Exact Algorithms and APX-Hardness Results for Geometric Set Cover

Timothy M. Chan* Elyot Grant†

Abstract

We study several geometric set cover problems in which the goal is to compute a minimum cover of a given set of points in Euclidean space by a family of geometric objects. We give a short proof that this problem is APX-hard when the objects are axis-aligned fat rectangles, even when each rectangle is an ϵ-perturbed copy of a single unit square. We extend this result to several other classes of objects including almost-circular ellipses, axis-aligned slabs, downward shadows of line segments, downward shadows of graphs of cubic functions, 3-dimensional unit balls, and axis-aligned cubes, as well as some related hitting set problems. Our hardness results are all proven by encoding a highly structured minimum vertex cover problem which we believe may be of independent interest.

In contrast, we give a polynomial-time dynamic programming algorithm for 2-dimensional set cover where the objects are pseudodisks containing the origin or are downward shadows of pairwise 2-intersecting x-monotone curves. Our algorithm extends to the weighted case where a minimum-cost cover is required.

1 Introduction

In a *geometric set cover problem*, we are given a range space $(X, \mathcal{S})$—a universe X of points in Euclidean space and a pre-specified configuration $\mathcal{S}$ of regions or geometric objects. The goal is to select a minimum-cardinality subfamily $\mathcal{C} \subseteq \mathcal{S}$ such that each point in X lies inside at least one region in $\mathcal{C}$. In the *weighted* generalization, we are also given a vector of positive costs $\mathbf{w} \in \mathbb{R}^{\mathcal{S}}$ and we wish to minimize the total cost of all objects in $\mathcal{C}$. Instances without costs are termed *unweighted.*

Geometric covering problems have found many applications to real-world engineering and optimization problems in ares such as wireless network design, image compression, and circuit-printing [11] [15]. Unfortunately, even for very simple classes of objects such as unit disks or unit squares in the plane, computing the exact minimum set cover is strongly NP-hard [18]. Consequently, much of the research surrounding geometric set cover has focused on approximation algorithms. A large number of constant and almost-constant approximation algorithms have been obtained for various hitting set and set cover problems of low VC-dimension via ϵ-net based methods [8] [13]. These methods have spawned a rich literature concerning techniques for obtaining small ϵ-nets for various weighted and unweighted geometric range spaces [12] [1] [23]. Results include constant-factor linear programming based approximation algorithms for set cover with objects like fat rectangles in the plane and unit cubes in $\mathbb{R}^3$.

However, these approaches have limitations. So far, ϵ-net based methods have been unable to produce anything better than constant-factor approximations, and typically the constants involved are quite large. Their application is also limited to problems involving objects with combinatorial restrictions such as low union complexity (see [12] for details). A recent construction due to Pach and Tardos has proven that small ϵ-nets need not always exist for instances of the *rectangle cover problem*—geometric set cover where the objects are axis-aligned rectangles in the plane [21]. In fact, their result implies that the integrality gap of the standard set cover LP for the rectangle cover problem can be as big as $\Theta(\log n)$. Despite this, a constant approximation using other techniques has not been ruled out.

The approximability of problems like rectangle cover also has connections to related capacitated covering problems [10]. Recently, Bansal and Pruhs used these connections, along with a weighted ϵ-net based algorithm of Varadarajan [23], to obtain a breakthrough in approximating a very general class of machine scheduling problems by reducing them to a weighted covering problem involving points *4-sided boxes* in $\mathbb{R}^3$—axis-aligned cuboids abutting the xy and yz planes [9]. The 4-sided box cover problem generalizes the rectangle cover problem in $\mathbb{R}^2$ and thus inherits its difficulty.

In light of the drawbacks of ϵ-net based methods, Mustafa and Ray recently proposed a different approach. They gave a PTAS for a wide class of unweighted geometric hitting set problems (and consequently, related set cover problems) via a *local search* technique [20]. Their method yields PTASs for:

- Geometric hitting set problems involving halfspaces in $\mathbb{R}^3$ and pseudodisks (including disks, axis-aligned squares, and more generally homothetic copies of identical convex regions) in the plane.
- By implication, geometric set cover problems with

*David R. Cheriton School of Computer Science, University of Waterloo, tmchan@uwaterloo.ca

†Department of Combinatorics and Optimization, University of Waterloo, egrant@uwaterloo.ca

lower half-spaces in $\mathbb{R}^3$ (by geometric duality, see [5]), disks in $\mathbb{R}^2$ (by a standard lifting transformation that maps disks to lower halfspaces in $\mathbb{R}^3$, see [5]), and translated copies of identical convex regions in the plane (again, by duality).

Their results currently do not seem applicable to set cover with general pseudodisks in the plane. On a related note, Erlebach and van Leeuwen have obtained a PTAS for the weighted version of geometric set cover for the special case of unit squares [14].

1.1 Our Results

We present two main results—a series of APX-hardness proofs for several geometric set cover and related hitting set problems, and a polynomial-time exact algorithm for a different class of geometric set cover problems.

For a set Y of points in the plane, we define the *downward shadow* of Y to be the set of all points (a, b) such that there is a point $(a, y) \in Y$ with $y \geq b$.

Theorem 1 *Unweighted geometric set cover is APX-hard with each of the following classes of objects:*

(C1) *Axis-aligned rectangles in* $\mathbb{R}^2$*, even when all rectangles have lower-left corner in* $[-1, -1+\epsilon] \times [-1, -1+\epsilon]$ *and upper-right corner in* $[1, 1+\epsilon] \times [1, 1+\epsilon]$ *for an arbitrarily small* $\epsilon > 0$.

(C2) *Axis-aligned ellipses in* $\mathbb{R}^2$*, even when all ellipses have centers in* $[0, \epsilon] \times [0, \epsilon]$ *and major and minor axes of length in* $[1, 1 + \epsilon]$.

(C3) *Axis-aligned slabs in* $\mathbb{R}^2$*, each of the form* $[a_i, b_i] \times [-\infty, \infty]$ *or* $[-\infty, \infty] \times [a_i, b_i]$.

(C4) *Axis-aligned rectangles in* $\mathbb{R}^2$*, even when the boundaries of each pair of rectangles intersect exactly zero times or four times.*

(C5) *Downward shadows of line segments in* $\mathbb{R}^2$.

(C6) *Downward shadows of (graphs of) univariate cubic functions in* $\mathbb{R}^2$.

(C7) *Unit balls in* $\mathbb{R}^3$*, even when all the balls contain a common point.*

(C8) *Axis-aligned cubes in* $\mathbb{R}^3$*, even when all the cubes contain a common point and are of similar size.*

(C9) *Half-spaces in* $\mathbb{R}^4$.

Additionally, unweighted geometric hitting set is APX-hard with each of the following classes of objects:

(H1) *Axis-aligned slabs in* $\mathbb{R}^2$.

(H2) *Axis-aligned rectangles in* $\mathbb{R}^2$*, even when the boundaries of each pair of rectangles intersect exactly zero times or four times.*

(H3) *Unit balls in* $\mathbb{R}^3$.

(H4) *Half-spaces in* $\mathbb{R}^4$.

Mustafa and Ray ask if their local improvement approach might yield a PTAS for a wider class of instances; Theorem 1 immediately rules this out for all of the covering and hitting set problems listed above by proving that no PTAS exists for them unless P = NP. Item (C1) demonstrates that even tiny perturbations can destroy the behaviour of the local search method. (C2) rules out the possibility of a PTAS for arbitrarily fat ellipses (that is, ellipses that are within ϵ of being perfect circles). (C5) and (C6) stand in contrast to our algorithm below, which proves that geometric set cover is polynomial-time solvable when the objects are downward shadows of horizontal line segments or quadratic functions. In the case of (C4) and (H2), the intersection graph of the rectangles is a comparability graph (and hence a perfect graph); even then, neither set cover nor hitting set admits a PTAS. (C7), (C8), (C9), (H3), and (H4) complement the result of Mustafa and Ray by showing that their algorithm fails in higher dimensions.

All of our hardness results are proven by directly encoding a restricted version of unweighted set cover, which we call *SPECIAL-3SC*:

Definition 2 *In an instance of SPECIAL-3SC, we are given a universe* $U = A \cup W \cup X \cup Y \cup Z$ *comprising disjoint sets* $A = \{a_1, \ldots, a_n\}$, $W = \{w_1, \ldots, w_m\}$, $X = \{x_1, \ldots, x_m\}$, $Y = \{y_1, \ldots, y_m\}$*, and* $Z = \{z_1, \ldots, z_m\}$ *where* $2n = 3m$*. We are also given a family* $\mathcal{S}$ *of* $5m$ *subsets of* U *satisfying the following two conditions:*

- *For each* $1 \leq t \leq m$*, there are integers* $1 \leq i < j < k \leq n$ *such that* $\mathcal{S}$ *contains the sets* $\{a_i, w_t\}$, $\{w_t, x_t\}$, $\{a_j, x_t, y_t\}$, $\{y_t, z_t\}$*, and* $\{a_k, z_t\}$ *(summing over all* t *gives the* $5m$ *sets contained in* $\mathcal{S}$*.)*
- *For all* $1 \leq t \leq n$*, the element* a_t *is in exactly two sets in* S.

In section 2, we show:

Lemma 3 *SPECIAL-3SC is APX-hard.*

Our second result is a dynamic programming algorithm that exactly solves weighted geometric set cover with various simple classes of objects:

Theorem 4 *There exists a polynomial-time exact algorithm for the weighted geometric set cover problem involving downward shadows of pairwise 2-intersecting x-monotone curves in* $\mathbb{R}^2$*. Moreover, it runs in* $O(m^2n(m + n))$ *time on a set system consisting of* n *points and* m *regions.*

Our algorithm is a generalization and simplification of a similar algorithm appearing in [10] for a combinatorial problem equivalent to geometric set cover with downward shadows of horizontal line segments in $\mathbb{R}^2$. We believe that our current presentation is much shorter and cleaner; in particular, we do not require shortest path as a subroutine. We can also extend our algorithm to some related geometric set systems:

Corollary 5 *There exists a polynomial-time exact algorithm for the weighted geometric set cover problem involving a configuration of pseudodisks in $\mathbb{R}^2$ where the origin lies within the interior of each pseudodisk. Furthermore, it runs in $O(mn^2(m+n))$ time on a set system consisting of n points and m pseudodisks.*

Proof. We apply Lemma 2.11 of [4] to transform the arrangement of pseudodisks into a topologically equivalent one where all the psuedodisks are star-shaped about the origin, and then apply a standard polar-to-cartesian transformation, mapping the star-shaped pseudodisks to the downward shadows of 2-intersecting x-monotone functions on $[0, 2\pi)$. □

1.2 Related Work

The problem of assembling a given rectilinear polygon from a minimum number of (possibly overlapping) axis-aligned rectangles was first proven to be MAX-SNP-complete by Berman and Dasgupta [6], which rules out the possibility of a PTAS unless P = NP. Since set cover with axis-aligned rectangles can encode these instances, it too is MAX-SNP-complete. However, the proof in [6] cannot be applied to produce an instance using only fat rectangles. The current best approximability for the rectilinear polygon cover problem on a polygon with n edges is $O(\sqrt{\log n})$ via an algorithm of Kumar and Ramesh [19].

In his recent Ph.D. thesis, van Leeuwen proves APX-hardness for geometric set cover and dominating set with axis-aligned rectangles and ellipses in the plane [24]. Har-Peled provides a simple proof that set cover with triangles is APX-hard, even when all triangles are fat and of similar size [16]. Har-Peled also notes that set cover with circles (that is, with boundaries of disks) is APX-hard for a similar reason. However, neither the results of van Leeuwen nor Har-Peled can be directly extended to fat axis-aligned rectangles or fat ellipses.

There are few non-trivial examples of geometric set cover problems that are known to be poly-time solvable. Har-Peled and Lee give a dynamic programming algorithm for weighted cover of points in the plane by half-planes [17]; their method runs in $O(n^5)$ time on an instance with n points and half-planes. Our algorithm both generalizes theirs and reduces the run time by a factor of n. Ambühl et al. give a poly-time dynamic programming algorithm for weighted covering of points in a narrow strip using unit disks [3]; their method appears to be unrelated to ours.

An interesting PTAS result is that of Har-Peled and Lee, who give a PTAS for minimum weight cover with any class of fat objects, provided that each object is allowed to expand by a small amount. Our results show that without allowing this, a PTAS cannot be obtained.

2 APX-Hardness of SPECIAL-3SC

In this section, we prove Lemma 3. We recall that a pair of functions (f, g) is an L-reduction from a minimization problem A to a minimization problem B if there are positive constants α and β such that for each instance x of A, the following hold:

(L1) The function f maps instances of A to instances of B such that $\mathrm{OPT}(f(x)) \leq \alpha \cdot \mathrm{OPT}(x)$.

(L2) The function g maps feasible solutions of $f(x)$ to feasible solutions of x such that $c_x(g(y)) - \mathrm{OPT}(x) \leq \beta \cdot \big(c_{f(x)}(y) - \mathrm{OPT}(f(x))\big)$, where c_x and $c_{f(x)}$ are the cost functions of the instances x and $f(x)$ respectively.

We exhibit an L-reduction from minimum vertex cover on 3-regular graphs (hereafter known as 3VC) to SPECIAL-3SC. Since 3VC is APX-hard [2], this proves that SPECIAL-3SC is APX-hard (see [22] for details).

Given an instance x of 3VC on edges $\{e_1, \ldots, e_n\}$ with vertices $\{v_1, \ldots, v_m\}$ where $3m = 2n$, we define $f(x)$ be the SPECIAL-3SC instance containing the sets $\{a_i, w_t\}$, $\{w_t, x_t\}$, $\{a_j, x_t, y_t\}$, $\{y_t, z_t\}$, and $\{a_k, z_t\}$ for each 4-tuple (t, i, j, k) such that v_t is a vertex incident to edges e_i, e_j, and e_k with $i < j < k$. To define g, we suppose we are given a solution y to the SPECIAL-3SC instance $f(x)$. We take vertex v_t in our solution $g(y)$ of the 3VC instance x if and only if at least one of $\{a_i, w_t\}$, $\{a_j, x_t, y_t\}$, or $\{a_k, z_t\}$ is taken in y. We observe that g maps feasible solutions of $f(x)$ to feasible solutions of x since e_i is covered in $g(y)$ whenever a_i is covered in y.

Our key observation is the following:

Proposition 6 $\mathrm{OPT}(f(x)) = \mathrm{OPT}(x) + 2m$.

Proof. For $1 \leq t \leq m$, let $\mathcal{P}_t = \{\{w_t, x_t\}, \{y_t, z_t\}\}$ and $\mathcal{Q}_t = \{\{a_i, w_t\}, \{a_j, x_t, y_t\}, \{a_k, z_t\}\}$. Call a solution $\mathcal{C}$ of $f(x)$ *segregated* if for all $1 \leq t \leq m$, $\mathcal{C}$ either contains all sets in $\mathcal{P}_t$ and no sets in $\mathcal{Q}_t$, or contains all sets in $\mathcal{Q}_t$ and no sets in $\mathcal{P}_t$.

Via local interchanging, we observe that there exists an optimal solution to $f(x)$ that is segregated. Additionally, our function g, when restricted to segregated solutions of $f(x)$, forms a bijection between them and feasible solutions of x. We check that g maps segregated solutions of size $2m + k$ to solutions of x having cost precisely k, and the result follows. □

Proposition 6 implies that f satisfies property (L1) with $\alpha = 5$, since $\mathrm{OPT}(x) \geq \frac{m}{2}$. Moreover, $c_x(g(y)) +$

$2m \le c_{f(x)}(y)$ since both $\{w_t, x_t\}$ and $\{y_t, z_t\}$ must be taken in y whenever v_t is not taken in $g(y)$, and at least three of $\{\{a_i, w_t\}, \{w_t, x_t\}, \{a_j, x_t, y_t\}, \{y_t, z_t\}, \{a_k, z_t\}\}$ must be taken in y whenever v_t is taken in $g(y)$. Together with Proposition 6, this proves that g satisfies property (L2) with $\beta = 1$, completing the proof that (f, g) is an L-reduction.

3 Encodings of SPECIAL-3SC via Geometric Set Cover

In this section, we prove Theorem 1 using Lemma 3, by encoding instances of various classes of geometric set cover and hitting set problems as instances of SPECIAL-3SC. The beauty of SPECIAL-3SC is that it allows many of our geometric APX-hardness results to follow immediately from simple "proofs by pictures" (see Figure 3). The key property of SPECIAL-3SC is that we can divide the elements into two sets A and $B = W \cup X \cup Y \cup Z$, and linearly order B in such a way that all sets in $\mathcal{S}$ are either two adjacent elements from B, one from B and one from A, or two adjacent elements from B and one from A. We need only make $[w_t, x_t, y_t, z_t]$ appear consecutively in the ordering of B.

For (C1), we simply place the elements of A on the line segment $\{(x, x-2) : x \in [1, 1+\epsilon]\}$ and place the elements of B, in order, on the line segment $\{(x, x+2) : x \in [-1, -1+\epsilon]\}$, for a sufficiently small $\epsilon > 0$. As we can see immediately from Figure 3, each set in $\mathcal{S}$ can be encoded as a fat rectangle in the class (C1).

(C2) is similar. It is not difficult to check that each set can be encoded as a fat ellipse in this class.

For (C3), we place the elements of A on a horizontal line (the top row). For each $1 \le t \le m$, we create a new row containing $\{w_t, x_t\}$ and another containing $\{y_t, z_t\}$ as shown in Figure 3. This time, we will need the second property in Definition 2—that each a_i appears in two sets. If $\{a_i, w_t\}$ is the first set that a_i appears in, we place w_t slightly to the left of a_i; if it is the second set instead, we place w_t slightly to the right of a_i. Similarly, the placement of x_t, y_t (resp. w_t) depends on whether a set of the form $\{a_j, x_t, y_t\}$ (resp. $\{a_k, w_t\}$) is the first or second set that a_j (resp. a_k) appears in. As we can see from Figure 3, each set in $\mathcal{S}$ can be encoded as a thin vertical or horizontal slab.

(C4) is similar to (C3), with the slabs replaced by thin rectangles. For example, if $\{a_i, w_t\}$ and $\{a_i, w_{t'}\}$ are the two sets that a_i appears in, with w_t located higher than $w_{t'}$, we can make the rectangle for $\{a_i, w_t\}$ slightly wider than the rectangle for $\{a_i, w_{t'}\}$ to ensure that these two rectangles intersect 4 times.

For (C5), we can place the elements of A on the ray $\{(x, -x) : x > 0\}$ and the elements of B, in order, on the ray $\{(x, x) : x < 0\}$. The sets in $\mathcal{S}$ can be encoded as downward shadows of line segments as in Figure 3.

(C6) is similar. One way is to place the elements of A on the line segment $\ell_A = \{(x, x) : x \in [-1, -1+\epsilon]\}$ and the elements of B (in order) on the line segment $\ell_B = \{(x, 0) : x \in [1.5, 1.5+\epsilon]\}$. For any $a \in [-1, -1+\epsilon]$ and $b \in [1.5, 1.5+\epsilon]$, the cubic function $f(x) = (x-b)^2[(a+b)x - 2a^2]/(b-a)^3$ is tangent to ℓ_A and ℓ_B at $x = a$ and $x = b$. (The function intersects $y = 0$ also at $x = 2a^2/(a+b) \gg 1.5+\epsilon$, far to the right of ℓ_B.) Thus, the size-2 sets in $\mathcal{S}$ can be encoded as cubics. A size-3 set $\{a_j, x_t, y_t\}$ can also be encoded if we take a cubic with tangents at a_j and x_t, shift it upward slightly, and make x_t and y_t sufficiently close.

For (C7), we place the elements in A on a circular arc $\gamma_A = \{(x, y, 0) : x^2 + y^2 \le 1,\ x, y \ge 0\}$ and the elements in B (in order) on the vertical line segment $\ell_B = \{(0, 0, z) : z \in [1-2\epsilon, 1-\epsilon]\}$. (This idea is inspired by a known construction [7], after much simplification.) We can ensure that every two points in A have distance $\Omega(\sqrt{\epsilon})$ if $\epsilon \ll 1/n^2$. The technical lemma below allows us to encode all size-2 sets (by setting $b = b'$) and size-3 sets by unit balls containing a common point.

Lemma 7 *Given any $a \in \gamma_A$ and $b, b' \in \ell_B$, there exists a unit ball that (i) intersects γ_A at an arc containing a of angle $O(\sqrt{\epsilon})$, (ii) intersects ℓ_B at precisely the segment from b to b', and (iii) contains $(1/\sqrt{2}, 1/\sqrt{2}, 1)$.*

Proof. Say $a = (x, y, 0)$, $b = (0, 0, z-h)$, $b' = (0, 0, z+h)$. Consider the unit ball K centered at $c = ((1-h^2)x, (1-h^2)y, z)$. Then (ii) is self-evident and (iii) is straightforward to check. For (i), note that a lies in K since $\|a - c\|^2 = h^2 + z^2 \le \epsilon^2 + (1-\epsilon)^2 < 1$. On the other hand, if a point $p \in \gamma_A$ lies in the unit ball, then letting $a' = ((1-h^2)x, (1-h^2)y, 0)$, we have $\|p - c\|^2 = \|p - a'\|^2 + z^2 \le 1$, implying $\|p - a\| \le \|p - a'\| + \|a' - a\| \le \sqrt{1 - z^2} + h = O(\sqrt{\epsilon})$. □

(C8) is similar to (C1); we place the elements in A on the line segment $\ell_A = \{(t, t, 0) : t \in (0, 1)\}$ and the elements in B on the line segment $\ell_B = \{(0, 3-t, t) : t \in (0, 1)\}$. For any $(a, a, 0) \in \ell_A$ and $(0, 3-b, b) \in \ell_B$, the cube $[-3+b+2a, a] \times [a, 3-b] \times [-3+a+2b, b]$ is tangent to ℓ_A at $(a, a, 0)$, is tangent to ℓ_B at $(0, 3-b, b)$, and contains $(0, 1, 0)$. Size-3 sets $\{a_j, x_t, y_t\}$ can be encoded by taking a cube with tangents at a_j and x_t, expanding it slightly, and making x_t and y_t sufficiently close.

(C9) follows from (C7) by the standard lifting transformation [5].

For (H1), we map each element a_i to a thin vertical slab. For each $1 \le t \le m$, we map w_t, x_t, y_t, z_t to a cluster of four thin horizontal slabs as in Figure 3. Each set in $\mathcal{S}$ can be encoded as a point in the arrangement of these slabs.

(H2) is similar; see Figure 3.

(H3) follows from (C7) by duality.

(H4) follows from (C9) by duality.

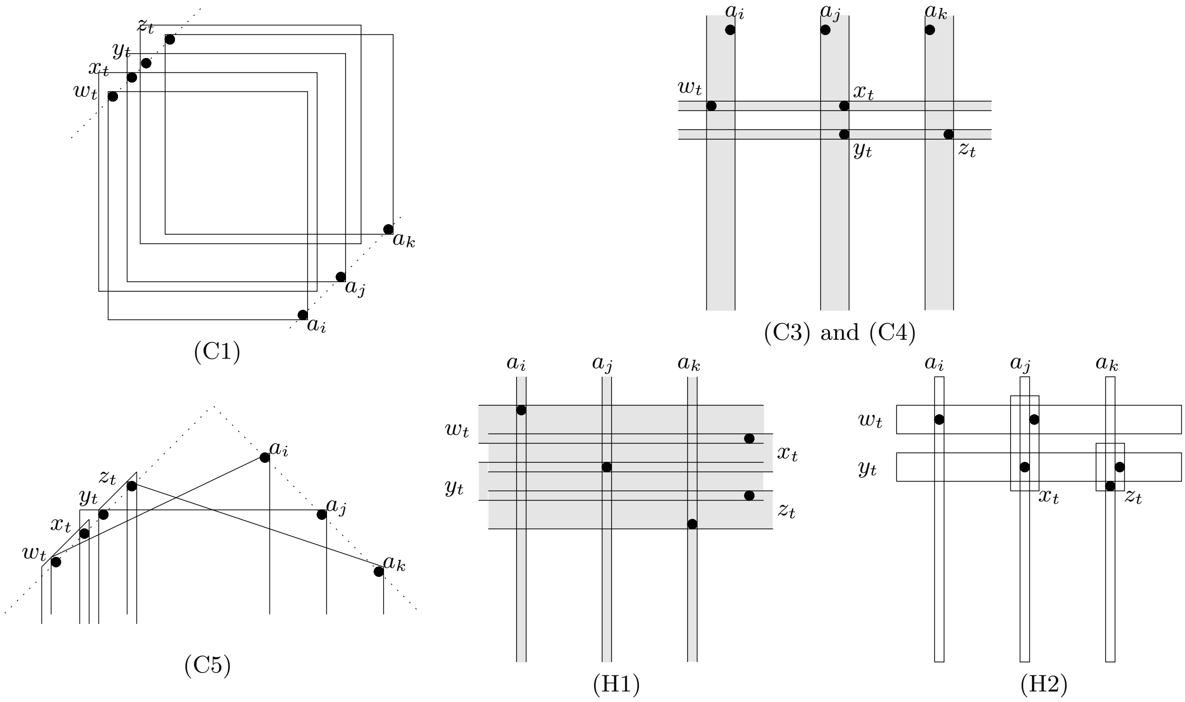

Figure 1: APX-hardness proofs of geometric set cover problems.

4 Algorithm for Weighted Covering by Downward Shadows of 2-Intersecting x-Monotone Curves

Here, we prove Theorem 4 by giving a polynomial-time dynamic programming algorithm for the weighted cover of a finite set of points $X \subseteq \mathbb{R}^2$ by a set $\mathcal{S}$ of downward shadows of 2-intersecting x-monotone curves $C_1, \ldots, C_m$. For $1 \leq i \leq m$, define the region $S_i \in \mathcal{S}$ to be the downward shadow of the curve C_i and let it have positive cost w_i. Define $n = |X|$.

We shall assume that each C_i is the graph of a smooth univariate function with domain $[-\infty, \infty]$, that all intersections are transverse (no pair of curves intersect tangentially), and that no points in X lie on any curve C_i. It is not difficult to get around these assumptions, but we retain them to simplify our explanation.

We shall slightly abuse notation by writing $C_i(x)$ for the unique $y \in \mathbb{R}$ such that (x, y) lies on the curve C_i. We say curve C_i is *wider* than curve C_j (written $C_i \succ C_j$) whenever $C_i(x) > C_j(x)$ for all sufficiently large x. We may also write $S_i \succ S_j$ whenever $C_i \succ C_j$. We note that $\succ$ is a total ordering and thus we can order all curves by width, so we assume without loss of generality that $C_i \succ C_j$ whenever $i > j$. The width-based ordering of curves is useful because of the following key observation:

Proposition 8 *If $C_i \succ C_j$, then $S_j \setminus S_i$ is connected.*

Proof. This is clearly true if C_i and C_j intersect once or less. If C_i intersects C_j twice—say, at (x_1, y_1) and (x_2, y_2) with $x_2 > x_1$—then since all intersections are transverse, the area above C_i but below C_j can only be disconnected if $C_j(x) > C_i(x)$ for $x < x_1$ and $x > x_2$, implying $C_j \succ C_i$. □

For all $1 \leq i \leq m$ and all intervals $[a, b]$, define $X[a, b]$ to be all points in X with x-coordinate in $[a, b]$, and define $X[a, b, i]$ to be $X[a, b] \setminus S_i$. Define $\mathcal{S}_{<i}$ to be the set $\{S_1, \ldots, S_{i-1}\}$ of all regions of width less than S_i. Let $M[a, b, i]$ denote the minimum cost of a solution to the weighted set cover problem on the set system $(X[a, b, i], \mathcal{S}_{<i})$ (with weights inherited from the original problem). If such a covering does not exist, $M[a, b, i] = \infty$. For notational simplicity, we assume that C_m, the widest curve, contains no points in its downward shadow (that is, $X \cap S_m$ is empty). Our goal is then to determine $M[-\infty, \infty, m]$ via dynamic programming; the key structural result we need is the following:

Proposition 9 *If $X[a, b, i]$ is non-empty, then*

$$M[a, b, i] = \min \Big\{ \min_{c \in (a,b)} \{M[a, c, i] + M[c, b, i]\}, \min_{j<i} \{M[a, b, j] + w_j\} \Big\}.$$

Proof. Clearly $M[a, b, i] \leq M[a, c, i] + M[c, b, i]$ for all $c \in (a, b)$. Also, for $j < i$, $M[a, b, j] + w_j$ is the cost of purchasing S_j and then covering the remaining points in $X[a, b]$ using regions less wide than S_j (and hence less wide than S_i). Thus $M[a, b, j] + w_j$ is a cost of a feasible solution to $(X[a,b,i], \mathcal{S}_{<i})$ and hence is at least $M[a,b,i]$. It follows that $M[a,b,i]$ is bounded above by the right hand side.

To show that $M[a,b,i]$ is bounded below by the right hand side, we let $\mathcal{Z} \subseteq \mathcal{S}_{<i}$ be a feasible set cover for $(X[a,b,i], \mathcal{S}_{<i})$. We consider two cases:

Case 1: There is some $c \in (a,b)$ such that $(c, C_i(c))$ is not covered by $\mathcal{Z}$. Let $\mathcal{Z}_{<c}$ be the set of all regions in $\mathcal{Z}$ containing a point in $X[a,c,i]$, and let $\mathcal{Z}_{>c}$ be the set of all regions in $\mathcal{Z}$ containing a point in $X[c,b,i]$. Let $Z \in \mathcal{Z}$. Since $Z \prec S_i$, by Proposition 8, $Z \setminus S_i$ is connected and thus cannot contain points both in $X[a,c,i]$ and $X[c,b,i]$. Hence $\mathcal{Z}_{<c} \cap \mathcal{Z}_{>c} = \emptyset$ and thus the cost of $\mathcal{Z}$ is at least $M[a,c,i] + M[c,b,i]$.

Case 2: For all $c \in (a,b)$, the point $(c, C_i(c))$ is covered by $\mathcal{Z}$. Then $\mathcal{Z}$ covers $X[a,b,i] \cup S_i$ and hence covers all points in $X[a,b]$. Let C_j be the widest curve in $\mathcal{Z}$, noting that $j < i$. Then the cost of $\mathcal{Z}$ is at least $w_j + M[a,b,j]$ since $\mathcal{Z} \setminus S_j$ must cover all points in $X[a,b,j]$.

It follows that $\mathcal{Z}$ must cost as much as either $\min_{c\in(a,b)}\{M[a,c,i] + M[c,b,i]\}$ or $\min_{j<i}\{M[a,b,j] + w_j\}$, and the result follows. □

Proposition 9 immediately implies the existence of a polynomial-time dynamic programming algorithm to compute $M[-\infty, \infty, m]$ and return a cover having that cost. We note that there are at most $n+1$ combinatorially relevant values of a and b when computing optimal costs $M[a,b,i]$ for subproblems, so there are $O(mn^2)$ distinct values of $M[a,b,i]$ to compute. Recursively computing $M[a,b,i]$ requires $O(m+n)$ table lookups, so the total runtime of our algorithm is $O(mn^2(m+n))$, assuming a representation allowing primitive operations in $O(1)$ time.

References

[1] B. Aronov, E. Ezra, and M. Sharir. Small-size ϵ-nets for axis-parallel rectangles and boxes. In *ACM Symposium on Theory of Computing* (2009), 639648.

[2] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Comp. Sci.* 237 (2000) 123-134.

[3] C. Ambühl, T. Erlebach, M. Mihal'ák, and M. Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *APPROX and RANDOM* (2006) 3-14.

[4] P. K. Agarwal, E. Nevo, J. Pach, R. Pinchasi, M. Sharir, and S. Smorodinsky. Lenses in arrangements of pseudo-circles and their applications. *J. ACM* 51(2) (2004), 139-186.

[5] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications (Third edition)*. Springer-Verlag, Heidelberg, (2008).

[6] P. Berman and B. DasGupta. Complexities of efficient solutions of rectilinear polygon cover problems. *Algorithmica* 17(4) (1997), 331356.

[7] H. Brönnimann and O. Devillers The union of unit balls has quadratic complexity, even if they all contain the origin. arXiv:cs/9907025v1 [cs.CG] (1999).

[8] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.* 14 (1995), 263279.

[9] N. Bansal and K. Pruhs. The geometry of scheduling. *IEEE 51st Annual Symposium on Foundations of Computer Science* (2010), 407-414.

[10] D. Chakrabarty, E. Grant, and J. Koenemann. On column-restricted and priority covering integer programs. In *Integer Programming and Combinatorial Optimization* (2010) 355-368.

[11] Y. Cheng, S.S. Iyengar, and R.L. Kashyap. A new method for image compression using irreducible covers of maximal rectangles. *IEEE Trans. Software Engrg.* 14 (1988), 651658.

[12] K. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete Comput. Geom.* 37 (2007), 4358.

[13] G. Even, D. Rawitz, and S. Shahar. Hitting sets when the VC-dimension is small. *Information Processing Letters* 95(2) (2005), 358362.

[14] T. Erlebach and E. J. van Leeuwen PTAS for weighted set cover on unit squares. In *APPROX and RANDOM* (2010), 166-177.

[15] A. Hegedüs. Algorithms for covering polygons with rectangles. *Comput. Aided Geom. Design* 14 (1982), 257260.

[16] S. Har-Peled. Being Fat and Friendly is Not Enough. arXiv:0908.2369v1 [cs.CG] (2009).

[17] S. Har-Peled and M. Lee. Weighted geometric set cover problems revisited. Unpublished manuscript, (2008).

[18] D.S. Hochbaum and W. Maass. Fast approximation algorithms for a nonconvex covering problem. *J. Algorithms* 8(3) (1987), 305323.

[19] V.S.A. Kumar and H. Ramesh. Covering rectilinear polygons with axis-parallel rectangles. *SIAM J. Comput.* 32(6) (2003), 1509-1541.

[20] N.H. Mustafa and S Ray. Improved results on geometric hitting set problems. *Discrete Comput. Geom.* 44(4) (2010), 883895.

[21] J. Pach, G. Tardos. Tight lower bounds for the size of epsilon-nets. In *Proc. 27th ACM Sympos. Comput. Geom.* (2011), to appear.

[22] C.H. Papadimitriou, M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Systems Sci.* 43 (1991), 425-440.

[23] K. Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *ACM Symposium on Theory of Computing* (2010).

[24] E. J. van Leeuwen. *Optimization and Approximation on Systems of Geometric Objects*. PhD thesis, Universiteit van Amsterdam, (2009).

Appendix

This will not be in the printed proceedings.