

Optimally Decomposing Coverings with Translates of a Convex Polygon

Matt Gibson · Kasturi Varadarajan

Received: 23 March 2010 / Revised: 9 March 2011 / Accepted: 6 April 2011 /
Published online: 29 June 2011
© Springer Science+Business Media, LLC 2011

Abstract We show that any k -fold covering using translates of an arbitrary convex polygon can be decomposed into $\Omega(k)$ covers. Such a decomposition can be computed using an efficient (polynomial-time) algorithm.

Keywords Cover decomposability · Geometric set cover

1 Introduction

Let us call an object (set) P in the plane *plane-cover-decomposable* if there exists a constant $c > 0$ (which may depend on P) such that any collection of translates of P , with the property that every point in the plane has c or more translates covering it, can be partitioned into two covers of the plane. Pach conjectured in the 1980s that every convex object is cover-decomposable [4, 5], and this remains open. In this article, we focus on a finite version of this definition. We say that a collection of objects is a *c-fold covering* of a point set X if every point in X is contained within at least c of the objects. We say that the object P is *cover-decomposable* if there exists a constant $c > 0$ such that any *finite* collection of translates of P that is a c -fold covering of some point set X can be partitioned into two sub-collections, so that each sub-collection covers (every point in) X . That is, if whenever a collection of translates of P form a c -fold covering of a point set X the translates can be partitioned into two 1-fold

A preliminary version of the results in this article appears in *Gibson and Varadarajan, Decomposing Coverings and the Planar Sensor Cover Problem, Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS), 2009*.

M. Gibson (✉) · K. Varadarajan
Department of Computer Science, The University of Iowa, Iowa City, IA 52242-1419, USA
e-mail: mrgibson@cs.uiowa.edu

K. Varadarajan
e-mail: kvaradar@cs.uiowa.edu

covers of X , then we say that P is cover-decomposable. We note that if P is open and is cover-decomposable, then it is also plane-cover-decomposable.

In the 1980s, Mani and Pach [3] showed in an unpublished manuscript that the unit disk is cover-decomposable (with the constant c being 33). Also in the 1980s, Pach [4] showed that any centrally-symmetric convex polygon is cover-decomposable. Tardos and Tóth [11] showed somewhat more recently that any triangle is cover-decomposable. Finally, a very recent result due to Pálvölgyi and Tóth [8] shows that any convex polygon is cover-decomposable. The constant c in the results of [4] and [8] depends on the convex polygon, in particular the number of its sides, and that is why these results say nothing about the original conjecture of Pach on the cover-decomposability of an arbitrary convex set. On the other hand, convexity is a necessary restriction—Pach, Tardos, and Tóth [6] and Pálvölgyi [7] have shown that there exist non-convex polygons that are not cover-decomposable [7].

Motivated partly by questions in scheduling sensors [2], an extension of the cover-decomposability question has recently attracted a lot of attention: Given a collection of translates of P and any integer k , partition the collection into as many sub-collections as possible so that each sub-collection covers every point covered by k or more of the original translates. That is, we would like to be able to decompose a k -fold cover into as many disjoint covers as possible. While the original results on cover-decomposability do yield non-trivial bounds for this question, these are usually far from optimal. For instance, the proof of the cover-decomposability of a triangle given by Tardos and Tóth [11] implies that any k -fold cover with translates of a triangle can be partitioned into $\Omega(\log k)$ covers.

Initiating this line of work, Pach and Tóth [5] showed that any k -fold cover with a *centrally-symmetric* convex polygon P can be decomposed into $\Omega(\sqrt{k})$ covers, where the constant as before depends on P . Aloupis et al. [1] improved this result and obtained an optimal bound for centrally-symmetric polygons, showing that one can obtain $\Omega(k)$ covers. Both of these results have corresponding efficient algorithms that compute the desired decompositions.

The problem of decomposing multiple coverings seems to be harder if instead of a convex polygon we have a unit disk. Pandit, Pemmaraju and Varadarajan [9] consider a special case where the point set that needs to be covered is the same as the centers of the given disks. For this version of the problem, better known as the *domatic partition* problem for *unit disk graphs* [10], they show that it is possible to compute $\Omega(k)$ disjoint covers.

1.1 Our Contribution

Our main contribution here is an optimal result for translates of an arbitrary convex polygon:

Theorem 1 *For any convex polygon P in the plane, there exists a constant $\alpha \geq 1$ so that for any $k \geq 1$ and any finite collection of translates of P , we can partition the collection into k/α sub-collections, each of which covers any point in the plane that is covered by k or more translates in the original collection.*

Our techniques build upon the recent work of Aloupis et al. [1] for centrally-symmetric convex polygons. Suppose a convex polygon has its centroid at the origin and contains the point $p = (p_x, p_y) \in \mathbb{R}^2$. The polygon is *centrally-symmetric* if it also contains the point $-p = (-p_x, -p_y)$. A key idea of theirs is to focus on the level curves corresponding to the wedges at the vertices of P . The interaction of these level curves can be complex, but they show that it is sufficient to work within a region where the interaction is much more controlled. It is only for centrally-symmetric convex polygons that they establish such nice properties of the interaction. The notion of level curves is also central to our work, but the main point of departure is the simplicity of the new way in which we handle the level curve interactions. As we show, this new approach turns out to be flexible enough to handle arbitrary convex polygons.

We begin our proof of Theorem 1 by establishing it in Sect. 3 for centrally-symmetric polygons. Aloupis et al. [1] have already shown such a result, but we do this in a way that illustrates our new approach and serves as a nice foundation for describing our algorithm for general convex polygons. In Sect. 4, we establish our main result, by extending our approach to arbitrary convex polygons, highlighting the one crucial ingredient needed for this generalization. But first, we must recall in Sect. 2 the toolkit developed in previous work on the problem of decomposing multiple coverings.

2 Preliminaries

It is convenient to prove Theorem 1 in its dual form as done in [1, 11]. Suppose we are given a closed, centrally-symmetric convex polygon P . Fix O , the centroid of P , as the origin in the plane. For a planar set T and a point x in the plane, let $T(x)$ denote the translate of T with centroid x . Let \bar{P} be the reflection through O of the polygon P . For points p and x in the plane, $p \in P(x)$ if and only if $x \in \bar{P}(p)$.

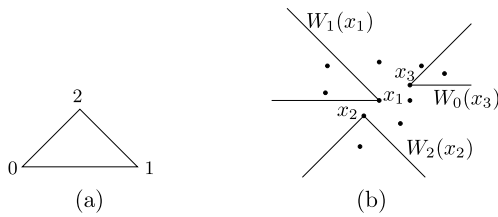
Because of this transformation, it is sufficient for us to show that there exists a constant $\alpha \geq 1$ so that for any $k \geq 1$ and any collection Q of points in the plane, it is possible to assign each point in Q a color from $\{1, 2, \dots, \frac{k}{\alpha}\}$, so that any translate of \bar{P} with $|\bar{P} \cap Q| \geq k$ contains a point colored i , for each $1 \leq i \leq \frac{k}{\alpha}$. Abusing notation, we will refer to this reflected polygon as P for the remainder of the paper.

Polygons to Wedges Let the vertices of P be $p_0, p_1, p_2, \dots, p_{\mu-1}$ in counterclockwise order. Addition and subtraction of indices of these vertices will be taken modulo μ throughout the paper. The set of indices between index i and index j in counterclockwise order are denoted $[i, j]$. We now transform the problem further, so that instead of dealing with translates of P , we can deal with translates of the μ wedges corresponding to the vertices of P [1, 5, 11].

Let c be equal to half the minimum distance between two points on non-consecutive edges of P . We lay a square grid of side c on the plane; any translate of P intersects $\beta \in O(1)$ grid cells, and each grid cell intersects at most two sides of a translate; moreover, if a grid cell does intersect two sides of a translate, then these sides must be adjacent in P .

Fig. 1 An illustration for the wedges of a polygon.

(a) Suppose this triangle is our polygon with vertices indexed accordingly. (b) A 0-wedge, 1-wedge, and 2-wedge with respect to the polygon



For a subset (region) R of the plane and for a finite subset X of points, denote $\text{load}_X(R)$ to be the number of points in X that lie in R . We call this value the load of region R with respect to X . Since each translate $P(u)$ intersects at most β grid cells, $P(u)$ must contain load at least k/β within some grid cell if its load with respect to Q is at least k . We can therefore make the points of Q within such a grid cell “responsible” for $P(u)$.

Since each grid cell intersects at most two edges of $P(u)$, it must be that the intersection of a grid cell with $P(u)$ is the same as the intersection of the grid cell with a wedge whose bounding halflines are parallel to two adjacent edges of $P(u)$. If one halfline of the wedge is parallel to the edge $p_{i-1}p_i$ of P and the other is parallel with $p_i p_{i+1}$ of P , then we call the wedge an i -wedge. For a point q in the plane, we denote $W_i(q)$ to be the i -wedge with apex q . See Fig. 1 for an illustration.

Because of these observations, Theorem 1 is established by applying the following theorem to the points Y within each grid cell G .

Theorem 2 *There exists a constant $\alpha' \geq 1$ so that for any $k \geq 1$ and any collection Y of points in the plane, it is possible to assign each point in Y a color from $\{1, 2, \dots, \frac{k}{\alpha'}\}$, so that any i -wedge (corresponding to an i -wedge of P) that contains k or more points from Y contains a point colored j , for each $1 \leq j \leq \frac{k}{\alpha'}$.*

We prove Theorem 2 for wedges of a centrally-symmetric convex polygon in Sect. 3, and we prove the same theorem for wedges of a general convex polygon in Sect. 4. We assume that the point set Y is in general position—a line parallel to a side of P contains at most one point in Y . It is straightforward to perturb the input to the original problem so that this assumption holds for Y .

Level Curves We will now define a boundary for an $i \in \{0, 1, \dots, \mu - 1\}$ and positive integer r . This boundary has the property that any i -wedge placed on or “inside” the boundary has load at least r with respect to Y , and any i -wedge placed “outside” the boundary has load less than r . That is, the number of points in $W_i(x) \cap Y$ for any x inside the boundary or on the boundary is at least r and is less than r for any x outside the boundary. See Fig. 2 for an illustration. This boundary is called a *level curve* [1] and extends the definition of *boundary points* [4, 5]. Let \mathcal{W}_i^j be the set of apices of all i -wedges W such that $\text{load}_Y(W) = j$. For each $i = 0, 1, \dots, \mu - 1$, let the level curve $\mathcal{C}_i(r)$ be the boundary of the region $\mathcal{W}_i^{\geq r} = \bigcup_{j \geq r} \mathcal{W}_i^j$ for each $i = 0, 1, \dots, \mu - 1$.

Note that $\mathcal{C}_i(r)$ is a monotone staircase polygonal path with edges that are parallel to the edges of an i -wedge; see Fig. 2. We have the following observations:

Fig. 2 An example of a level curve $C_i(r)$ for $r = 2$. Note that any i -wedge with apex on $C_i(2)$ (e.g., the dotted wedge) contains load at least 2

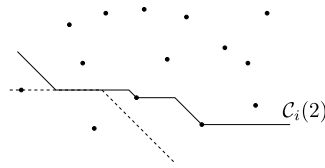
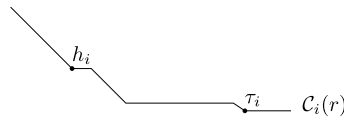


Fig. 3 Level curve $C_i(r)$ with h_i and τ_i denoted



Observation 3 For any $x \in C_i(r)$, $r \leq \text{load}_Y(W_i(x)) \leq r + 1$.

Observation 4 Any i -wedge W such that $\text{load}_Y(W) \geq r$ contains an i -wedge whose apex belongs to $C_i(r)$.

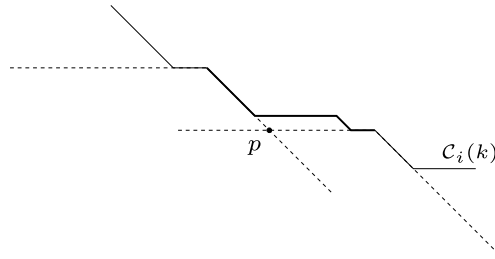
Observe that one of the two extreme edges of the level curve $C_i(r)$ is a semi-infinite ray parallel to edge $p_{i-1}p_i$. Let h_i denote the origin of this ray. We call h_i the *head* of $C_i(r)$. Note that for all points y on the ray, $W_i(y) \cap Y = W_i(h_i) \cap Y$. The other extreme edge of $C_i(r)$ is parallel to edge $p_i p_{i+1}$. Let τ_i denote the origin of this ray. We call τ_i the *tail* of $C_i(r)$. Note that for all points y on the ray, $W_i(y) \cap Y = W_i(\tau_i) \cap Y$; see Fig. 3.

Simple Algorithm for One Level Curve Observation 4 implies that it is sufficient to prove Theorem 2 for the i -wedges with apex on $C_i(k)$, for each $0 \leq i \leq \mu - 1$. In order to do this, we will need a procedure that takes as input one level curve $C_i(k)$, a positive integer t and a subset $Q \subseteq Y$. The input to the procedure has the guarantee that for any i -wedge W with apex on $C_i(k)$, we have $|W \cap Q| \geq 2t$. The goal is to output a partial coloring of the points of Q with colors $\{1, 2, \dots, t\}$ so that any i -wedge W with apex on $C_i(k)$ (a) contains a point colored j , for $1 \leq j \leq t$, and (b) contains at most $2t$ colored points.

It is known [1] that such a procedure exists. The reason is that for any $q \in Q$, the set $I(q) = \{u \in C_i(k) \mid q \in W_i(u)\}$ of apexes of i -wedges containing q is an “interval” of $C_i(k)$. See Fig. 4 for an illustration. We consider these intervals in an order such that if interval I properly contains interval I' , then we consider I before I' . Considering intervals in such an order, we add an interval into our working set if it covers a point of $C_i(k)$ that is not covered by previous intervals in the working set. Notice that after all intervals have been considered, the working set forms a cover of $C_i(k)$. Now, we repeatedly throw out intervals from the working set that are redundant—an interval is redundant if throwing it out of the current working set does not affect coverage of $C_i(k)$.

The final non-redundant working set covers $C_i(k)$, but also has no more than two intervals covering any point of $C_i(k)$. We give the color 1 to the points in Q that give rise to the intervals in our working set. We repeat this process $t - 1$ more times after removing the colored points from Q . It is easy to verify that the overall procedure, which we call `computeCover`(i, Q, t), successfully achieves properties (a) and (b).

Fig. 4 An example of an interval $I(p)$ (in bold). Note that the i -wedges with apex on $C_i(k)$ that contain p are the dotted wedges and all wedges with apex “in between” the apices of the dotted wedges



Algorithm 1

- 1: $Y' \leftarrow Y$
- 2: **for** $i \leftarrow 0$ to $\mu - 1$ **do**
- 3: $L \leftarrow \min\{\text{load}_{Y'}(W_z(x)) : x \in C_z(k) \text{ and } z = i, i + 1, \dots, \mu - 1\}$
- 4: $X_i \leftarrow Y' \cap \mathcal{W}_i^{\leq k}$
- 5: Run `computeCover`($i, X_i, \frac{L}{64\mu}$). Let $Y_i \subseteq X_i$ be the points assigned a color during this call.
- 6: Let Y' denote the uncolored points (i.e., $Y' \leftarrow Y' \setminus Y_i$).

We have the following observation whose second claim easily follows from the manner in which we pick our non-redundant working set.

Observation 5 *The partial cover computed by `computeCover`(i, Q, t) has the property that any i -wedge with apex on $C_i(k)$ has at most $2t$ colored points. Furthermore, if q and q' are points in Q such that $q \in W_i(q')$ (that is, $I(q)$ properly contains $I(q')$), then q' is colored only if q is colored.*

3 Our Algorithm for Centrally-Symmetric Polygons

This section is devoted to a proof of Theorem 2 for the case where P , and hence P , is a centrally-symmetric convex polygon. Such a proof is, of course, the main result of Aloupis et al. [1]. However, this is a good context to introduce our proof method which is slightly different. Moreover, portions of the proof will be invoked for the case of general convex polygons. Referring to Theorem 2, we may assume that $|Y| \geq k \geq 1000\mu \cdot (5\mu)^\mu$.

The algorithm that achieves the coloring claimed in Theorem 2 is Algorithm 1. Algorithm 1 calls `computeCover`(i, X_i, t) for each $0 \leq i \leq \mu - 1$. The set X_i in the i th iteration consists of all the points in $Y \cap \mathcal{W}_i^{\leq k}$ not colored in iterations $0, 1, \dots, i - 1$. At the beginning of the i th iteration, let L denote, as in the algorithm, the smallest number of uncolored points in a j -wedge with apex on $C_j(k)$, for $i \leq j \leq \mu - 1$. The parameter t is chosen to be $\frac{L}{64\mu}$. After the call to `computeCover`($i, X_i, \frac{L}{64\mu}$), any i -wedge with apex on $C_i(k)$ contains points colored $1, 2, \dots, L/64\mu$. This is $\Omega(k)$ colors provided $L \in \Omega(k)$. This is established in the remainder of the section.

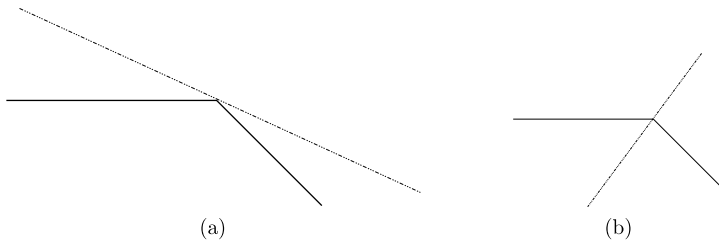


Fig. 5 An illustration for the definition of tangent. (a) This line is tangent to the wedge. (b) This line is not tangent to the wedge

We will show that L , which is at least k before the 0th iteration, drops by at most a constant factor (i.e., $O(\mu)$) with each iteration. More specifically, we will show that L drops by at most a constant factor during iteration i for all $j > i$.

We use the following terminology for iteration i : for two distinct points q and q' , if $W_i(q) \subseteq W_i(q')$, we say that q dominates q' . Notice that if q and q' are both uncolored before iteration i , then q' is colored in iteration i only if q is already colored. (This is Observation 5.) For the rest of this section, let X_i denote set of candidate points that are eligible to be colored in iteration i (as constructed in the algorithm), and let Y_i denote the points that are actually colored in iteration i .

The analysis will rely heavily upon the following observation, which follows directly from Observation 5 and the fact that in Step 5 of the algorithm, we invoke $\text{computeCover}(i, X_i, t)$ with $t = \frac{L}{64\mu}$.

Observation 6 For any $z \in \mathcal{C}_i(k)$, we have that $\text{load}_{Y_i}(W_i(z)) \leq \frac{L}{32\mu}$.

A line through a vertex of P is *tangent* to P if it intersects P only at the vertex. Similarly, a line through the apex of a wedge is *tangent* to the wedge if it intersects the wedge only at the apex; see Fig. 5. We will consider the following two cases:

- p_i and p_j are antipodal vertices of P , that is, there are parallel lines through p_i and p_j such that both of the lines are tangent to P , and P is between the two parallel lines.
- p_i and p_j are not antipodal vertices of P .

Suppose we have a j -wedge W . If p_i and p_j are not antipodal vertices of P , then we say that W is *nonantipodal with respect to an i -wedge*. Now suppose that W is a wedge of any type, i.e., not necessarily a type corresponding to a vertex of P . Suppose the apex of W is the point x , and consider the i -wedge $W_i(x)$; see Fig. 6. We say that W is *subantipodal with respect to an i -wedge* if (1) every line that is tangent to $W_i(x)$ is also tangent to W , and (2) W and $W_i(x)$ are on “opposite sides” of each of these tangent lines; see Fig. 7.

We will argue that each wedge W (of type other than i) will have at most a constant factor of its uncolored points assigned a color during iteration i of the algorithm. Lemma 8 handles the case when W is nonantipodal with respect to an i -wedge, and Lemma 9 handles the case when W is subantipodal with respect to an i -wedge. The reader will note that the lemmas are written in a form that is more general than needed

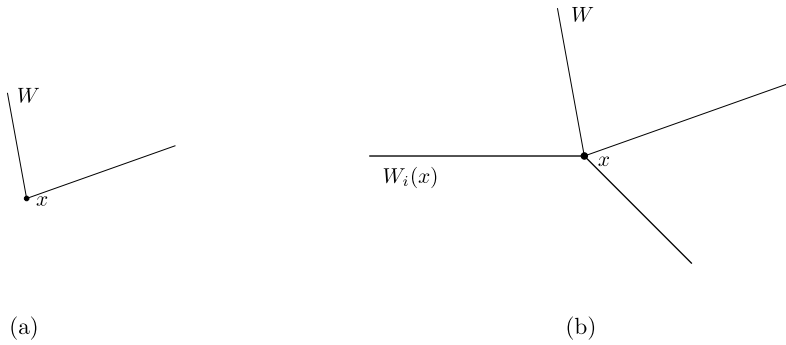


Fig. 6 An illustration of W and $W_i(x)$. **(a)** The wedge W with apex at x . **(b)** $W_i(x)$ and W both have their apex at x

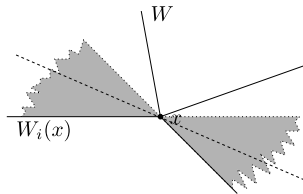


Fig. 7 All of the lines tangent to $W_i(x)$ must lie in the shaded region (e.g., the dotted line). Every such line is also tangent to W , and the wedges are on opposite sides of every such line. By definition, W is subantipodal with respect to an i -wedge

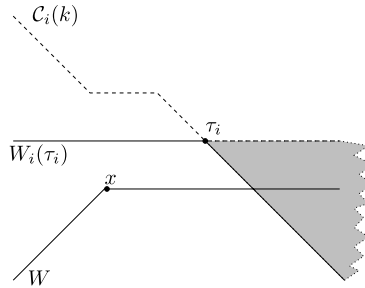
for the centrally-symmetric case—first, the statement of the lemmas only assumes that an i -wedge with apex on $C_i(k)$ has load at least $L/2$ from points in X_i ; second, the notion of subantipodal for which Lemma 9 is stated is more general than is needed for the centrally-symmetric case. We use this more general form for these lemmas because we are going to need it later in the proof for polygons that are not centrally-symmetric.

Note that every wedge of a centrally-symmetric convex polygon of type other than i is either nonantipodal or subantipodal with respect to i . Thus, assuming Lemmas 8 and 9 are true, the following theorem easily follows. Note that this theorem is simply a restatement of Theorem 2 for the centrally-symmetric case.

Theorem 7 *Let P be any centrally-symmetric convex polygon with μ vertices. Let $Y \subset \mathbb{R}^2$ be any set of points so that $|Y| \geq k \geq 1000\mu \cdot (5\mu)^\mu$, where k is a parameter. Algorithm 1 colors the points in Y with $\Omega(k)$ colors in a way such that for any j -wedge W corresponding to P such that $|W \cap Y| \geq k$, W contains a point of each color.*

Proof In iteration i , we color points in X_i so that each i -wedge contains points of $t = \frac{L}{64\mu}$ different colors, where L is as defined in the algorithm. This gives $\Omega(k)$ different colors as long as L (which is at least k prior to iteration 0 of the algorithm) is $\Omega(k)$ prior to iteration i of the algorithm. Lemmas 8 and 9 ensure this by showing

Fig. 8 Illustration for Lemma 8. Note that there cannot be a point in the shaded region by definition of the tail τ_i



that L falls by at most a constant factor in each iteration of the algorithm for all j -wedges such that $j > i$.

Let W be any wedge of type j such that $j > i$. Note that if $\text{load}_{X_i}(W) < \frac{L}{6}$ then, clearly, W will contain at least $L - \frac{L}{6}$ uncolored points after iteration i (only points from X_i will be assigned a color). Therefore, we only need to handle the case when $\text{load}_{X_i}(W) \geq \frac{L}{6}$. Also note that in the case of centrally-symmetric convex polygons, every j -wedge with $j \neq i$ is either nonantipodal with respect to an i -wedge or is subantipodal with respect to an i -wedge. If W is nonantipodal with respect to an i -wedge then we can invoke Lemma 8. If W is subantipodal with respect to an i -wedge then we can invoke Lemma 9. In both cases, the number of uncolored points in W falls by a factor of at most 5μ in iteration i . Therefore, when we reach iteration j of the algorithm, W will still contain $\Omega(k)$ points which have not yet received a color. \square

The following lemma handles the case for wedges which are nonantipodal with respect to an i -wedge.

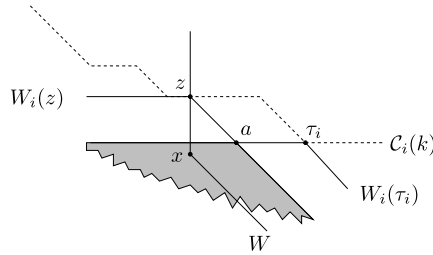
Lemma 8 *Suppose that W is a wedge that is nonantipodal with respect to an i -wedge and $\text{load}_{X_i}(W) \geq \frac{L}{6}$. Further suppose that before the call to $\text{computeCover}(i, X_i, \frac{L}{64\mu})$ in iteration i , all i -wedges with apex on $C_i(k)$ have load at least $\frac{L}{2}$ from points in X_i . After the call to $\text{computeCover}(i, X_i, \frac{L}{64\mu})$ in the i th iteration of the algorithm, W has load at least $\frac{L}{5\mu}$ from points in Y' . (Note that Y' always denotes the uncolored points in the algorithm.)*

Proof Let x denote the apex of W . Suppose that $W \cap C_i(k) = \emptyset$. For this to be the case, one of the halflines of W must be parallel with a halfline of an i -wedge. We will focus on the case when W has a halfline parallel to the side $p_i p_{i+1}$ as the other case is symmetric; see Fig. 8. In this case, we have that $W \cap X_i \subseteq W_i(\tau_i)$ by the definition of the tail τ_i . We thus have $\text{load}_{Y_i}(W) \leq \text{load}_{Y_i}(W_i(\tau_i)) \leq \frac{L}{32\mu}$ where the last inequality comes from Observation 6. Therefore, the load of uncolored points in W after iteration i is at least $\frac{L}{6} - \frac{L}{32\mu} > \frac{L}{5\mu}$.

So let us assume that $W \cap C_i(k) \neq \emptyset$. Recall that since W is nonantipodal with respect to an i -wedge, it is of type j where j corresponds to a vertex p_j of P . There are two cases to consider:

1. We encounter p_j after p_i and before the vertices antipodal to p_i when walking counter-clockwise around P , and

Fig. 9 Illustration for the nonantipodal case



2. We encounter p_j after the vertices antipodal to p_i and before p_i .

We will focus on the first case, since the other is symmetric. Let z be the intersection point of the boundary of W and $C_i(k)$ (to be precise, let z be the last point on $W \cap C_i(k)$ as one walks “clockwise” around the boundary of W). If W does not contain in its interior the tail τ_i of the level curve $C_i(k)$, then $W \cap Y_i \subseteq W_i(z) \cap Y_i$, and so $\text{load}_{Y_i}(W) \leq \text{load}_{Y_i}(W_i(z)) \leq \frac{L}{32\mu}$. It follows that the load of uncolored points in W after iteration i is at least

$$\frac{L}{6} - \frac{L}{32\mu} > \frac{L}{5\mu}.$$

Let us therefore assume that W does contain in its interior the tail τ_i of $C_i(k)$; see Fig. 9. Let a denote the point where the boundaries of the wedges $W_i(z)$ and $W_i(\tau_i)$ intersect. If $\text{load}_{X_i}(W_i(a)) \geq \frac{L}{8\mu}$, then since $\text{load}_{Y_i}(W_i(a)) \leq \text{load}_{Y_i}(W_i(\tau_i)) \leq \frac{L}{32\mu}$, there are uncolored points in $W_i(a)$ after iteration i . Since any point in $W_i(a)$ dominates points in $W \cap Y_i$ that are not contained in $W_i(z) \cup W_i(\tau_i)$, we conclude that $W \cap Y_i \subseteq (W_i(z) \cup W_i(\tau_i)) \cap Y_i$. Thus,

$$\text{load}_{Y_i}(W) \leq \text{load}_{Y_i}(W_i(z)) + \text{load}_{Y_i}(W_i(\tau_i)) \leq \frac{L}{16\mu}.$$

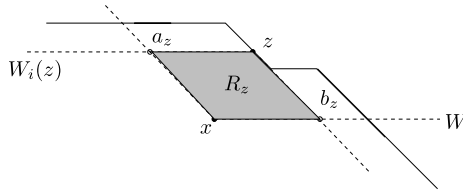
Therefore, there must be at least $\frac{L}{6} - \frac{L}{16\mu} > \frac{L}{5\mu}$ uncolored points left in W .

Let us therefore consider the case where $\text{load}_{X_i}(W_i(a)) < \frac{L}{8\mu}$. This means that $\text{load}_{X_i}(W_i(\tau_i) \setminus W_i(a)) > \frac{L}{2} - \frac{L}{8\mu} > \frac{L}{3}$. Again, $\text{load}_{Y_i}(W_i(\tau_i) \setminus W_i(a)) \leq \text{load}_{Y_i}(W_i(\tau_i)) \leq \frac{L}{32\mu}$, and this means the load of the points in $W_i(\tau_i) \setminus W_i(a)$ that are uncolored after iteration i is at least $\frac{L}{3} - \frac{L}{32\mu} > \frac{L}{5\mu}$. But $W_i(\tau_i) \setminus W_i(a) \subseteq W$, and this means that the load of the uncolored points in W after iteration i is at least $\frac{L}{5\mu}$. \square

The following lemma handles wedges which are subantipodal with respect to an i -wedge.

Lemma 9 *Suppose that W is a wedge that is subantipodal with respect to an i -wedge and $\text{load}_{X_i}(W) \geq \frac{L}{6}$. Further suppose that before the call to $\text{computeCover}(i, X_i, \frac{L}{64\mu})$ in iteration i , all i -wedges with apex on $C_i(k)$ have load at least $\frac{L}{2}$ from points in X_i . After the call to $\text{computeCover}(i, X_i, \frac{L}{64\mu})$ in the i th iteration of the algorithm, W*

Fig. 10 Illustration for Case 2(a): the region R_z



has load at least $\frac{L}{5\mu}$ from points in Y' . (Note that Y' always denotes the uncolored points in the algorithm.)

Proof Let x denote the apex of W . We first consider the case when both halflines of W intersect $C_i(k)$. Consider any point $z \in W \cap C_i(k)$. Let a_z denote the “leftmost” point where the boundaries of W and $W_i(z)$ intersect, and let b_z denote the “rightmost” point where the boundaries of W and $W_i(z)$ intersect (we say leftmost and rightmost because the intersection could be an interval and not just a single point). Let R_z be the quadrilateral with vertices $a_z, x, b_z,$ and z . That is, $R_z = W \cap W_i(z)$.¹ Suppose that $\text{load}_{X_i}(R_z) \geq \frac{L}{5\mu} + \frac{L}{32\mu}$. By Observation 6, $\text{load}_{Y_i}(W_i(z)) \leq \frac{L}{32\mu}$, and since all points in R_z are in $W_i(z)$, R_z contains (uncolored) load at least $\frac{L}{5\mu}$ after iteration i . Since $R_z \subseteq W$, W contains (uncolored) load at least $\frac{L}{5\mu}$ after iteration i , and we are done. See Fig. 10 for an illustration.

So we now assume that $\text{load}_{X_i}(R_z) \leq \frac{L}{5\mu} + \frac{L}{32\mu}$ for each $z \in W \cap C_i(k)$. Since $\text{load}_{X_i}(W_i(z)) \geq \frac{L}{2}$, we must have $\text{load}_{X_i}(W_i(a_z) \cup W_i(b_z)) \geq \frac{L}{2} - (\frac{L}{5\mu} + \frac{L}{32\mu}) > \frac{L}{8}$. Let z_1 be the “leftmost” point on $C_i(k) \cap W$, and let z_2 be the “rightmost” point on $C_i(k) \cap W$. Let us define the points a_{z_1} and b_{z_2} with similarly to how we defined a_z and b_z before. Notice that a_{z_1} is just z_1 itself, and so $\text{load}_{X_i}(W_i(a_{z_1})) \geq \frac{L}{2}$. Similarly, $\text{load}_{X_i}(W_i(b_{z_2})) \geq \frac{L}{2}$. Let z' be the last point on $C_i(k)$, while walking from z_1 to z_2 , such that $\text{load}_{X_i}(W_i(a_{z'})) \geq \frac{L}{16}$. (Note that $a_{z_1}, b_{z_2}, a_{z'}$, and $b_{z'}$ are defined using $z_1, z_2,$ and z' in the same way as a_z and b_z were defined with z before.) Thus

$$\text{load}_{X_i}(W_i(b_{z'})) \geq \text{load}_{X_i}(W_i(a_{z'}) \cup W_i(b_{z'})) - \frac{L}{16} \geq \frac{L}{8} - \frac{L}{16} = \frac{L}{16}.$$

See Fig. 11 for an illustration. Intuitively, we are choosing our point z' to be a point that will contain two wedges $W_i(a_{z'})$ and $W_i(b_{z'})$ that each have “a lot” of uncolored points.

Now consider any point $z'' \in W \setminus W_i(z')$. It must be that $W_i(z'')$ either contains $W_i(a_{z'})$ or contains $W_i(b_{z'})$ which both have load in X_i of at least $\frac{L}{16}$. Suppose that $W_i(z'')$ contains $W_i(a_{z'})$; the other case is similar. The points in $W_i(a_{z'})$ dominate z'' and we will not color z'' in iteration i until we have colored all points in $W_i(a_{z'}) \cap X_i$. But since $\text{load}_{Y_i}(W_i(a_{z'})) \leq \frac{L}{32\mu} < \frac{L}{16} \leq \text{load}_{X_i}(W_i(a_{z'}))$, this means we will not color z'' .

¹To be more precise, a_z is encountered after x and before z while traversing the boundary of R_z clockwise. Then b_z is encountered after z and before x .

Fig. 11 Illustration for Case 2(a): the constructed point z'

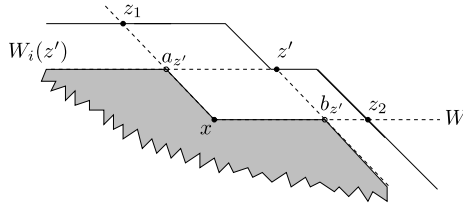


Fig. 12 Illustration of the case when the halflines of W do not intersect $C_i(k)$

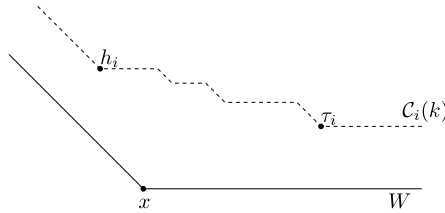
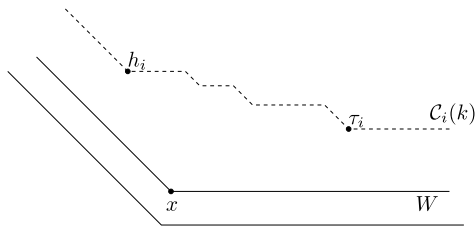


Fig. 13 The first step in constructing W'



It follows that $W \cap Y_i \subseteq W_i(z') \cap Y_i$, and thus $\text{load}_{Y_i}(W) \leq \text{load}_{Y_i}(W_i(z')) \leq \frac{L}{32\mu}$. And so the load of uncolored points in W after iteration i is at least $\frac{L}{6} - \frac{L}{32\mu} \geq \frac{L}{5\mu}$.

We now consider the case when one or both halflines of W do not intersect $C_i(k)$; see Fig. 12. We will show that we can find a wedge W' such that both halflines of W' intersect $C_i(k)$, $W' \cap Y = W \cap Y$, and W' is subantipodal with respect to an i -wedge. Given that W' exists, we can use the previous arguments to show that W' contains $\Omega(k)$ uncolored points after iteration i . It then follows that W contains $\Omega(k)$ uncolored points after iteration i because $W' \cap Y = W \cap Y$.

We will now describe how to find the wedge W' . We begin by placing a wedge identical to W “just behind” W so that the new wedge contains exactly the same points of Y that W contains; see Fig. 13. We then “bend in” the halflines of the wedge just enough so that the wedge still contains the same points as W and the halflines are no longer parallel with the edges of $C_i(k)$. This is our wedge W' ; see Fig. 14. Both halflines now intersect $C_i(k)$ and W' is subantipodal with respect to an i -wedge. By our previous analysis, W' contains at least $\frac{L}{5\mu}$ uncolored points after iteration i . Since W and W' contain the same points in Y , it follows that W contains at least $\frac{L}{5\mu}$ uncolored points after iteration i . \square

The previous two lemmas complete the proof of Theorem 7, which then in turn completes the proof of Theorem 2 for the wedges of a centrally-symmetric convex polygon.

Fig. 14 The second step in constructing W'

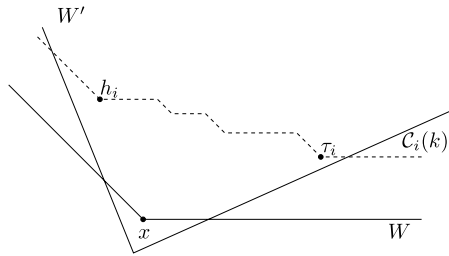
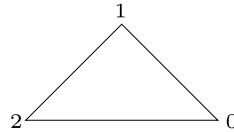


Fig. 15 The triangle used in Fig. 16



4 Arbitrary Convex Polygons

In this section, we prove Theorem 2 for an arbitrary convex polygon. We begin by showing that our algorithm for centrally-symmetric polygons does not work in the case of a triangle, a polygon which is not centrally-symmetric. We then describe how the algorithm is to be modified. We analyze the algorithm when the input polygon is a triangle, the simplest case that is not centrally-symmetric. Following this, we extend the analysis to the general case, invoking the tools developed for the previous special cases.

4.1 A Bad Example for Algorithm 1

The reason that Algorithm 1 works for centrally-symmetric polygons, but does not work for triangles, is because of the ways that wedges from a triangle can intersect. Using the definitions from the previous section, a triangle could have a wedge that is not nonantipodal or subantipodal with respect to an i -wedge. Because of this, it could be possible that when running `computeCover()` for $C_i(k)$ that all of the points inside of some j -wedge could be eligible to be colored in iteration i , and it could be possible that they are all assigned the same color.

For an example of this, see Figs. 15 and 16. Suppose we are working with the triangle P in Fig. 15. In Fig. 16, each of the small y_i dots correspond to a single point in Y , and the larger \mathcal{Y}_i dots correspond with $k - 1$ points in Y . The level curve $C_0(k)$ is drawn with dotted lines, and a 2-wedge $W_2(x)$ is drawn with solid lines. $W_2(x)$ contains only the points y_i for $i = 1, 2, \dots, k$. We want to run `computeCover()` for $C_0(k)$ to obtain $\Omega(k)$ colors, while hoping that $\Omega(k)$ points from $W_2(x)$ are left uncolored. However, `computeCover()` could pick all of the y_i points in its first iteration and assign all of these points the same color. $W_2(x)$ contains at least k points from Y , but all of the points it contains have now been colored, and assigned the same color. Thus, the algorithm fails.

Fig. 16 Algorithm 1 would assign all of the points in $W_2(x)$ the same color when coloring points for $C_0(k)$

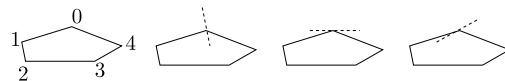
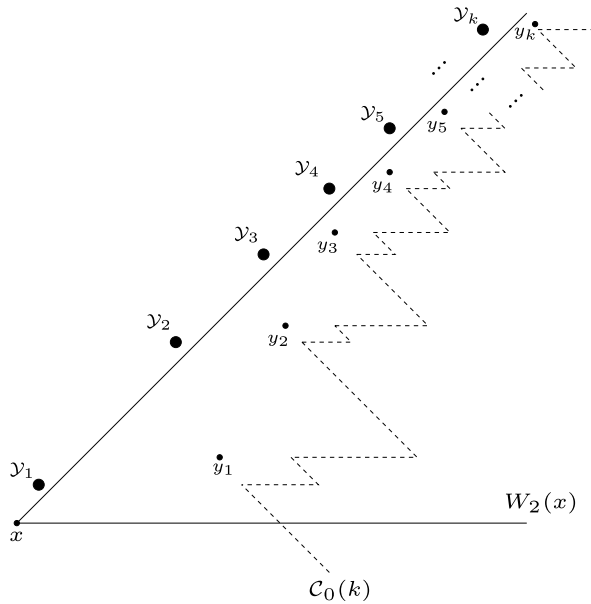


Fig. 17 An example of a set A_i . In this example, $A_0 = \{2\}$ since only the side of P parallel with p_2p_3 has the qualifying property

4.2 The Algorithm for General Convex Polygons

Recall that in the i th iteration of the algorithm, `computeCover()` takes as input a set of points $X_i \subseteq Y$. Only points from X_i are eligible to be assigned a color during iteration i . The way we modify the algorithm to work for all convex polygons is by choosing our set X_i more carefully. For centrally-symmetric polygons, it suffices to let X_i consist of all uncolored points in $\mathcal{W}_i^{\leq k}$. In the case of triangles, we want to choose a subset of the uncolored points in $\mathcal{W}_i^{\leq k}$. Intuitively, we want to only color the points which are the “furthest away” from $C_i(k)$. By doing this, we can show that if we assign a color to a point within some j -wedge, then there must be “a lot” of points remaining within the j -wedge.

Now we will introduce a notion which will be crucial in determining which points are “far away” from $C_i(k)$. Consider the natural linear ordering of the lines parallel to side $p_j p_{j+1}$ of P with the line through vertices p_j and p_{j+1} being smaller than the line through any of the other vertices of P . For $x, y \in \mathbb{R}^2$, we define the partial order $<_j$ such that $x <_j y$ if the $p_j p_{j+1}$ parallel line through x is less than the $p_j p_{j+1}$ parallel line through y . A similar notion was used in [11].

For a vertex p_i , let A_i denote the set of all indices j such that P has a tangent line through p_i parallel to $p_j p_{j+1}$. See Fig. 17 for an example.

Algorithm 2

- 1: $Y' \leftarrow Y$
 - 2: **for** each $i \in \{0, 1, 2, \dots, \mu - 1\}$ **do**
 - 3: $L \leftarrow \min\{\text{load}_{Y'}(W_z(x)) : x \in \mathcal{C}_z(k) \text{ and } z = i, i + 1, \dots, \mu - 1\}$
 - 4: **for** each $c \in \mathcal{C}_i(k)$ **do**
 - 5: **for** each $j \in A_i$ **do**
 - 6: Let $X^j(c)$ be the first $\frac{L}{2\mu}$ points in $W_i(c) \cap Y'$ in decreasing order with respect to the ordering $<_j$.
 - 7: $X(c) \leftarrow \{W_i(c) \cap Y'\} \setminus \bigcup_{j \in A_i} X^j(c)$
 - 8: $X_i \leftarrow \bigcup_{c \in \mathcal{C}_i(k)} X(c)$
 - 9: Run `computeCover`($i, X_i, \frac{L}{64\mu}$). Let $Y_i \subseteq X_i$ be the points assigned a color during this call.
 - 10: Let Y' denote the uncolored points (i.e., $Y' \leftarrow Y' \setminus Y_i$).
-

The algorithm is Algorithm 2. Note that the only difference between this algorithm and Algorithm 1 is in how X_i is chosen.

We will first analyze the algorithm in the case when P is a triangle. We will then show that we can use the ideas from the triangle case as well as the centrally-symmetric case to analyze the algorithm when P is an arbitrary convex polygon.

4.3 Analysis for Triangles

First, we will elaborate on how Algorithm 2 chooses the set X_i for a triangle. Recall the definition of the set A_i . For triangles, $A_i = \{i + 1\}$ since only the line parallel with $p_{i+1}p_{i+2}$ has the qualifying property. Note that this line is parallel with the side of the triangle opposite of p_i .

So when we get to the loop in Step 5 of the algorithm in iteration i , we will only compute the set $X^{i+1}(c)$ in Step 6. This set will consist of the first $\frac{L}{6}$ points in $W_i(c) \cap Y'$ in decreasing order with respect to the ordering $<_{i+1}$. The sorting is thus done with respect to the side of the triangle opposite of p_i . Thus in Step 7, the set $X(c)$ is the last $\frac{5L}{6}$ points in $W_i(c) \cap Y'$ in decreasing order with respect to the ordering $<_{i+1}$. Then finally in Step 8, X_i is simply the union of $X(c)$ for all $c \in \mathcal{C}_i(k)$.

The analysis for this algorithm has the same flavor as the analysis of Algorithm 1. In iteration i of the algorithm, we will assign colors to some of the points in X_i so that any i -wedge with apex on $\mathcal{C}_i(k)$ contains points colored with $\Omega(L)$ colors where L is as defined in the algorithm. This is $\Omega(k)$ different colors as long as L , which is k before the 0th iteration, is $\Omega(k)$ before the i th iteration. This is ensured in the remainder of this section.

The analysis will again rely upon Observation 6, which is that for any $z \in \mathcal{C}_i(k)$, we have that $\text{load}_{Y_i}(W_i(z)) \leq \frac{L}{32\mu}$. This holds also for Algorithm 2. We also need the following observation.

Observation 10 For any $z \in \mathcal{C}_i(k)$, $\text{load}_{X_i}(W_i(z)) \geq \frac{L}{2}$.

Fig. 18 The wedge W is triangular with respect to an i -wedge because there is a triangle T with the required property

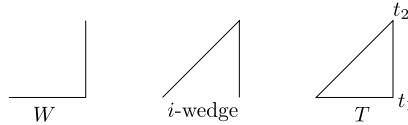
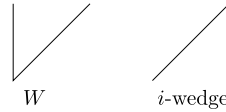


Fig. 19 The wedge W is not triangular with respect to an i -wedge because the required triangle does not exist



To see Observation 10, note that for each $c \in C_i(k)$, we have:

$$\text{load}_{X_i}(W_i(c)) \geq \text{load}_{X(c)}(W_i(c)) \geq L - |A_i| \frac{L}{2\mu} \geq \frac{L}{2}.$$

Suppose we have a wedge W of any type other than i . We say that W is *triangular with respect to an i -wedge* if there exists a triangle T with vertices t_1 and t_2 such that the edges of T adjacent to t_1 are parallel with the halflines of W and the edges of T adjacent to t_2 are parallel with the halflines of an i -wedge. Note that for this to be possible, W must have a halfline that is parallel with a halfline of an i -wedge. Also note that by the definition, a wedge of a triangle is triangular with respect to every other type of wedge of the same triangle; see Figs. 18 and 19.

We will argue that each wedge W (of type other than i) will have at most a constant factor of its uncolored points assigned a color during iteration i of the algorithm. Again, we write the lemma in a general form as we will use it again in a later section.

Lemma 11 *Suppose that W is a wedge that is triangular with respect to an i -wedge and $\text{load}_{X_i}(W) \geq \frac{L}{6}$. Suppose that the side of W that is not parallel to any of the two sides of the i -wedge is parallel to $p_j p_{j+1}$, where $j \in A_i$. (Note that if P is a triangle then $j = i + 1$.) Further suppose that before the call to $\text{computeCover}(i, X_i, \frac{L}{64\mu})$ in iteration i , all i -wedges with apex on $C_i(k)$ have load at least $\frac{L}{2}$ from points in X_i . After the call to $\text{computeCover}(i, X_i, \frac{L}{64\mu})$ in the i th iteration of the algorithm, W has load at least $\frac{L}{5\mu}$ from points in Y' . (Note that Y' always denotes the uncolored points in the algorithm.)*

Proof Let x denote the apex of W . First suppose that $W \cap C_i(k) = \emptyset$. Note that one of the halflines of W is parallel with one of the halflines of an i -wedge. We will assume that W has a halfline parallel with the side $p_{i-1} p_i$. The other case is symmetric; see Fig. 20. In this case, we have that $W \cap X_i \subseteq W_i(h_i) \cap X_i$ by the definition of the head h_i . Since $W \cap X_i \subseteq W_i(h_i) \cap X_i$, we have that $\text{load}_{Y_i}(W) \leq \frac{L}{32\mu}$. Therefore, there is (uncolored) load at least $\frac{L}{6} - \frac{L}{32\mu} > \frac{L}{5\mu}$ after iteration i .

Now we will assume that $W \cap C_i(k) \neq \emptyset$. Let $z \in C_i(k)$ be a point such that $W_i(z) \cap W \neq \emptyset$. Since we know that W has a halfline that is parallel with one of the halflines of $W_i(z)$, we know that there are only two types of intersections between these two wedges:

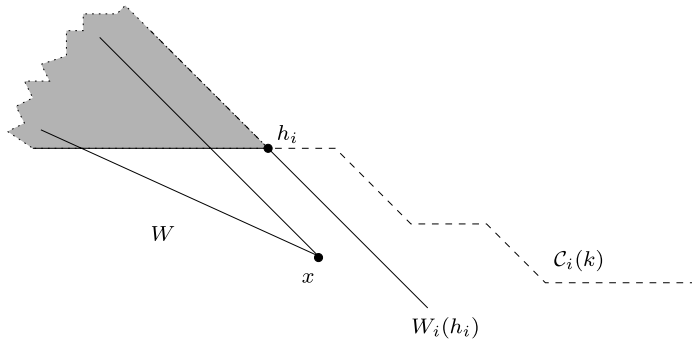
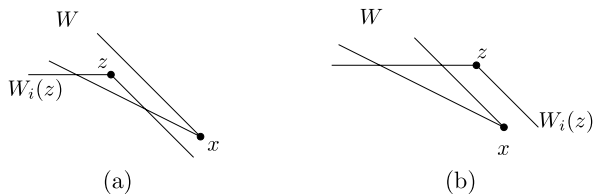


Fig. 20 Illustration for Lemma 11. Note that there cannot be any points in the shaded region due to the definition of the head h_i

Fig. 21 An illustration for the triangular case. (a) A type 1 intersection. (b) A type 2 intersection



1. $z \in W$, one halfline of W intersects both halflines of $W_i(z)$, and the other halfline of W does not intersect with $W_i(z)$.
2. $x \in W_i(z)$, one halfline of $W_i(z)$ intersects both halflines of W , and the other halfline of $W_i(z)$ does not intersect with W .

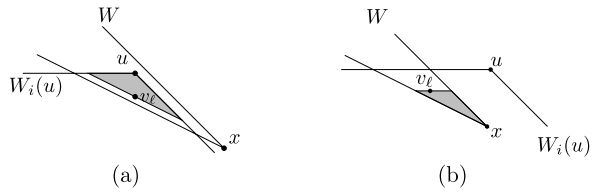
See Fig. 21 for an illustration.

Let $\{v_1, v_2, v_3, \dots\}$ denote the points in $X_i \cap W$ in decreasing order according to the ordering $<_i$. Let $\ell = \max\{t | v_t \in Y_i\}$. If $\ell \leq \frac{L}{10}$ then $\text{load}_{Y'}(W) \geq \frac{L}{6} - \frac{L}{10} \geq \frac{L}{5\mu}$ after iteration i , so assume that $\ell > \frac{L}{10}$. For $t \in \{0, 1, \dots, \mu - 1\}$ and for a point $s \in \mathbb{R}^2$, let $H_t(s)$ denote the halfplane consisting of all points y such that $y \leq_t s$.

Since $v_\ell \in X_i$ there is a $u \in C_i(k)$ so that $v_\ell \in X(u)$ in iteration i of the algorithm. Suppose that the intersection between $W_i(u)$ and W is a type 1 intersection. Let $T_{v_\ell} = W_i(u) \setminus H_j(v_\ell)$. Since $v_\ell \in X(u)$, we know that v_ℓ is one of the last $L - \frac{L}{2\mu}$ points in $W_i(u) \cap Y'$ with respect to the ordering $<_j$. (See Algorithm 2 for the notation.) Because we chose the points for $X(u)$ with respect to this ordering, we know that the first $\frac{L}{2\mu}$ points with respect to the ordering $<_j$ from $W_i(u) \cap Y'$ must be in T_{v_ℓ} . Since, $\text{load}_{Y_i}(W_i(u)) \leq \frac{L}{32\mu}$, there must be at least $\frac{L}{2\mu} - \frac{L}{32\mu} > \frac{L}{5\mu}$ uncolored points left in T_{v_ℓ} after iteration i . Since we are dealing with a type 1 intersection, $T_{v_\ell} \subset W$, and thus W will contain at least $\frac{L}{5\mu}$ uncolored points after iteration i and the lemma holds. See Fig. 22(a) for an illustration.

Now suppose that the intersection between $W_i(u)$ and W is a type 2 intersection. Consider the region $T'_{v_\ell} = W \setminus H_i(v_\ell)$. Since we are assuming $\ell > \frac{L}{10}$ and since $v_\ell \in W_i(u)$, it must be that $\text{load}_{X_i}(T'_{v_\ell}) \geq \frac{L}{10}$. Since we are dealing with a type 2

Fig. 22 An example of the key triangular regions. (a) An illustration of T_{v_ℓ} (the shaded region). (b) An illustration of T'_{v_ℓ} (the shaded region)



intersection, it must be that $T'_{v_\ell} \subset W_i(u)$. Since $\text{load}_{Y_i}(W_i(u)) \leq \frac{L}{32\mu}$, we have that $\text{load}_{Y_i}(T'_{v_\ell}) \leq \frac{L}{32\mu}$ and thus there will be at least $\frac{L}{10} - \frac{L}{32\mu} > \frac{L}{5\mu}$ uncolored points left in T'_{v_ℓ} after iteration i . Since $T'_{v_\ell} \subseteq W$, there must be (uncolored) load at least $\frac{L}{5\mu}$ in W after iteration i . See Fig. 22(b) for an illustration. \square

At this point, it is easy to prove a theorem similar to Theorem 7 to show that Theorem 2 holds for the wedges of a triangle.

4.4 Analysis for Convex Polygons

In this section, we will analyze Algorithm 2 in the case when P is an arbitrary convex polygon, thus proving Theorem 2. We will show that any j -wedge that contains $\Omega(k)$ uncolored points prior to coloring points for $C_i(k)$ will still contain $\Omega(k)$ uncolored points after doing the coloring. We will use Lemmas 8, 9, and 11 in the analysis. We note that the statements of Lemmas 8 and 9 and their proofs hold for Algorithm 2 as well. If p_i and p_j are not antipodal vertices of P , then we can use the analysis of Lemma 8. If p_i and p_j are antipodal vertices of P , then we show that we can partition the j -wedge into at most 3 wedges. One of these wedges can be analyzed with Lemma 9 (the subantipodal case used for centrally-symmetric polygons), and the other two wedges (if they exist) can be analyzed with the same arguments from Lemma 11 (the triangle case).

Lemma 12 *Suppose at the beginning of iteration i , all j -wedges with apex on $C_j(k)$ have load at least L from points in Y' for $j \geq i$, where L is as computed in the i th iteration of the algorithm. (Note that Y' always denotes the uncolored points in the algorithm.) After the i th iteration of the algorithm, any j -wedge $W_j(x)$, for $j > i$, and with apex x on $C_j(k)$ has load at least $\frac{L}{5\mu}$ from points in Y' .*

Proof There are two main cases to consider:

- p_i and p_j are antipodal vertices of P , that is, there are parallel lines through p_i and p_j such that both of the lines are tangent to P .
- p_i and p_j are not antipodal vertices of P .

Case 1: p_i and p_j are not antipodal vertices of P .

In this case, we can invoke Lemma 8 for each j -wedge $W_j(x)$ with apex $x \in C_j(k)$ such that $\text{load}_{X_i}(W_j(x)) \geq \frac{L}{6}$ (Lemma 12 trivially holds for $W_j(x)$ if $\text{load}_{X_i}(W_j(x)) < \frac{L}{6}$). We simply use the arguments from Lemma 8 with $W := W_j(x)$.

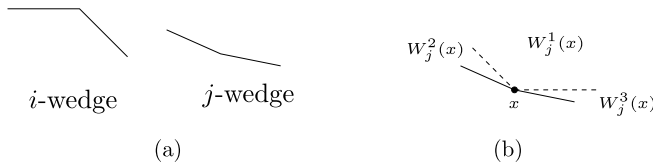
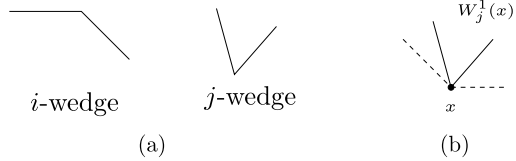


Fig. 23 An illustration for the antipodal case. If we are working with the corresponding i -wedge and j -wedge (part (a)), then we obtain the corresponding $W_j^1(x)$, $W_j^2(x)$, and $W_j^3(x)$ (part (b))

Fig. 24 An illustration for the antipodal case. If we are working with the corresponding i -wedge and j -wedge (part (a)), then $W_j^1(x) = W_j(x)$ and $W_j^2(x) = W_j^3(x) = \emptyset$ (part (b))



Case 2: p_i and p_j are antipodal vertices of P .

Since $\text{load}_Y(W_j(x)) \geq L$, if $\text{load}_{X_i}(W_j(x)) \leq \frac{L}{2}$ then $W_j(x)$ will clearly have load at least $\frac{L}{5\mu}$ after iteration i . So assume that $\text{load}_{X_i}(W_j(x)) > \frac{L}{2}$.

Consider the line parallel with $p_{i-1}p_i$ through x and the line parallel with $p_i p_{i+1}$ through x . Note these lines are parallel with the halflines of an i -wedge. Let $H_i(x)$ denote the halfplane consisting of all points y such that $y \leq_t x$. Let $W_j^1(x) = H_{i-1}(x) \cap H_i(x) \cap W_j(x)$. Let $W_j^2(x) = (H_i(x) \cap W_j(x)) \setminus W_j^1(x)$. Let $W_j^3(x) = (H_{i-1}(x) \cap W_j(x)) \setminus W_j^1(x)$. See Figs. 23 and 24 for an illustration. Note that $W_j^1(x)$, $W_j^2(x)$, and $W_j^3(x)$ form a partition of $W_j(x)$. Also note that $W_j^1(x)$ cannot be empty but $W_j^2(x)$ or $W_j^3(x)$ could be empty. Since these three sets form a partition of $W_j(x)$ and $\text{load}_{X_i}(W_j(x)) > \frac{L}{2}$, it must be that one of the three sets has load at least $\frac{L}{6}$ from X_i .

Note that $W_j^1(x)$ is subantipodal with respect to an i -wedge. We thus handle the case when $\text{load}_{X_i}(W_j^1(x)) \geq \frac{L}{6}$ by invoking Lemma 9 with $W := W_j^1(x)$.

Note that $W_j^2(x)$ and $W_j^3(x)$ (assuming they are nonempty) are triangular with respect to an i -wedge. Thus we can handle the cases when $\text{load}_{X_i}(W_j^2(x)) \geq \frac{L}{6}$ and $\text{load}_{X_i}(W_j^3(x)) \geq \frac{L}{6}$ with Lemma 11 with $W := W_j^2(x)$ or $W := W_j^3(x)$, completing the proof. \square

We will now show that Theorem 2 holds for the wedges of any convex polygon:

Theorem 13 Let \bar{P} be any convex polygon with μ vertices. Let $Y \subset \mathbb{R}^2$ be any set of points so that $|Y| \geq k \geq 1000\mu \cdot (5\mu)^\mu$, where k is a parameter. Algorithm 2 colors the points in Y with $\Omega(k)$ colors in a way such that for any j -wedge W corresponding to \bar{P} such that $|W \cap Y| \geq k$, W contains a point of each color.

Proof Algorithm 2 calls `computeCover`(i, X_i, t) for each $0 \leq i \leq \mu - 1$. The set X_i in the i th iteration is an appropriately chosen subset of the points in Y not colored

in iterations $0, 1, \dots, i - 1$. At the beginning of the i th iteration, let L denote, as in the algorithm, the smallest number of uncolored points in a j -wedge with apex on $\mathcal{C}_j(k)$, for $i \leq j \leq \mu - 1$. The parameter t is chosen to be $\frac{L}{64\mu}$, and we have $\min_{c \in \mathcal{C}_i(k)} |W_i(c) \cap X_i| \geq \frac{L}{2}$ (due to the manner in which X_i is chosen in the algorithm, see Observation 10). After the call to `computeCover`($i, X_i, \frac{L}{64\mu}$), any i -wedge with apex on $\mathcal{C}_i(k)$ contains points colored $1, 2, \dots, L/64\mu$. Thus, the algorithm produces a coloring as required in Theorem 2, provided $L \in \Omega(k)$. This is established by Lemma 12. It states that L , which equals k before the 0th iteration, drops by a factor of at most 5μ with each iteration. \square

This completes the proof of Theorem 2, and thus Theorem 1. We point out that our proof technique in fact yields an efficient algorithm corresponding. That is, there is a polynomial time algorithm that, given a k -fold covering using translates of the convex polygon P , will output $\Omega(k)$ covers.

5 Conclusion and Open Problems

We now know that k -fold coverings by convex polygons can be “optimally decomposed”. The major open problem is to obtain results for disks, even in the case when all of the disks have unit radius. The techniques for obtaining $\Omega(k)$ covers for convex polygons involve reducing the problem to coloring points inside of wedges. We are able to understand the complications involving the interactions of the wedges, and we are able to color the points in a clever way to obtain our result. This largely has to do with the fact that there are only a constant number of “types” of wedges; however, we do not seem to have the tools to handle the interactions of the disks.

Acknowledgements We would like to thank the anonymous referees for their valuable feedback.

References

1. Aloupis, G., Cardinal, J., Collette, S., Langerman, S., Orden, D., Ramos, P.: Decomposition of multiple coverings into more parts. In: SODA'09: Proceedings of the Nineteenth Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 302–310. Society for Industrial and Applied Mathematics, Philadelphia (2009)
2. Buchsbaum, A.L., Efrat, A., Jain, S., Venkatasubramanian, S., Yi, K.: Restricted strip covering and the sensor cover problem. In: SODA'07: Proceedings of the Eighteenth Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 1056–1063. Society for Industrial and Applied Mathematics, Philadelphia (2007)
3. Mani, P., Pach, J.: Decomposition problems for multiple coverings with unit balls. Manuscript (1986)
4. Pach, J.: Covering the plane with convex polygons. *Discrete Comput. Geom.* **1**, 73–81 (1986)
5. Pach, J., Tóth, G.: Decomposition of multiple coverings into many parts. *Comput. Geom.* **42**(2), 127–133 (2009)
6. Pach, J., Tardos, G., Tóth, G.: Indecomposable coverings. In: Akiyama, J., Chen, W.Y.C., Kano, M., Li, X., Yu, Q. (eds.) CJDCGCT. Lecture Notes in Computer Science, vol. 4381, pp. 135–148. Springer, Berlin (2005)
7. Pálvölgyi, D.: Indecomposable coverings with concave polygons. *Discrete Comput. Geom.* **44**(3), 577–588 (2010). doi:[10.1007/s00454-009-9194-y](https://doi.org/10.1007/s00454-009-9194-y)

8. Pálvölgyi, D., Tóth, G.: Convex polygons are cover-decomposable. *Discrete Comput. Geom.* **43**(3), 483–496 (2010)
9. Pandit, S., Pemmaraju, S.V., Varadarajan, K.R.: Approximation algorithms for domatic partitions of unit disk graphs. In: Dinur, I., Jansen, K., Naor, J., Rolim, J.D.P. (eds.) *APPROX-RANDOM*. Lecture Notes in Computer Science, vol. 5687, pp. 312–325. Springer, Berlin (2009)
10. Pemmaraju, S.V., Pirwani, I.A.: Energy conservation via domatic partitions. In: *MobiHoc'06: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 143–154. ACM, New York (2006)
11. Tardos, G., Tóth, G.: Multiple coverings of the plane with triangles. *Discrete Comput. Geom.* **38**(2), 443–450 (2007)