilon}{2n|qs_1|^\alpha}$. Next, for each of the k annuli, we compute the number x of points of S lying in it, as the difference in the numbers of points in the two disks defined by its bounding circles, obtained by counting queries. We then increment $\widetilde{\text{intrf}}(q)$ by $\frac{x}{d^\alpha}$, where d is the radius of the inner circle of the current annulus.

An update is performed by updating the two underlying data structures.

We omit the detailed performance analysis of this version, as a better data structure is described next.

2.2 Polygonal rings

We now present a more efficient solution similar to the previous one, except that we replace the circular annuli by polygonal rings. Set $x = (1 + \frac{\varepsilon}{2})^{\frac{1}{\alpha}}$, and consider any three circles C_0, C_1, C_2 centered at q , such that $r_1/r_0 = r_2/r_1 = \sqrt{x}$, where r_i is the radius of C_i . Set $l = \lceil \frac{\pi}{\sqrt{2 - \frac{2}{\sqrt{x}}}} \rceil$, and let B_i be the regular l -gon inscribed in C_i , so that one its vertices lies on the upward vertical ray through q , for $i = 1, 2$. We now show that C_{i-1} is contained in B_i , for $i = 1, 2$, and therefore, the polygonal ring defined by B_1 and B_2 is contained in the annulus $A_q(r_0, r_2)$. The elementary proofs of the following claims can be found in the complete version [2].

► **Claim 6.** $l = O(1/\sqrt{\varepsilon})$.

► **Claim 7.** C_{i-1} is contained in B_i , for $i = 1, 2$.

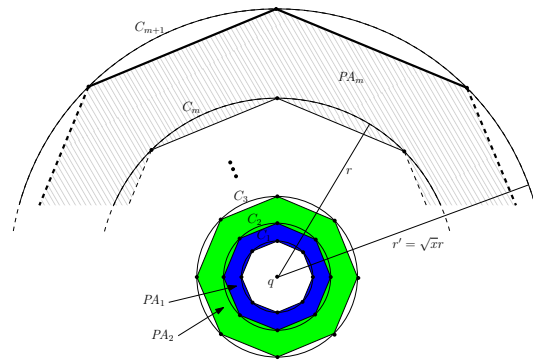
► **Corollary 8.** The polygonal ring defined by B_1 and B_2 is contained in an annulus centered at q with radii ratio $x = (1 + \frac{\varepsilon}{2})^{\frac{1}{\alpha}}$.

2.2.1 Query algorithm

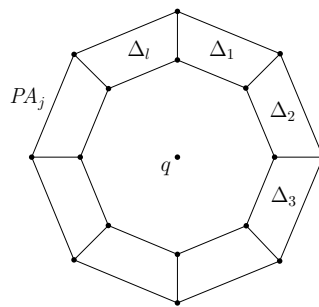
We highlight the differences with the query algorithm from Section 2.1.1. Recall that we divided the transmitters into two subsets according to whether they were closer or farther than r from the query point q . We adjust the definitions slightly by setting $r' = |qs_1|x^{m/2}$, where m is the smallest integer for which $|qs_1|x^{m/2} \geq \sqrt{x}r$, and considering a transmitter close to q whenever it lies in the interior of the regular l -gon inscribed in the circle of radius r' centered at q , see Figure 2. The set of such transmitters is the new S_c ; the remaining transmitters constitute S_f . The contribution of $s_i \in S_f$ to the sum $\text{intrf}(q)$ is, by Claim 7,

$$\frac{1}{|qs_i|^\alpha} \leq \frac{1}{r^\alpha} = \frac{\varepsilon}{2n|qs_1|^\alpha}.$$

Thus the overall contribution of the transmitters in S_f is again at most $\frac{\varepsilon}{2|qs_1|^\alpha}$.



■ **Figure 2** The polygonal rings PA_1, \dots, PA_m . The ring PA_j lies between the circles C_{j-1} and C_{j+1} , for $j > 1$.



■ **Figure 3** PA_j is the union of l trapezoids.

We now partition $A_q(|qs_1|, r')$ into m annuli, each with outer-to-inner radius ratio \sqrt{x} . For each of the $m + 1$ circles defining these annuli, draw the regular l -gon inscribed in it. Let PA_1, \dots, PA_m be the resulting sequence of polygonal rings, numbered from the innermost outwards, see Figure 2; each ring is semi-open: it includes its inner, but not its outer boundary. By Claim 7, each $PA_j, j > 1$, is contained in the union of two consecutive annuli, which in turn is an annulus of ratio x ; $S_c \cap PA_1$ is contained in the innermost annulus. Also notice that $m = O(k)$, where k is the number of annuli in the circular annulus version, so, by Lemma 5, $m = O(\frac{1}{\epsilon} \log n)$.

For each ring PA_j , we bound from above the contribution of each $s_i \in S_c \cap PA_j$ by $1/d^\alpha$, where d is the inner radius of the annulus of ratio x containing PA_j . By Lemma 3 and the subsequent corollary, we obtain a $(1 + \frac{\epsilon}{2})$ -approximation of the overall contribution of the transmitters in S_c to $\text{intrf}(q)$; $\widetilde{\text{intrf}}(q)$ is obtained by combining the two estimates, one from S_c and one from S_f .

2.2.2 Implementation

Each polygonal ring PA_j is the union of l isosceles trapezoids; moreover the i th trapezoids of all rings are homothets of each other (refer to Figure 3) and therefore are delimited by lines of exactly three different orientations. In the preprocessing stage we compute the following data structures for the set of transmitters S .

Dynamic nearest neighbor: The data structure of Kaplan *et al.* [12] (see Section 2.1.2).

Dynamic trapezoid range counting: We use l instances of the data structure, one for each family of trapezoids. For the i th family we build a three-level orthogonal range counting structure, one for each of the three edge orientations of the trapezoids in the family. The answer to a trapezoid range counting query is the number of points of S lying in the trapezoid.

A standard three-level orthogonal range counting structure requires $O(n \log^2 n)$ space, is constructed in $O(n \log^2 n)$ time, and supports $O(\log^3 n)$ -time range queries [8]. It can be modified to support insertions and deletions in $O(\log^3 n)$ amortized time using the standard partial-rebuilding technique [1, 14]. (One can use any of several different optimized variants of these structures [7, 15]. For example, He and Munro [9] describe one with linear space and $O((\log n / \log \log n)^2)$ worst-case query and amortized update time; we stay with comparison-based algorithms and do not attempt to optimize the polylogarithmic factors.)

Now given a query point q , we find its closest and second closest transmitter using the data structure for dynamic nearest neighbor in $O(\log^5 n)$ time, compute the distance r' , and construct the (polygonal) rings PA_1, \dots, PA_m , where $m = O(\frac{1}{\varepsilon} \log n)$. For each ring PA_j we proceed as follows. For each of the l trapezoids Δ_i forming PA_j , we perform an orthogonal range counting query in the i th data structure. Let n_j be the sum of the l results. Unless $j = 1$, we add to the value being computed the term $n_j \frac{1}{r_{j-1}^\alpha}$, where r_{j-1} is the radius of C_{j-1} (the inner circle of the annulus containing PA_j). If $j = 1$, we simply add the term $n_1 \frac{1}{r_1^\alpha}$. Finally, we add to the value being computed the term $|S_f| \cdot \frac{\varepsilon}{2n|qs_1|^\alpha} = (n - 1 - \sum_{j=1}^m n_j) \cdot \frac{\varepsilon}{2n|qs_1|^\alpha}$.

In summary, to implement an SINR query, we need to perform one search for the nearest and second-nearest neighbor, followed by $O(\frac{1}{\varepsilon} \log n) = O(\frac{1}{\varepsilon^{3/2}} \log n)$ range searches.

An update is applied to all the underlying data structures. The following theorem summarizes the main result of this section, while extensions can be found in the complete version [2].

► **Theorem 9.** *Given the locations of n uniform-power transmitters, one can preprocess them in $O((n/\sqrt{\varepsilon}) \log^2 n)$ time and space into a data structure that can answer approximate SINR queries in $O(\log^5 n + (1/\varepsilon^{3/2}) \log^4 n)$ time. Transmitters can be inserted in $O((1/\sqrt{\varepsilon}) \log^3 n)$ and deleted in $O(\log^5 n + (1/\sqrt{\varepsilon}) \log^3 n)$ amortized time.*

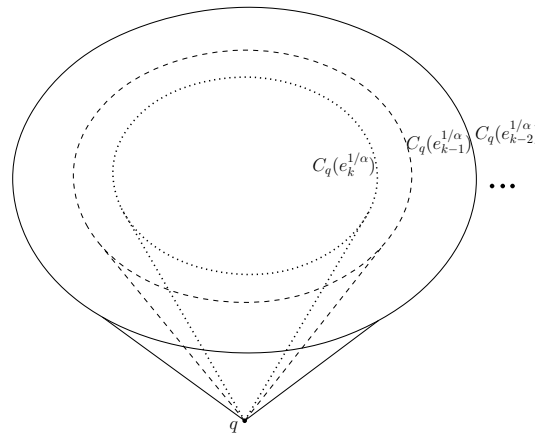
3 Non-uniform power

Let q be a receiver. For a transmitter $s \in S$, the *strength* of its signal at q is $nrg(s, q) = \frac{p(s)}{|qs|^\alpha}$ and the (multiplicatively-weighted) distance between q and s is $dist(q, s) = nrg(s, q)^{-1/\alpha} = \frac{1}{p(s)^{1/\alpha}} \cdot |qs|$. Let s be the closest transmitter to q according to $dist$. Set $intrf(q) = \sum_{s' \in S \setminus \{s\}} nrg(s', q)$, then $sinr(q, s) = \frac{nrg(s, q)}{intrf(q)}$, where we once again assume for clarity of presentation that there is no background noise, i.e., $N = 0$. When s is the closest transmitter to q , we will write $sinr(q)$ instead of $sinr(q, s)$.

Fix $\varepsilon > 0$. Again, we wish to approximate $sinr(q)$ by computing $\widetilde{intrf}(q)$ such that $intrf(q) \leq \widetilde{intrf}(q) \leq (1 + \varepsilon)intrf(q)$ and setting $\widetilde{sinr}(q) = \frac{nrg(s, q)}{\widetilde{intrf}(q)}$. As in the uniform case, we start with a more straightforward but less efficient solution and then improve it.

3.1 Conical shells

Let q be a receiver and let $s \in S$ be the closest transmitter to q according to $dist$, i.e., the one whose signal strength at q is the highest. Let s_1, \dots, s_{n-1} be the transmitters in $S \setminus \{s\}$, and assume without loss of generality that s_1 is the second closest transmitter to q among



■ **Figure 4** Partitioning $D_q(e_0^{1/\alpha}, e_k^{1/\alpha})$ into sub-shells.

the transmitters in S . Recall that $\text{intrf}(q) = \sum_{i=1}^{n-1} \text{nrg}(s_i, q)$ and that we wish to compute a value $\widetilde{\text{intrf}}(q)$ such that $\text{intrf}(q) \leq \widetilde{\text{intrf}}(q) \leq (1 + \varepsilon)\text{intrf}(q)$. We will need the following simple observation.

► **Observation 10.** *$\text{intrf}(q)$ is the sum of $n - 1$ positive terms of which $\text{nrg}(s_1, q)$ is the largest, so we have $\text{nrg}(s_1, q) \leq \text{intrf}(q) \leq (n - 1)\text{nrg}(s_1, q) \leq n \cdot \text{nrg}(s_1, q)$.*

Query algorithm. Let q be a query point. First, we find s and s_1 , as defined above. Next, we set $e_0 = \frac{\varepsilon}{2n} \text{nrg}(s_1, q)$, and divide the transmitters in $S \setminus \{s\}$ into two subsets, S_c and S_f , where S_c consists of the transmitters with signal strength at q greater than e_0 , and S_f of the remaining ones. We now approximate the overall contribution to intrf of the transmitters in S_f and in S_c separately and let $\widetilde{\text{intrf}}$ be the sum of the two approximations.

The contribution of a single transmitter $s_i \in S_f$ to the sum $\text{intrf}(q)$ is $\text{nrg}(s_i, q) \leq e_0 = \frac{\varepsilon}{2n} \text{nrg}(s_1, q)$, for a total of at most $|S_f| \cdot \frac{\varepsilon}{2n} \text{nrg}(s_1, q) \leq \frac{\varepsilon}{2} \text{nrg}(s_1, q)$ over all of S_f .

We identify the plane containing the transmitters and receivers with the xy -plane in \mathbb{R}^3 . Let $C_q(\rho)$ denote (the surface of) the vertical cone with apex q whose z -coordinate at $t = (t_x, t_y)$ is $\rho|qt|$, where $\rho > 0$ is a constant. Let $D_q(\rho_1, \rho_2)$, $\rho_2 > \rho_1 > 0$, be the set of all points in 3-space lying above (i.e., in the interior of) the cone $C_q(\rho_1)$ and below or on (i.e., not in the interior of) the cone $C_q(\rho_2)$. Informally, $D_q(\rho_1, \rho_2)$ is the region between $C_q(\rho_1)$ and $C_q(\rho_2)$; we call it a (conical) shell.

Recall that $e_0 = \frac{\varepsilon}{2n} \text{nrg}(s_1, q)$. Let $e_i = (1 + \frac{\varepsilon}{2})e_{i-1}$, for $i = 1, \dots, k - 1$, where $k - 1$ is the largest integer for which $e_i < \text{nrg}(s_1, q)$, and set $e_k = \text{nrg}(s_1, q)$. We partition the range $I = (e_0, e_k]$ of signal strengths at q into k sub-ranges, $I_1 = (e_0, e_1]$, $I_2 = (e_1, e_2]$, \dots , $I_k = (e_{k-1}, e_k]$, and count, for each sub-range I_j , the number of transmitters whose signal strength at q lies in I_j .

Consider a sub-range $I_j = (e_{j-1}, e_j]$; we want to count the number transmitters whose signal strength at q lies in I_j . This occurs whenever $e_{j-1}^{1/\alpha} |qs_i| < p_i^{1/\alpha} \leq e_j^{1/\alpha} |qs_i|$, or whenever the point $(s_i, p_i^{1/\alpha})$ in \mathbb{R}^3 lies in the shell $D_q(e_{j-1}^{1/\alpha}, e_j^{1/\alpha})$. Thus, we have reduced the problem to the difference of two conical range-counting queries.

We raise each of the transmitters $s_i \in S \setminus \{s\}$ to height $p_i^{1/\alpha}$, and preprocess the resulting set of points for conical range counting queries. If the number of points in the shell corresponding to I_j is x_j , then we add the term $e_j x_j$ to our approximation of $\text{intrf}(q)$, that is, we approximate the contribution of each transmitter s_i whose corresponding point lies in

the shell by e_j . (This corresponds to vertically projecting the point $(s_i, p_i^{1/\alpha})$ onto the cone $C_q(e_j^{1/\alpha})$.) We prove in the complete version [2] that this yields a $(1 + \frac{\varepsilon}{2})$ -approximation of the overall contribution of the transmitters in S_c to $\text{intrf}(q)$.

It remains to show that $\widetilde{\text{intrf}}(q)$, which is the sum of the approximations for S_f and for S_c , satisfies the requirements, i.e., that $\text{intrf}(q) \leq \widetilde{\text{intrf}}(q) \leq (1 + \varepsilon)\text{intrf}(q)$. From the description above it is clear that $\widetilde{\text{intrf}}(q) \geq \text{intrf}(q)$, so we only need to establish the upper bound. Indeed, using Observation 10, we conclude that

$$\widetilde{\text{intrf}}(q) \leq \frac{\varepsilon}{2|qs_1|^\alpha} + (1 + \frac{\varepsilon}{2}) \sum_{s_i \in S_c} \text{nrg}(s_i, q) \leq \frac{\varepsilon}{2} \text{intrf}(q) + (1 + \frac{\varepsilon}{2}) \text{intrf}(q) = (1 + \varepsilon) \text{intrf}(q).$$

Implementation. A straightforward calculation shows (analogously to Lemma 5) that k , the number of shells into which $D_q(e_0^{1/\alpha}, e_k^{1/\alpha})$ is partitioned, is $O(\frac{1}{\varepsilon} \log n)$.

We preprocess the set of raised transmitters for cone range reporting/counting queries. Then, given a query point q , we find s and s_1 as follows. Pick a random sample T of $\sqrt{n} \log n$ transmitters and let $t_1 \in T$ be the transmitter whose signal strength at q is the strongest. With high probability, the number of transmitters in S that are closer to q than t_1 in terms of signal strength at q is $O(\sqrt{n})$, and we perform a range reporting query with the cone C_1 corresponding to t_1 in order to find them. The closest and second-closest points among the reported points are clearly s and s_1 .

As for shell range counting queries, for each such query we issue two cone range counting queries – with the outer cone and the inner cone – and return the difference of the answers.

We omit the analysis of this version, since we describe below a more efficient variant, in which cones are replaced by pyramids.

3.2 Pyramidal shells

We now replace the conical shells by pyramidal ones to obtain an improved solution. Set $x = (1 + \frac{\varepsilon}{2})^{\frac{1}{\alpha}}$, and consider any three cones $C_q(\rho_0)$, $C_q(\rho_1)$ and $C_q(\rho_2)$ with apex at q , such that $\rho_0/\rho_1 = \rho_1/\rho_2 = \sqrt{x}$. Let $P_q(\rho_i)$ be a regular l -pyramid inscribed in $C_q(\rho_i)$, where $l = \lceil \frac{\pi}{\sqrt{2 - \frac{2}{\sqrt{x}}}} \rceil$. That is, $P_q(\rho_i)$'s apex is at q , its edges emanating from q are contained in (the surface of) $C_q(\rho_i)$, and the cross section of $P_q(\rho_i)$ and $C_q(\rho_i)$, using any horizontal cutting plane above q , is a regular l -gon and its circumcircle, respectively. The *pyramidal shell* defined by $P_q(\rho_2)$ and $P_q(\rho_1)$ and denoted $PS_q(\rho_2, \rho_1)$ is the semi-open region consisting of all points in the interior of $P_q(\rho_2)$ but not in the interior of $P_q(\rho_1)$. From Claim 7 and the observation above, it follows that $C_q(\rho_{i-1})$ is contained in $P_q(\rho_i)$, for $i = 1, 2$, and therefore, $PS_q(\rho_2, \rho_1)$ is contained in $D_q(\rho_2, \rho_0)$.

Query algorithm. We highlight the differences with the conical-shell based approach. First, we find s and s_1 , the closest and the second-closest transmitters to q , respectively, as described in detail below. Previously, the transmitters were divided into two subsets lying close to q and lying far from it, with the threshold $e_0 = \frac{\varepsilon}{2n} \cdot \text{nrg}(s_1, q)$. Here, we set $e'_0 = \frac{\text{nrg}(s_1, q)}{x^{m/2}}$, where m is the smallest integer for which $\frac{\text{nrg}(s_1, q)}{x^{m/2}} \leq \frac{e_0}{\sqrt{x}}$, and consider a transmitter close to q whenever it lies in the interior of the pyramid $P_q(e'_0)^{\frac{1}{\alpha}}$, i.e., the pyramid inscribed in $C_q(e'_0)^{\frac{1}{\alpha}}$. The contribution of a single transmitter $s_i \in S_f$ to the sum $\text{intrf}(q)$ is $\text{nrg}(s_i, q) \leq e_0 = \frac{\varepsilon}{2n} \cdot \text{nrg}(s_1, q)$, for a total of at most $\frac{\varepsilon}{2|qs|^\alpha}$, as before.

Consider the conical shell $D_q(e'_0)^{\frac{1}{\alpha}}, \text{nrg}(s_1, q)^{\frac{1}{\alpha}}$ and partition it into m conical shells, such that the ratio between the parameters of the inner and outer cone of a shell is \sqrt{x} . For each

of the $m + 1$ cones defining these conical shells, draw its inscribed regular l -pyramid. Let $P_q(\rho_1), \dots, P_q(\rho_{m+1})$ be the resulting sequence of pyramids, where $P_q(\rho_1)$ is the innermost one, and consider the corresponding sequence of m nested pyramidal shells. Notice that $m = O(k)$, where k is the number of cones in the conical shells version, so once again $m = O(\frac{1}{\varepsilon} \log n)$. Moreover, each of the pyramidal shells, except for $PS_q(\rho_2, \rho_1)$, is contained in the union of two consecutive conical shells, which is a conical shell of ratio x . For $PS_q(\rho_2, \rho_1)$, we observe that $S_c \cap PS_q(\rho_2, \rho_1)$ is contained in the innermost conical shell.

We assign a transmitter s_i to a shell $PS_q(\rho_{j+1}, \rho_j)$ if s_i lands in the shell after being raised to height $p_i^{1/\alpha}$. Now, for each shell $PS_q(\rho_{j+1}, \rho_j)$ and each $s_i \in S_c$ assigned to it, we estimate the contribution of s_i from above by $1/\rho_{j-1}^\alpha$, i.e., by projecting s_i onto the inner cone of the conical shell $D_q(\rho_{j+1}, \rho_{j-1})$ containing $PS_q(\rho_{j+1}, \rho_j)$. We obtain a $(1 + \frac{\varepsilon}{2})$ -approximation of the overall contribution of the transmitters in S_c to $\widetilde{intrf}(q)$, where the missing details can be found in the complete version [2]. Adding our previous estimate for those in S_f yields the promised $\widetilde{intrf}(q)$.

Implementation. Observe that each regular l -pyramid $P_q(\rho_j)$ is the union of l 3-sided wedges, where the i th wedge is defined by two planes of fixed orientation (perpendicular to the xy -plane) and a third plane containing the i th face of the pyramid.

In the preprocessing stage we construct l data structures over the set S , one for each family of wedges, supporting dynamic 3-dimensional 3-sided wedge range counting queries (a restricted form of simplex range counting in three dimensions). Each data structure handles wedges of the same “type”; the orientations of the two vertical bounding planes are fixed, while the orientation of the third plane varies (but remains perpendicular to the vertical plane bisecting the first two). The data structure for the i th family is a three-level search structure, where the first two levels allow us to represent the points of S that lie in the 2-sided wedge formed by the two vertical planes delimiting our 3-sided wedge, as a small collection of canonical subsets. For each canonical subset of the second level of the structure, we raise each of its points s_i to height $p_i^{1/\alpha}$ and then project it onto a vertical plane which is parallel to the bisector of the two vertical wedge boundaries. Finally, we construct for the resulting set of points a data structure for two-dimensional halfplane range counting queries. We will also need the corresponding reporting structure, see below.

Using standard tools for dynamic multilevel structures and, for example, Matoušek’s data structure for halfplane range counting at the bottom level, we obtain a structure of $O(n \text{ polylog } n)$ size that supports wedge counting (and reporting) queries in $O(n^{1/2} \text{ polylog } n)$ time and updates in $O(\text{polylog } n)$ amortized time.

Now, given a pyramidal shell $PS_q(\rho_{j+1}, \rho_j)$, we can count the number of raised points that lie in it as follows. We first perform l queries for the pyramid $P_q(\rho_{j+1})$, one in each of the l data structures, to obtain the total number of points that lie in it. We repeat the process for the pyramid $P_q(\rho_j)$ and finally subtract the latter number from the former one.

Below we describe how to find s and s_1 , the closest and second-closest points to q , in randomized $O(\sqrt{n} \log n)$ time w.h.p. plus l wedge reporting queries. Once again, an update is performed by modifying the underlying data structures. We summarize the main result of this section.

► **Theorem 11.** *One can preprocess n arbitrary-power transmitters, in $O(\frac{n}{\sqrt{\varepsilon}} \text{ polylog } n)$ time and space, into a data structure that can answer approximate SINR queries in randomized $O(\frac{\sqrt{n}}{\sqrt{\varepsilon}} \text{ polylog } n)$ time w.h.p. and perform updates in $O(\frac{1}{\sqrt{\varepsilon}} \text{ polylog } n)$ amortized time.*

Finding the closest and second-closest transmitters to q . We have assumed that given a query point q , we can find the closest s and second-closest s_1 transmitters to q , efficiently. This section deals with this initial stage.

We begin by observing that we do not really need to find s_1 , provided that we can obtain a sufficiently good approximation of $nrg_1 = nrg(s_1, q)$. Let $\widetilde{nrg}_1 \in \mathbb{R}$ such that $\widetilde{nrg}_1 \leq nrg_1 \leq (1 + \delta)\widetilde{nrg}_1$, where $\delta > 0$ is a sufficiently small constant. Then, it is easy to modify our query algorithm so that it uses \widetilde{nrg}_1 instead of nrg_1 . For simplicity of presentation, we refer in this paragraph to the algorithm using conical shells. Set $\tilde{e}_0 = \frac{\varepsilon}{2n}\widetilde{nrg}_1$. A transmitter will be considered close to q if and only if its signal strength at q is greater than \tilde{e}_0 . If $s_i \in S$ is far from q , then $nrg(s_i, q) \leq \tilde{e}_0 \leq \frac{\varepsilon}{2n}nrg_1 = \frac{\varepsilon}{2n}nrg(s_1, q)$, so the overall contribution of the transmitters in S_f is bounded by $\frac{\varepsilon}{2}nrg(s_1, q)$, as before. Next, we partition the range $I = (\tilde{e}_0, (1 + \delta)\widetilde{nrg}_1]$ into $k = O(\frac{1}{\varepsilon} \log n)$ sub-ranges, such that the ratio between the extreme values of a sub-range is at most $1 + \varepsilon/2$ and proceed exactly as before.

We now describe how to find s . Our algorithm may or may not find s_1 . However, if it does not find s_1 , it returns a transmitter $t_1 \in S$ such that $nrg(t_1, q) \leq nrg(s_1, q) \leq (1 + \delta)nrg(t_1, q)$, where $\delta > 0$ is a sufficiently small constant, so we can set $\widetilde{nrg}_1 = nrg(t_1, q)$ and apply the above modified query algorithm.

Pick a random sample T of $\sqrt{n} \log n$ transmitters and let $t_1 \in T$ be the transmitter whose signal strength at q is the strongest. This can be done in $O(\sqrt{n} \log n)$ time. With high probability the number of transmitters in S that are closer to q than t_1 , in terms of signal strength at q , is $O(\sqrt{n})$.

We first lift each transmitter $s = (s_x, s_y) \in S$ to the point $\hat{s} = (s_x, s_y, p(s)^{1/\alpha})$. Draw the cone C_1 corresponding to t_1 , i.e., the cone whose z -coordinate above point s is $nrg(t_1, q)^{1/\alpha}|qs| = (p(t_1)^{1/\alpha}/|qt_1|)|qs|$. Let l be as above and consider the l -pyramid P_1 inscribed in C_1 . Let C_0 be the cone inscribed in P_1 , so that P_1 lies between C_0 and C_1 . Notice that C_0 is the cone whose z -coordinate above point s is $(1 + \delta)nrg(t_1, q)^{1/\alpha}|qs|$ (where we set $\delta = (1 + \frac{\varepsilon}{2})^{\frac{1}{2\alpha}} - 1$).

Perform a range reporting query with P_1 (i.e., find all lifted points that lie in the interior of P_1 or on P_1). Since P_1 is inside C_1 , with high probability the number of points in P_1 is $O(\sqrt{n})$. If the resulting set is non-empty, then in randomized $O(\sqrt{n})$ time w.h.p. we can find s and also s_1 (provided the number of returned points is greater than 1).

Otherwise, if P_1 is empty, we claim that the answer to the SINR query must be NO, i.e., q cannot receive any transmitter. Indeed, in the best scenario \hat{s} lies on C_0 , where $s \in S$ is the closest transmitter to q , and the rest of the $O(\sqrt{n})$ transmitters, lifted to 3-space, lie on the cone C_1 . But this will imply that $\text{sinr}(s) < 1$. Indeed $nrg(s, q) = (1 + \delta)^\alpha nrg(t_1, q)$ and, for any other of the \sqrt{n} transmitters s' , $nrg(s', q) = nrg(t_1, q)$, implying $\text{sinr}(s) < (1 + \delta)^\alpha / \sqrt{n} \ll 1$.

If only one point lies in P_1 , then we use t_1 as an approximation of s_1 as described above.

4 Successive interference cancellation (SIC)

Fix a receiver location q . SIC is a technique that enables q to receive a specific transmitter t , even when $\text{sinr}(q, t) < \beta$. More specifically, order the transmitters s_1, \dots, s_n in S by increasing signal strength at q , assume $t = s_k$, and let $\text{sinr}_i(q)$ denote the SIN ratio for the signal of s_i at q , while ignoring transmitters s_1, \dots, s_{i-1} . If $\text{sinr}_1(q) = \text{sinr}(q, s_1) \geq \beta$, q can subtract s_1 's signal from the combined signal. If, in addition, $\text{sinr}_2(q) \geq \beta$, q can also subtract s_2 from the combined signal of the transmitters s_2, \dots, s_n , and so on. If $\text{sinr}_i(q) \geq \beta$, for $i = 1, \dots, k$, we say that *SIC succeeds for s_k at q , in k rounds*. We can simulate this process using our data structures for approximate SINR queries via a sequence

of k queries and $k - 1$ deletions, and determine (approximately) whether SIC succeeds for s_k at q . Observe that we need t only to terminate the query, while Avin et al. [4] need t to identify the part of the data structure in which to initiate the search; in particular, we can generate all the transmitters accessible via SIC given a location q in polylogarithmic time per transmitter, while they need to consult each of the n parts of the data structure. We obtain the following theorem.

► **Theorem 12.** *Assuming $t = s_k$, the simulation above can be performed in $O((1/\varepsilon^{3/2})k \text{ polylog } n)$ time in the uniform-power version, and in $O((1/\sqrt{\varepsilon})k\sqrt{n} \text{ polylog } n)$ time in the non-uniform version.*

References

- 1 Pankaj K. Agarwal. Range searching. In *Handbook of Discrete and Computational Geometry, Second Edition*, pages 809–837. CRC Press LLC, 2004. doi:10.1201/9781420035315.ch36.
- 2 Boris Aronov, Gali Bar-On, and Matthew J. Katz. Resolving SINR queries in a dynamic setting. arXiv:1804.10654.
- 3 Boris Aronov and Matthew J. Katz. Batched point location in SINR diagrams via algebraic tools. *ACM Transactions on Algorithms*, 2018. Accepted for publication; see also arXiv:1412.0962.
- 4 Chen Avin, Asaf Cohen, Yoram Haddad, Erez Kantor, Zvi Lotker, Merav Parter, and David Peleg. SINR diagram with interference cancellation. *Ad Hoc Networks*, 54:1–16, 2017. doi:10.1016/j.adhoc.2016.08.003.
- 5 Chen Avin, Yuval Emek, Erez Kantor, Zvi Lotker, David Peleg, and Liam Roditty. SINR diagrams: Convexity and its applications in wireless networks. *J. ACM*, 59(4):18:1–18:34, 2012. doi:10.1145/2339123.2339125.
- 6 Timothy M. Chan. A dynamic data structure for 3-D convex hulls and 2-D nearest neighbor queries. *J. ACM*, 57(3):16:1–16:15, 2010. doi:10.1145/1706591.1706596.
- 7 B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17(3):427–462, 1988.
- 8 M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 3rd edition, 2008. URL: <http://www.cs.ruu.nl/geobook/>.
- 9 Meng He and J. Ian Munro. Space efficient data structures for dynamic orthogonal range counting. *Comput. Geom.*, 47(2):268–281, 2014. doi:10.1016/j.comgeo.2013.08.007.
- 10 Erez Kantor, Zvi Lotker, Merav Parter, and David Peleg. The topology of wireless communication. In *Proceedings 43rd ACM Symposium on Theory of Computing, STOC 2011*, pages 383–392, 2011. doi:10.1145/1993636.1993688.
- 11 Erez Kantor, Zvi Lotker, Merav Parter, and David Peleg. Nonuniform SINR+Voronoi diagrams are effectively uniform. In *Proceedings 29th International Symposium on Distributed Computing, DISC 2015*, pages 588–601, 2015. doi:10.1007/978-3-662-48653-5_39.
- 12 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. Dynamic planar Voronoi diagrams for general distance functions and their algorithmic applications. In *Proceedings 28th ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 2495–2504, 2017. See also arXiv:1604.03654. doi:10.1137/1.9781611974782.165.
- 13 J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.
- 14 M. H. Overmars. *The Design of Dynamic Data Structures*, volume 156 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, West Germany, 1983.
- 15 D. E. Willard and G. S. Lueker. Adding range restriction capability to dynamic data structures. *J. ACM*, 32:597–617, 1985. doi:10.1145/3828.3839.