Improved Approximation Algorithms for Geometric Set Cover*

Kenneth L. Clarkson[1] and Kasturi Varadarajan[2]

[1]Bell Labs, 600 Mountain Avenue,
Murray Hill, NJ 07974, USA
clarkson@research.bell-labs.com

[2]University of Iowa,
101E, MacLean Hall, Iowa City, IA 52241, USA
kvaradar@cs.uiowa.edu

Abstract. Given a collection S of subsets of some set $\mathbb{U}$, and $\mathbb{M} \subset \mathbb{U}$, the *set cover* problem is to find the smallest subcollection $C \subset S$ that covers $\mathbb{M}$, that is, $\mathbb{M} \subseteq \bigcup(C)$, where $\bigcup(C)$ denotes $\bigcup_{Y \in C} Y$. We assume of course that S covers $\mathbb{M}$. While the general problem is NP-hard to solve, even approximately, here we consider some geometric special cases, where usually $\mathbb{U} = \mathbb{R}^d$. Combining previously known techniques [4], [5], we show that polynomial-time approximation algorithms with provable performance exist, under a certain general condition: that for a random subset $R \subset S$ and nondecreasing function $f(\cdot)$, there is a decomposition of the complement $\mathbb{U} \setminus \bigcup(R)$ into an expected at most $f(|R|)$ regions, each region of a particular simple form. Under this condition, a cover of size $O(f(|C|))$ can be found in polynomial time. Using this result, and combinatorial geometry results implying bounding functions $f(c)$ that are nearly linear, we obtain $o(\log c)$ approximation algorithms for covering by fat triangles, by pseudo-disks, by a family of fat objects, and others. Similarly, constant-factor approximations follow for similar-sized fat triangles and fat objects, and for fat wedges. With more work, we obtain constant-factor approximation algorithms for covering by unit cubes in $\mathbb{R}^3$, and for guarding an x-monotone polygonal chain.

1. Introduction

Given a collection S of subsets of some set $\mathbb{U}$, and $\mathbb{M} \subset \mathbb{U}$, the *set cover* problem is to find the smallest subcollection $C \subset S$ that covers $\mathbb{M}$, that is, $\mathbb{M} \subseteq \bigcup(C)$. We assume

that S covers $\mathbb{M}$. In the geometric setting, almost always $\mathbb{U} = \mathbb{R}^d$. For example, $\mathbb{M}$ could be a finite set of points, and S a given finite set of balls. The family S can be specified implicitly; an example is when S is the set of all unit balls. Another interesting example is when $\mathbb{M}$ is the set of points in a simple polygon in $\mathbb{R}^2$, and S is the set of visibility regions of the vertices of the polygon.

The general set cover problem is hard to solve, even approximately, and the simple greedy algorithm has performance very close to best possible for a polynomial-time algorithm, assuming a certain widely believed complexity theoretic assumption [17], [25]. Even in the geometric setting, most versions of the problem are believed to be NP-hard, and indeed NP-hardness has been shown for several versions [30], [29]. (In some cases, hardness of approximation has been shown as well [2].) The focus of current work is therefore on obtaining approximation algorithms that run in polynomial time. Often one obtains a polynomial-time algorithm guaranteeing a logarithmic factor approximation by reducing the geometric set cover problem to the combinatorial set cover problem [6], [20], [24].

In many cases the approximation factor can be made $O(\log c)$, where c is the size of the optimal solution. Such a result was achieved for the case of polytope approximation in general dimension [8], by applying the *iterative reweighting* approach [23], [35], [9] to an associated set cover problem. (The reduction of polytope approximation to set cover was observed by Mitchell and Suri [31].)

Brönnimann and Goodrich [4] showed that a very similar algorithm applies in the general setting of set systems with finite VC dimension. A key observation of theirs was a connection with ε-*nets*. Consider the subset $\mathbb{U}_\varepsilon \subset \mathbb{U}$, comprising those points of $\mathbb{U}$ contained not just in one set in S, but in at least $\varepsilon|S|$ of them. An ε-net of S is a cover for such heavily covered points. (Thus, the set cover problem is to find the smallest possible $(1/|S|)$-net for $\mathbb{M} = \mathbb{U}$.) Suppose that the family S has a $(1/r)$-net of size $g(r)$, for every r with $1 \leq r \leq |S|$.[1] The algorithm of Brönnimann and Goodrich guarantees an approximation factor of $O(g(c)/c)$, where c is the size $|C|$ of the optimal solution. For many cases where $g(r) = O(r \log r)$ [7], [18], their algorithm gives an $O(\log c)$ approximation. Moreover, if $g(r) = O(r)$, such as when S is a family of disks in $\mathbb{R}^2$ or halfspaces in $\mathbb{R}^3$ [28], [26], they obtain an $O(1)$ approximation algorithm.

There have been a few other interesting instances where the $O(\log c)$ factor has been improved upon. Some recent ones include an $O(\sqrt{\log n})$ approximation factor for covering an isothetic n-vertex polygon (with holes) using a minimum number of rectangles contained in the polygon [22], and an $O(1)$ approximation algorithm for guarding an x-monotone polygonal chain [1].

Hochbaum and Maass [19] consider the problem of covering a set of points in the plane with the smallest number of unit disks. For this and some related problems, they present algorithms, that for any $\varepsilon > 0$, run in polynomial time and return a $(1 + \varepsilon)$-approximation. Since any unit disk may be chosen in the cover, the problem has a different flavor from that of covering the points using the minimum number of disks chosen from a set of specified unit disks.

---

[1] In fact, the requirement is somewhat stronger—that such ε-nets exist and can be computed efficiently in a weighted setting [4].

1.1. *Our Results*

We extend results giving small ε-nets for halfspaces [26] to a more general setting [11], [5], making a connection between the combinatorial complexity of the union of a set of objects and size of a net for the set of objects. Suppose that S is a set of objects, say triangles in the plane for concreteness. Suppose that there is a bound $f(j) \geq j$ on the combinatorial complexity of the boundary of the union of any j objects from S. (More precisely, we need the number of simple regions in a canonical decomposition of the exterior of the union of the j objects to be at most $f(j)$, and $f(j)$ is a nondecreasing function of j.) As demonstrated in Theorem 2.2, there is a $(1/r)$-net of size $O(f(r))$, for every $r \leq |S|$. This is easily shown by applying a "repair" or "alteration" technique, where a random sample is used to divide the problem into roughly small subproblems, followed by a "repair" step in each subproblem. The approach is very similar to that of Chazelle and Friedman [5].

As noted, this implies a polynomial-time algorithm that guarantees an $O(f(c)/c)$ approximation factor for covering a set $\mathbb{M}$ of points using objects from S, where c is the size of the optimal cover [4, Theorem 3.2]. (Note that the result is only interesting for $f(r) = o(r \log |S|)$; otherwise the greedy algorithm could be used.)

We give several applications of this result. If S is a set of fat triangles in the plane, then the combinatorial complexity of the union of any j elements of S is $O(j \log \log j)$ [27], and thus we obtain $(1/r)$-nets of size $O(r \log \log r)$ for fat triangles. This implies, as stated in Theorem 3.1, a polynomial-time algorithm for the corresponding set cover problem, for which an approximation factor of $O(\log \log c)$ is guaranteed. If the triangles in S have roughly the same diameter, then the union of any j elements from S has a combinatorial complexity of $O(j)$ [27], and we obtain $(1/r)$-nets of size $O(r)$ and an algorithm for the corresponding set cover problem that guarantees an $O(1)$ approximation. There are other applications in this vein.

Such cover problems are related to wireless network planning, where the sets in S correspond to antenna coverage areas. Prior work has sometimes approximated the coverage areas as circular disks [12], but often such an idealized model would be far from ideal. Thus the results for more general "fat" objects reported here are relevant.

Another problem that can be viewed as a special case of wireless network planning is that of guarding a one-dimensional terrain. Here, the problem is to guard the region above an x-monotone polygonal chain using the minimum number of point guards, who are constrained to be on the chain. The problem was recently studied by Ben-Moshe et al. [1] who presented a fairly sophisticated polynomial-time algorithm that guarantees an $O(1)$ approximation. We show that a different polynomial-time constant-factor approximation algorithm can be derived quite naturally from our paradigm. The approximation result is Theorem 3.5, and applies a generalization of the "Order Claim" of [1] to show, in Lemma 3.4, that an associated sequence is Davenport–Schinzel.

We next consider the case where S is a set of axis-parallel unit cubes in $\mathbb{R}^3$. Boissonnat et al. [3] have shown that the combinatorial complexity of the union of j such cubes is $O(j)$. Such a bound is however not readily available for a canonical decomposition of the exterior of the union. We nevertheless exploit the fact that all the cubes have roughly the same size to obtain a $(1/r)$-net of size $O(r)$ and, as stated in Theorem 3.7, a polynomial algorithm for the corresponding set cover problem that guarantees an approximation factor of $O(1)$.

2. General Results

2.1. *Small ε-Nets from Small 0-Region Sets*

In a geometric setting, the set cover and ε-net problems often have the helpful property that the complement of the union of a subcollection of *objects* can be decomposed canonically into locally defined pieces called *regions*. This "object/region" framework was introduced in [11] and [10] and is similar to the "starter/stopper" framework of Mulmuley [32]. For completeness and clarity, we present the framework in a self-contained manner.

Let S be a collection of n subsets of a universe $\mathbb{U}$. The elements of S are called *objects*, and S is the input of a geometric set cover problem. Let $\mathcal{F}$ be a set of subsets of $\mathbb{U}$, which we call *regions*. In construction problems [11], the desired computation is usually to find all the regions which are *defined* by S, and which have an empty intersection with S. (The precise meaning of "defined" will be made clear shortly.) In geometric set cover problems, the complement of the union of S can usually be decomposed into a union of regions defined by S. Moreover, each of these regions is defined by a constant number b of objects. These notions can be formalized as follows [11]: for some integer b, let $S^{(b)}$ be the collection of subsets of S with at most b elements. Let δ be a relation between $\mathcal{F}$ and $S^{(b)}$. We say that a region y in $\mathcal{F}$ is *defined* by $B_y \in S^{(b)}$ if $F \delta B_y$. We assume that each element of $S^{(b)}$ defines $O(1)$ elements of $\mathcal{F}$. The set of regions defined by S is denoted by

$$\mathcal{F}(S) = \{y \in \mathcal{F} \mid y \delta B_y, B_y \in S^{(b)}\}.$$

A region $y \in \mathcal{F}(S)$ may have a non-empty intersection with some object $s \in S$, in which case we say that s *meets* or *conflicts with* y. The regions in $\mathcal{F}(S)$ that are in conflict with no object in S are called 0-*regions* of S, and their set is denoted by $\mathcal{F}_0(S)$. As in [11], we assume that regions do not conflict with objects that define them, that is, if $y \delta B_y$, then $y \cap \bigcup(B_y) = \emptyset$. We are also interested in regions in $\mathcal{F}(S)$ that meet objects in S. We say that a region in $\mathcal{F}(S)$ is a j-*region* if it meets j objects in S; the set of j-regions of S is denoted by $\mathcal{F}_j(S)$.

For instance, $\mathbb{U}$ could be $\mathbb{R}^2$ and S is a set of triangles, or discs, in $\mathbb{R}^2$. The regions $\mathcal{F}(S)$ would be (generalized) trapezoids with vertical sides and lower and upper sides that are either segments or circular arcs, those that occur in the trapezoidal decomposition (or vertical visibility map) of the arrangement of S', denoted by $\mathcal{T}(S')$, where S' is any subset of S. Each region $y \in \mathcal{F}$ is in the trapezoidal map $\mathcal{T}(X)$ of a subset $X \subset S$ of at most four objects [11], and so $b = 4$. For $y \in \mathcal{F}$ and $X \in S^{(4)}$, we set $y \delta X$ if (1) y is one of the trapezoids in $\mathcal{T}(X)$ and y is contained in the complement of $\bigcup(X)$, and (2) for no proper subset $X' \subset X$ is y a trapezoid in $\mathcal{T}(X')$ and contained in the complement of $\bigcup(X')$. Hence a region y is in $\mathcal{F}(S)$ if and only if for some subset $X \in S^{(4)}$ it is defined as the closed trapezoid of $\mathcal{T}(X)$ that is contained in the complement of $\bigcup(X)$, minus the boundary points that belong to X. It is a j-region if it has a nonempty intersection with j objects in S. In particular, the regions in $\mathcal{F}_0(S)$ are the regions in $\mathcal{F}(S)$ which are entirely contained in the complement of the union of S.

As in [11], we can also define $\mathcal{F}(R)$ for a subset $R \subset S$ as the set of regions y such that $y \delta B_y$ for some $B_y \in R^{(b)}$, and $\mathcal{F}_0(R)$ and $\mathcal{F}_j(R)$ similarly. Note that $y \in \mathcal{F}_0(R)$

if and only if $B_y \in R$ and no $s \in R$ conflicts with y. Even though a region may be in $\mathcal{F}_0(R)$, it may be in $\mathcal{F}_j(S)$, that is, a 0-region with respect to R might be a j-region with respect to S for some $j > 0$. Our assumption that each element of $S^{(b)}$ defines $O(1)$ regions implies that $|\mathcal{F}(R)| = O(|R|^b)$.

Given a *configuration system*, that is, a combination of $\mathbb{U}$, objects S, regions $\mathcal{F}$, parameter b, and relation δ, we can express the regions $\mathcal{F}(R)$ for each $R \subseteq S$ and the notion of conflict. As stated, 0-regions of a set R must not intersect with objects in R. For geometric set cover problems, it is also crucial to assume that any point in $\mathbb{U}$ not covered in R is in some 0-region of R. That is, we must have $\mathbb{U} \setminus \bigcup(R) = \bigcup(\mathcal{F}_0(R))$. We say that a configuration system that has this property is *complete*. For instance, for a set S of triangles in $\mathbb{R}^2$, the semiclosed trapezoids in $\mathcal{F}_0(S)$ do cover the entire complement of $\bigcup(S)$.

It sometimes happens that for some $y \in \mathcal{F}(S)$, there is more than one minimal B_y that defines y. To reduce problems with such degenerate situations, one may consider the regions not only as subsets of $\mathbb{U}$, but as pairs (y, B_y) where B_y defines y. The definition of conflict is then extended by saying that that an object $s \in S$ meets or *conflicts* with (y, B_y) not only if $s \cap y$ is nonempty, but also if s takes precedence over a member of B_y, for tie-breaking or other reasons specific to an application. The sets $\mathcal{F}_0(R)$ and $\mathcal{F}_j(R)$ will be generalized to comprise such pairs, and a pair $(y, B_y) \in \mathcal{F}_0(R)$ if and only if $B_y \subseteq R$ and no $s \in R$ conflicts with (y, B_y) in this broader way. Even with this generalization, however, a complete configuration system still ensures $\mathbb{U} \setminus \bigcup(R) = \bigcup(\mathcal{F}_0(R))$ for any $R \subseteq S$, where here $\bigcup(\mathcal{F}_0(R))$ is defined as $\bigcup_{(y,B_y) \in \mathcal{F}_0(R)} y$. We may confuse (y, B_y) with y at times, but the situation should be clear in context.

The completeness property is the ingredient that enables the link between ε-nets and the object-region framework. Several properties of the geometric set cover problem follow from that framework. A basic property within the framework is the following version of ε-nets, proven in the objects/regions framework [7], and also in the framework of bounded VC dimension [18].

Lemma 2.1 (Large ε-Nets). *For a given complete configuration system, there is a constant K such that, for a random subset $R \subset S$ of size $Kr \log r$, with probability at least $1 - 1/r$, every 0-region of R is a $(\leq n/r)$-region with respect to S, that is, a j-region with respect to S for some $j \leq n/r$.*

Since our assumption here is that a point not covered by R is in some 0-region of R, it follows that R satisfying the condition of the lemma is an ε-net, for $\varepsilon = 1/r$ and $|R| \leq Kr \log r$. Call an ε-net under such conditions a *large ε-net*. (See Section 1 for the definition of an ε-net.)

Proof. See [7]; also, since the regions here have finite VC-dimension, the similar results of [18] apply. The proof is simply the union bound, applied to every (y, B_y); the probability is small that a particular j-region of S, with $j \geq n/r$, is a 0-region of R, and there are $O(n^b)$ j-regions. $\qquad\square$

We observe that Lemma 2.1 generalizes to yield a large ε-net for any subset $S' \subseteq S$.

Using the existence of large ε-nets and the objects/regions framework, we can describe the following scheme for small ε-nets. As mentioned, these results are very similar to those of Chazelle and Friedman [5]. To simplify some expressions, we assume that $|\mathcal{F}_0(R)| \geq R$ for each $R \subseteq S$.

Theorem 2.2. *For a given complete configuration system, let $\hat{f}(r) := E|\mathcal{F}_0(R)|$, and $f(r) := \max_{r' < r} \hat{f}(r')$, where $R \subset S$ is a random subset of size r. Then given $r \geq 2b$, there is a $(1/r)$-net of size $O(f(r))$.*

Proof. The construction is as follows. Pick a random subset $R' \subset S$ of size r. For each $y \in \mathcal{F}_0(R')$, suppose there are j objects in S that meet y; denote their set by S_y, and let $j' = (r/n)j$. If $j' < 1$, let $R_y := \emptyset$; otherwise, let R_y be a large $(1/j')$-net for S_y. Such an R_y will have size at most $Kj' \log j'$. Then $R := R' \cup (\bigcup_{y \in \mathcal{F}_0(R)} R_y)$ is a $(1/r)$-net for S, by construction. Indeed, any point p not covered by R is not covered by R' either, and so belongs to some 0-region $y \in \mathcal{F}_0(R')$ since the configuration system is complete. Any object s that contains p must meet y, and so belong to S_y. However, p also cannot be covered by R_y by construction of R, and by the $(1/j')$-net property of R_y, there can be at most $(j'n/r)/j' = n/r$ objects containing p.

The expected size of R is at most $r + \sum_{y \in \mathcal{F}_0(R')} Kj' \log j'$, and the latter term can be bounded using Theorem 3.6 of [11] with $c = 2$. For the work of $W(\binom{j}{2})$ of that theorem, with $j = j'n/r$, to be at least $Kj' \log j'$, it suffices to take $W(x)$ to be $4K(r/n)\sqrt{x} \log(x(r^2/n^2))$. Theorem 3.6 of [11] then implies that

$$E[|R|] - r \leq E \sum_{y \in \mathcal{F}_0(R')} Kj' \log j' \leq E \sum_{y \in \mathcal{F}_0(R')} W\left(\binom{j}{2}\right)$$

$$\leq O\left(W\left(\frac{n^2}{(r-b)^2} K_{2,b}\right)\right) \hat{f}(r),$$

with b constant. The term $K_{2,b} = E[|\mathcal{F}_2(R')|]/E[|\mathcal{F}_0(R')|]$ of the theorem is no more than $K'\hat{f}(\alpha r)/\hat{f}(r)$, for a constant K' and a value $\alpha \leq 1$, by virtue of Theorem 3.2 of [11]. Hence the expected size of R is $r + O(\sqrt{\hat{f}(\alpha r)\hat{f}(r)}) = O(f(r))$.   □

Computation of ε-Nets. We note that the proof of Theorem 2.2 suggests a natural randomized algorithm to compute a net. In order to bound the running time of such an algorithm in our general framework, however, we state some assumptions about the primitives that are available to us. These assumptions hold for all the applications in this paper. To simplify some of the expressions below, we assume that $|\mathcal{F}_0(R)| \leq O(|R| \log |R|)$. In any case, it is only when this assumption holds that our approach gives improved bounds for geometric set cover. We assume that for any $R \subseteq S$, we can compute the set $\mathcal{F}_0(R)$ of 0-regions in $O(|R|^2 \log |R|)$ *primitive steps.* The running time of a primitive step depends on the application—usually, it is $O(1)$, but in some applications it is larger. We also assume that given a region $y \in \mathcal{F}(S)$ and an object $s \in S$, we can tell in one primitive step if s meets y.

Lemma 2.3. *For any $S' \subseteq S$ and $1 \le r \le |S'|$, we can compute in $O(|S'|r \log^2 r)$ expected primitive steps a subset $R \subseteq S'$ of size at most $Kr \log r$ such that every 0-region of R is a $(\le |S'|/r)$-region with respect to S'.*

Proof. We use the natural algorithm corresponding to Lemma 2.1. We pick a random subset $R \subseteq S'$ of size $Kr \log r$ in $O(r \log r)$ time. We compute $\mathcal{F}_0(R)$ in $O(r^2 \log^3 r) = O(|S'|r \log^2 r)$ primitive steps. For each $y \in \mathcal{F}_0(R)$, we compute S'_y in $O(|S'|)$ primitive steps, where S'_y is the set of regions in S' that meet y. If $|S'_y| \le |S'|/r$ for each $y \in \mathcal{F}_0(R)$, we stop. Otherwise, we repeat these steps with a new random sample.

It is clear that when the algorithm terminates, it does so with an R as required by the lemma. Each iteration of the algorithm takes $O(|S'|r \log^2 r)$ primitive steps. By Lemma 2.1, the expected number of iterations is $1 + O(1/r)$. $\qquad\square$

We are now ready to derive the algorithmic equivalent of Theorem 2.2.

Theorem 2.4. *For a given complete configuration system, and $1 \le r \le n$, we can compute a $(1/r)$-net of size $O(f(r))$ for S in $O(nr \log r)$ primitive steps.*

Proof. Following the proof and notation of Theorem 2.2, we pick a random sample $R' \subseteq S$ of size r in $O(r)$ time. We compute $\mathcal{F}_0(R')$ in $O(r^2 \log r)$ primitive steps. For each $y \in \mathcal{F}_0(R')$ we compute the set S_y using $O(n)$ primitive steps. This gives a total of $O(nr \log r)$ primitive steps over all the 0-regions. Using Lemma 2.3, we compute a large $(1/j')$-net for each S_y in expected $O((n/r)j'^2 \log^2 j')$ primitive steps. The expected number of steps over all $y \in \mathcal{F}_0(R')$ is $(n/r) \sum_{y \in \mathcal{F}_0(R')} j'^2 \log^2 j'$. Arguing as in Theorem 2.2 (with $c = 3$), we can bound this by $O(nf(r)/r) = O(n \log r)$.

We now check if $|R|$ is within twice the bound on $E[|R|]$ that is guaranteed by the proof of Theorem 2.2. By Markov's inequality, the probability of this is at least $1/2$. If $|R|$ is too high, we repeat the entire process; otherwise we terminate with R.

Since the probability of termination is at least $1/2$ in each iteration, the expected number of iterations is $O(1)$. The expected number of steps in each iteration is bounded by $O(nr \log r)$. $\qquad\square$

We will need a version of this theorem that applies to the weighted setting. Suppose that each object $s \in S$ has an integer weight $w_s \ge 1$, and let $W = \sum_{s \in S} w_s$. An ε-net in this context is a subset $R \subseteq S$ that covers every point $p \in \mathbb{U}$ that is covered by a subset of S whose total weight is at least εW. To generalize Theorem 2.4 to this context, we may simply make w_s "copies" of each object $s \in S$, and extend the conflict relation with tie-breaking to allow at most one copy to contribute to the definition of a region. Applying Theorem 2.4 to the resulting configuration system gives a $(1/r)$-net of size $O(f(r))$. We will only need to consider the case where each $w_s \le n^4$. With some care, the expected number of primitive steps of the generalized algorithm still remains $O(nr \log r)$. (We do not actually make many copies of each object in the computation, but just keep track of the weights.)

2.2. *Small Covers from Small ε-Nets*

We return to the geometric set cover problem where we want to cover a target set $\mathbb{M} \subseteq \mathbb{U}$ using subsets of objects from S. We need access to a subroutine, called a *verifier*, that given a potential cover $R \subseteq S$, tells us whether R covers $\mathbb{M}$; if R does not cover $\mathbb{M}$, it returns an uncovered point $p \in \mathbb{M}$ along with the objects in S that contain p.

Theorem 2.5. *For a given complete configuration system, with $f(r)$ as in the last theorem, suppose there is a cover $C \subset S$ of size c for subset $\mathbb{M} \subset U$. Then a cover of $\mathbb{M}$ of size $O(f(4c))$ can be found in expected $O(nc^2 \log n \log c)$ primitive steps plus the time needed for $O(c \log n)$ calls to the verifier that provide the verifier with potential covers of size $O(c \log c)$.*

(Note that for particular instances a stronger time bound can be obtained.)

Proof. Brönnimann and Goodrich [4, Theorem 3.2] show that a cover of size $O(f(4c))$ can be found by $O(c \log n)$ calls to a procedure that computes a $(1/r)$-net of S of size $O(f(r))$ for $r \leq 4c$ (the weight w_s of each $s \in S$ in each of these calls is an integer between 1 and n^4), and $O(c \log n)$ calls to a verifier (each call to the verifier gives it a potential cover of size at most $O(f(c)) = O(c \log c)$). The theorem follows by using the algorithm of Theorem 2.4, generalized to the weighted setting, for computing the $(1/r)$-net.

In the algorithm given by Brönnimann and Goodrich, ε-nets are found many times, for slightly different sets of weights. If we only desire polynomial running time, an alternative approach is to solve the linear programming relaxation, and find a single ε-net, as discussed by Even et al. [16].   □

3. Applications

3.1. *Covering by Fat Triangles or Regions*

Our first applications of the general results follow fairly directly from existing combinatorial bounds and the low complexity of trapezoidal decompositions in the plane.

Theorem 3.1. *There is a randomized $O((n + m)c^2 \log n \log c)$-time algorithm that, given a set $\mathbb{M}$ of m points in $\mathbb{R}^2$, and a set S of n fat triangles that cover $\mathbb{M}$, computes a subset $S' \subseteq S$ of $O(c \log \log c)$ triangles that cover $\mathbb{M}$, where c is the size of the smallest subset of S that covers $\mathbb{M}$. If in addition the diameter of the triangles in S are in the range $[1, \alpha]$, where $\alpha \geq 1$ is a constant, we can in fact compute a cover of size $O(c)$.*

Proof. (Sketch) It is long known that the union of r fat triangles has combinatorial complexity $O(r \log \log r)$. (See [27], which also gives a definition of fatness.) The same bound applies to the canonical trapezoidal decomposition of the complement of their

union [32]. We can then apply Theorem 2.2 with these trapezoids as the regions, and obtain a $(1/r)$-net of size $O(f(r)) = O(r \log \log r)$. For the primitives needed for the computation in Theorem 2.4, we note that the union of r fat triangles can be computed in $O(r \log r \log \log r)$ time [27, Theorem 5.1]. The same time bound holds for the trapezoidal decomposition of the exterior of the union [32]. Furthermore, given a trapezoidal region and a triangle, we can determine if they conflict in $O(1)$ time. We may therefore assume that each primitive step takes one unit of time. As for the verifier needed by Theorem 2.5, we check if $\mathbb{M}$ is covered by $R \subseteq S$ in $O(m|R|)$ by checking each point in $\mathbb{M}$ against each triangle in R; if we find that some $p \in \mathbb{M}$ is not covered, we return the list of all triangles in S that contain p in additional $O(n)$ time. The verifier therefore runs in $O(m|R| + n)$ time. The running time claimed in the theorem now follows from Theorem 2.5.

Similar remarks apply for fat triangles of approximately the same size, relying on the sharper bounds known for the complexity of their union [27].   □

Theorem 3.2. *There is a randomized $O((n + m)c^2 \log n \log c)$-time algorithm that, given a set $\mathbb{M}$ of m points in $\mathbb{R}^2$ and a set S of n (α, β)-fat objects of approximately the same size that covers $\mathbb{M}$, computes a subset $S' \subseteq S$ of size $O(\lambda_{s+2}(c))$ that covers $\mathbb{M}$, where c is the size of the smallest subset of S that covers $\mathbb{M}$. Here s is the maximum number of intersections between the boundaries of two objects in S.*

The quantity $\lambda_{s+2}(n)$, related to the complexity of Davenport–Schinzel sequences [34], is a very-nearly linear function of n.

Proof (*Sketch*).   We use a result of Efrat [14] that the combinatorial complexity of the boundary of the union of k such fat objects is $O(\lambda_{s+2}(k))$, and proceed as in the case of triangles. We assume that each primitive step takes constant time, and that we can check whether an object contains a point in constant time.   □

We note also the following (using the brief unpublished summary of Sharir [33]). Here a Jordan region is a planar region bounded by a closed Jordan curve.

- **Pseudo-disks.** Pseudo-disks are Jordan regions where each pair of bounding Jordan curves intersects at most twice. The union of r such regions has no more than $6r - 12$ such intersection points on its boundary [21], and therefore its trapezoidization has $O(r)$ complexity, implying a constant-factor approximation algorithm.
- **Jordan curves.** A union complexity of $O(n\alpha(n))$ is known for a collection of regions that are each the intersection of a Jordan region with the nonnegative y halfplane, with also each pair of bounding curves intersecting at most three times, and not counting intersections on the x-axis [13]. This implies an $O(\alpha(c))$-factor approximation algorithm.
- **Fat wedges.** An arrangement of r fat wedges has $O(r)$ complexity [15], and so a constant-factor approximation algorithm.

3.2.   *Guarding a Monotone Polygonal Chain*

Let P be an x-monotone polygonal chain in $\mathbb{R}^2$ with m vertices. Let $\mathcal{G} := \{g_1, \ldots, g_n\}$ be a set of points, which we call *guards*, on P. Say that a guard g lying on polygonal chain P *sees* a point p if the line segment $\overline{gp}$ does not intersect the region in $\mathbb{R}^2$ that is strictly below P.

Consider the set $\mathbb{M}_P$ of points in $\mathbb{R}^2$ that are on or above P. For $g \in \mathcal{G}$, let $\mathrm{Vis}(g) := \{p \in \mathbb{R}^2 \mid g \text{ sees } p\}$, the *visibility polygon* of g, be the set of all points seen by g. The problem of guarding P is that of covering the set $\mathbb{M}_P$ by a small subset of $S := \{\mathrm{Vis}(g) \mid g \in \mathcal{G}\}$. (We assume that S itself covers $\mathbb{M}_P$.) For $S' \subseteq S$, the complement of the region covered by S' is the area between P and the lower envelope of the visibility polygons in S'. Each point on the x-axis has some corresponding point on the lower envelope with the same x-coordinate, perhaps at infinity. It will be helpful, for showing the existence of a low-complexity, locally defined description of the lower envelope, to consider visibility from the left or right separately. It will also be helpful to break ties among the guards determining the lower envelope at a given x coordinate.

Complexity of the Lower Envelope.   Say that g sees p *from the left* if g sees p and $x(g) \leq x(p)$, where $x(p)$ is the x-coordinate of point p; define visibility from the right analogously. For $g \in \mathcal{G}$, let $\mathrm{Lvis}(g) := \{p \in \mathbb{R}^2 \mid g \text{ sees } p \text{ from the left}\}$, the set of points that g sees from the left. Let $S_L := \{\mathrm{Lvis}(g) \mid g \in \mathcal{G}\}$.

Fix some subset $\mathcal{H} \subseteq \mathcal{G}$. For guard $g \in \mathcal{H}$ and point p say that g *owns p from the left* (relative to $\mathcal{H}$) if g sees p from the left, and is the leftmost guard in $\mathcal{H}$ that sees p. For the next few paragraphs, the "from the left" condition is assumed and not stated explicitly.

In other words, the space above P is partitioned by ownership, each point with its owner. Referring to Fig. 1, the ownership regions of the guards in $\mathcal{H} = \{a, b, c, d, e\}$ are shown, omitting some of unbounded regions owned by a and d.

If $g \in \mathcal{H}$ owns point p, and also p is the lowest point at x-coordinate $x(p)$ owned by any guard in $\mathcal{H}$, say that g owns $x(p)$ *at* p. If some x-coordinate x is owned by no point in $\mathcal{H}$, say that x has the owner NULL.

Figure 1 also shows the *ownership diagram* of a set of guards $\mathcal{H} \subseteq \mathcal{G}$, with respect to P. (This is for ownership from the left, but similar definitions and claims apply for ownership from the right.) The (left) ownership diagram is the partition of the x axis obtained from the connected components of each equivalence class of the relation "x
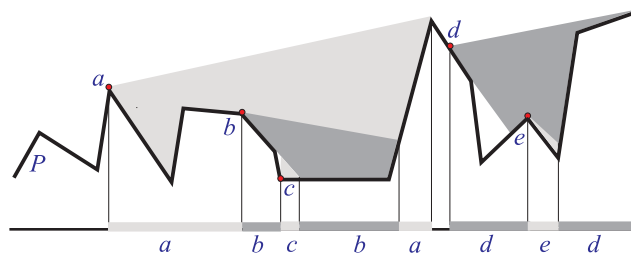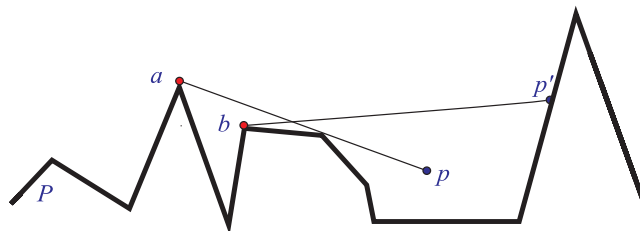

Fig. 1.   Ownership regions.

Fig. 2. Example for Lemma 3.3.

and x' have the same owner." Such components are intervals (or single points), and so this diagram is a sequence of intervals, each with one owner. Call the corresponding sequence of owners, but excluding NULL, the *ownership sequence* for $\mathcal{H}$. A key claim for a bound on the length of this sequence is the following, a slight generalization of Lemma 2.1 of [1].

Lemma 3.3. *Suppose* $a, b \in \mathcal{H} \subset \mathcal{G}$ *and* $x, x' \in \mathbb{R}$ *have* $x(a) < x(b) < x < x'$. *Suppose also* a *owns* x (*relative to* $\mathcal{H}$) *at a point* p, *and* $p' = (x', y')$ *is seen by* b. *Then* p' *is seen by* a *also.*

Proof. (See Fig. 2. Note that in the figure, a owns $x(p)$ at p, even though p is not on P.) Since a owns x at p, a sees p, and so P is not above line segment $\overline{ap}$. Since b is on P and between a and p, b in particular is not above $\overline{ap}$. Similarly, P is not above segment $\overline{bp'}$. Also p is not above $\overline{bp'}$: if p were above $\overline{bp'}$, it would be seen by b, and since P is not above $\overline{bp'}$, b would also see some point below p, but with the same x coordinate, contradicting the assumption that a owns x at p. So b and P are not above $\overline{ap}$, and p and P are not above $\overline{bp'}$. Therefore a sees p', as claimed. □

Lemma 3.4. *An ownership sequence for any set* $\mathcal{H}$ *of* r *guards is an* $(r, 2)$ *Davenport–Schinzel sequence, and therefore has length at most* $2r - 1$. *It follows that the number of ownership intervals is no more than* $2r$.

Proof. An $(r, 2)$ Davenport–Schinzel sequence [34] is a sequence of r symbols with no successive entries identical, and with no subsequence of the form $a \cdots b \cdots a \cdots b$. Consider $a, b \in \mathcal{H}$, and first suppose that $x(a) < x(b)$, as in the previous theorem. It may be that a owns intervals before b (with a smaller x coordinate than $x(b)$), and it may be that b owns some intervals to its right, but if a owns some x-coordinate at point p, strictly to the right of b, then from the previous lemma, any point p' with $x(p') > x(p)$ seen by b is also seen by a. Since $x(a) < x(b)$, such a coordinate would be owned by a if either a or b owns it, and so could not be owned by b. Therefore, there is no ownership sequence of the form $a \cdots b \cdots a \cdots b$. A similar argument works if $x(b) < x(a)$, and thus the first claim of the lemma follows. The length bound for such sequences is long known [34]. The final claim follows because there is at most one interval with owner NULL; this is the interval to the left of all the guards in $\mathcal{H}$. □

Guarding in the Objects/Regions Framework.   We employ Theorem 2.4 to compute a $(1/r)$-net for the set S_L of size $O(r)$. (Recall that such a net is a subset $S' \subset S_L$ such that any point belonging to more than $|S_L|/r$ sets from S_L also belongs to some set in S'.) In order to apply the theorem, we indicate explicitly how the configurations and conflicts are defined. There is a configuration corresponding to every interval in the ownership diagram for subsets of $\mathcal{G}$ of size at most 3. Consider an interval I in the ownership diagram of $\{a, b, c\} \subset \mathcal{G}$, and suppose b owns each $x \in I$, a owns the interval immediately to the left of I, and c owns the interval immediately to the right of I.[2] The set $\{a, b, c\}$ defines this configuration. (The *region* of this configuration is the set $\{(x, y) \mid x \in I, (x, y) \notin \mathrm{Lvis}(b)\}$.) A guard $d \in G \setminus \{a, b, c\}$ can conflict with this configuration in two ways:

1. Relative to the set $\{a, b, c, d\}$, d rather than b owns some point $x' \in I$. This of course happens if d sees some point with x-coordinate x' that lies below the point p at which b owns x' with respect to $\{a, b, c\}$. Note that this also happens if d sees p and $x(d) < x(b)$.
2. Relative to the set $\{a, b, c, d\}$, b continues to own all points in I but the interval immediately to the left of I is owned by d and not a. Because of the way we break ties in defining ownership, this is not a pathological situation at all. A conflict also occurs if d owns the interval immediately to the right of I in the ownership diagram of $\{a, b, c, d\}$.

With these definitions, observe that the size of $\mathcal{F}_0(\mathcal{H})$, for any subset $\mathcal{H} \subset \mathcal{G}$, is exactly equal to the number of intervals in the ownership diagram of $\mathcal{H}$, which is $O(|\mathcal{H}|)$ by Lemma 3.4. We can therefore use Theorem 2.2 to obtain a $(1/r)$-net for S_L of size $O(f(r)) = O(r)$. For the computation in Theorem 2.4, we first consider the complexity of computing the 0-regions for a set $\mathcal{H}$ of r guards. Clearly, it is enough to compute the ownership diagram of $\mathcal{H}$. This can be done using a standard divide-and-conquer [34]— we partition $\mathcal{H}$ into two sets of roughly equal size, and recursively compute the respective ownership diagrams. In the merge step we overlay the two diagrams to obtain a partition of the x-axis into $O(|\mathcal{H}|)$ intervals. Within each such interval, we have to resolve ownership between two guards. This is easily seen to require $O(m)$ time per interval, since the visibility polygon of any guard consists of $O(m)$ edges. The running time of the overall algorithm is therefore $O(mr)$. To determine whether a guard conflicts with a configuration, it is sufficient to compute the ownership diagram of at most four guards, which takes $O(m)$ time. We can therefore assume that each primitive step takes $O(m)$ time. Theorem 2.4 then gives us an algorithm to compute a $(1/r)$-net of size $O(r)$ in $O(nmr \log r)$ time.

We define S_R in a manner symmetric to S_L, and note that the union of a $(1/2r)$-net for S_L and a $(1/2r)$-net for S_R is a $(1/r)$-net for S. For the verifier in Theorem 2.5, we have to determine if a set $\mathcal{H}$ of r guards sees $\mathbb{M}_P$. For this, we compute the ownership diagram of $S'_L = \{\mathrm{Lvis}(h) \mid h \in \mathcal{H}\}$ (resp. S'_R) in $O(mr)$ time as described above. We then compute a set $\mathcal{I}$ (resp. $\mathcal{I}'$) of $O(mr)$ intervals on the x-axis that correspond to those portions of P that are not in $\bigcup(S'_L)$ (resp. $\bigcup(S'_R)$). We then check if $\bigcup(\mathcal{I}) \cap \bigcup(\mathcal{I}')$ is

---

[2] If a itself owns the interval immediately to the right of I, then such a configuration would be considered to be defined by the subset $\{a, b\}$.

empty in $O(mr)$ time. If it is empty, then $\mathcal{H}$ sees $\mathbb{M}_P$. Otherwise, we have a point p on P that is not seen by $\mathcal{H}$. We can then generate the list of guards in $\mathcal{G}$ that do not see p in $O(nm)$ time. The running time of the verifier is $O(nm + mr)$.

Applying Theorem 2.5, we obtain:

Theorem 3.5. *Let P be an x-monotone polygonal chain in $\mathbb{R}^2$ with m vertices. Let $\mathcal{G} := \{g_1, \ldots, g_n\} \subset \mathbb{R}^2$ be guards on P, such that $\mathbb{M}_P$ is seen by $\mathcal{G}$. Then a subset $C \subset \mathcal{G}$ that also sees $\mathbb{M}_P$, of size within $O(1)$ of optimal, can be found in randomized $O(nmc^2 \log n \log c)$ time.*

3.3. *Covering with Cubes*

We now consider the set cover problem where $\mathbb{M}$ is a set of m points in $\mathbb{R}^3$ and S is a set of n *axis-parallel* unit cubes in $\mathbb{R}^3$ that cover $\mathbb{M}$. We first show that for any $1 \leq r \leq n$, there is a $(1/r)$-net for S of size $O(r)$, by applying Theorem 2.2. That is, there is a subset $T \subseteq S$ with $|T| = O(r)$ such that any point that is contained in at least n/r cubes from S is also contained in some cube from T. We also present a randomized polynomial-time algorithm to compute such a $(1/r)$-net, by applying Theorem 2.4.

Let G be the vertices of a grid in $\mathbb{R}^3$ of side $\frac{1}{2}$. That is,

$$G := \{(i/2, j/2, k/2) \mid i, j, k, \text{ are integers}\}.$$

We "assign" each cube $C \in S$ to some point in G that lies in the interior of C. (Note that there is always at least one such point.) Let $S[p] \subseteq S$ denote the set of cubes assigned to the point $p \in G$. For each $p \in G$ such that $|S[p]| \geq n/\alpha r$, where $\alpha > 0$ is a suitably large constant, we compute a $(n/\alpha r|S[p]|)$-net $T[p]$ for $S[p]$ of size $O(|S[p]|\alpha r/n)$ using the procedure described below. Let

$$T := \bigcup_{p \in G; |S[p]| \geq n/\alpha r} T[p].$$

Clearly,

$$|T| \leq \sum_{p \in G} O\left(\frac{|S[p]|\alpha r}{n}\right) = O(\alpha r).$$

We argue that T is a $(1/r)$-net for S. Let $q \in \mathbb{R}^3$ be any point that is covered by at least n/r cubes from S. Consider the cube E of side length 2 that is centered at q. Each cube in S that contains q is contained in E, so it must have been assigned to one of the at most α points in $G \cap E$. It follows that there is a point $p \in G \cap E$ such that $S[p]$ has at least $n/\alpha r$ cubes that contain q. Thus $T[p]$, and hence T, will have a cube that contains q.

A Net for a Cluster. We now consider the computation of a $(1/r)$-net, for any $1 \leq r \leq |S[p]|$, for a "cluster" $S[p]$. The special property of $S[p]$ is that there is a point, namely p, that lies in the interior of all the cubes in $S[p]$. For any nonempty subset $S' \subseteq S[p]$, we define a canonical trapezoidization of the boundary of the union of the cubes in S'. This is obtained by taking, for each face of each cube in S', a canonical trapezoidization of

the (isothetic polygon corresponding to the) portion of the face that lies on the boundary of the union of S'. Let $\Gamma(S')$ denote the canonical set of trapezoids thus obtained.

Proposition 3.6. *For any subset $S' \subseteq S[p]$, $|\Gamma(S')| = O(|S'|)$.*

Proof. Boissonnat et al. [3] show that the combinatorial complexity of the boundary of the union of cubes in S' is $O(|S'|)$. The proposition follows because $\Gamma(S')$ is linearly bounded by the combinatorial complexity of the boundary of the union of S'.   □

We define the "region" μ_τ corresponding to the trapezoid $\tau \in \Gamma(S')$ to be the set of all points $q \in \mathbb{R}^3$ for which τ intersects the segment $\overline{qp}$ in the relative interior of the segment. It is easy to see, using the fact that p lies in the interior of all the cubes in $S[p]$, that the regions $\{\mu_\tau \mid \tau \in \Gamma(S')\}$ partition the exterior of the union of the cubes in S'. The sets that define and conflict with a region μ_τ are defined in the standard way: a cube $C \in S[p]$ will conflict with μ_τ if C contains a point in μ_τ. Since $|\mathcal{F}_0(S')| = O(|S'|)$, we can apply Theorem 2.2 to obtain a $(1/r)$-net for $S[p]$ of size $O(f(r)) = O(r)$. For the primitives needed for the computation in Theorem 2.4, we first note that the union of a set $S' \subseteq S[p]$ of r cubes can be computed in $O(r^2 \log r)$ time: we compute the complement of $\bigcup(S')$ within each face of each cube in S' in $O(r \log r)$ time using a divide-and-conquer algorithm similar to the one used for the union of fat triangles [27, Theorem 5.1]. Having computed $\bigcup(S')$, we can compute $\Gamma(S')$ in $O(r \log r)$ time. We thus obtain $\mathcal{F}_0(S') = \{\mu_\tau \mid \tau \in \Gamma(S')\}$ in $O(r^2 \log r)$ time. Clearly, we can check if a 0-region intersects a cube in $O(1)$ time. We may therefore assume that each primitive step takes one unit of time. Putting everything together, and applying Theorem 2.4, we see that a $(1/r)$-net for S of size $O(r)$ can be computed in $O(nr \log r)$ time.

As for the verifier needed by Theorem 2.5, we check if $\mathbb{M}$ is covered by $R \subseteq S$ in $O(m|R|)$ by checking each point in $\mathbb{M}$ against each cube in R; if we find that some $p \in \mathbb{M}$ is not covered, we return the list of all cubes in S that contain p in additional $O(n)$ time. The verifier therefore runs in $O(m|R| + n)$ time. Applying Theorem 2.5, we obtain:

Theorem 3.7. *There is a randomized $O((n + m)c^2 \log n \log c)$-time algorithm that, given a set $\mathbb{M} \subseteq \mathbb{R}^3$ of m points and a set S of n axis-parallel unit cubes in $\mathbb{R}^3$ that cover $\mathbb{M}$, computes a subset $T \subseteq S$ of $O(c)$ cubes that cover $\mathbb{M}$, where c is the size of the smallest subset of cubes from S that covers $\mathbb{M}$.*

We remark that the problem of covering a given set of points by the smallest number of axis-parallel unit cubes, where we are allowed to pick any axis-parallel unit cube in our cover, admits a polynomial-time approximation scheme [19].

4. Conclusion

We have presented a method for obtaining polynomial-time algorithms with improved approximation guarantees for a variety of geometric set cover problems. We have also

demonstrated that the running times of the algorithms are quite reasonable, that is, bounded by polynomials of low degree. However, we did not attempt to obtain the best possible running times. In fact, the running times for most of the applications considered here can probably be improved by a factor of c (the size of the optimal cover) using existing techniques: the application of randomized incremental construction should allow such an improvement for the construction of small ε-nets, and the use of preprocessing for point location would allow similar improvement for the verifier of a cover.

It is worth exploring other versions of the geometric set cover problem where better approximation guarantees can be obtained via improved bounds on ε-nets. Our work also highlights the need for a deeper understanding of the connection between bounds on the union and the size of ε-nets.

We close with a natural open problem, which is to obtain polynomial-time approximation algorithms with a sublogarithmic guarantee for the geometric set cover problem where $\mathbb{M}$ is a set of m points in $\mathbb{R}^3$, and S is a set of n unit balls whose union covers $\mathbb{M}$.

Acknowledgments

References

1. B. Ben-Moshe, M. J. Katz, and J. S. B. Mitchell. A constant-factor approximation algorithm for optimal terrain guarding. *Proc. ACM–SIAM Symposium on Discrete Algorithms*, pages 515–524, 2005.

2. P. Berman and B. Dasgupta. Approximating rectilinear polygonal cover problems. *Algorithmica*, 17(4):331–356, 1997.

3. J.-D. Boissonnat, M. Sharir, B. Tagansky, and M. Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. *Discrete Comput. Geom.*, 19(4):473–484, 1998.

4. H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14:263–279, 1995.

5. B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10:229–249, 1990.

6. V. Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4:233–235, 1979.

7. K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

8. K. L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd Workshop on Algorithms and Data Structures*, pages 246–252. Volume 709 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1993.

9. K. L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42(2):488–499, 1995.

10. K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Applic.*, 3:185–121, 1993.

11. K. L. Clarkson and P. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

12. G. Călinescu, I. Măndoiu, P. Wan, and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. *Mobile Networks Applic.*, 9:101–111, 2004.

13. H. Edelsbrunner, L. Guibas, J. Hershberger, J. Pach, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink. On arrangements of Jordan arcs with three intersections per pair. *Discrete Comput. Geom*, 4:523–539, 1989.

14. A. Efrat. The complexity of the union of (α, β)-covered objects. *SIAM J. Comput.*, 34(4):775–787, 2005.
15. A. Efrat, G. Rote, and M. Sharir. On the union of fat wedges and separating a collection of segments by a line. *Comput. Geom. Theory Applic.*, 3:277–288, 1994.
16. G. Even, D. Rawitz, and S. Shahar. Hitting sets when the VC-dimension is small. *Inform. Process. Lett.* 95(2):358–362, 2005.
17. U. Feige. A threshold of ln *n* for approximating set cover. *J. ACM*, 45:634–652, 1998.
18. D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.
19. D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32:130–136, 1985.
20. D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.
21. K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.
22. V. S. A. Kumar and H. Ramesh. Covering rectilinear polygons with axis-parallel rectangles. *SIAM J. Comput.*, 32(6):1509–1541, 2003.
23. N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. In *Proc. 28th IEEE Symp. on Foundations of Computer Science*, pages 68–77, 1987.
24. L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390, 1975.
25. C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41:960–981, 1994.
26. J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Applic.*, 2(3):169–186, 1992.
27. J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. *SIAM J. Comput.*, 23:154–169, 1994.
28. J. Matoušek, R. Seidel, and E. Welzl. How to net a lot with little: small ε-nets for disks and halfspaces. In *Proc. 6th Annu. ACM Symp. on Computational Geometry*, pages 16–22, 1990.
29. M. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13:182–196, 1984.
30. N. Megiddo and A. Tamir. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.*, 1:194–197, 1982.
31. J. S. B. Mitchell and S. Suri. Separation and approximation of polyhedral objects. *Comput. Geom. Theory Applic.*, 5:95–114, 1995.
32. K. Mulmuley. *Computational Geometry*: *An Introduction Through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
33. M. Sharir. The complexity of the union of planar regions. http://imu.org.il/Meeting97/Abstracts/sharir.ps, 1997.
34. M. Sharir and P. K. Agarwal. *Davenport–Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.
35. E. Welzl. Partition trees for triangle counting and other range searching problems. In *Proc. Fourth ACM Symp. on Computational Geometry*, pages 23–33, 1988.