Collapsing the Hidden-Set Convex-Cover Inequality

Reilly Browne ✉
Stony Brook University http://rajalo.github.io

Eric Chiu ✉
Stony Brook University https://github.com/echiu12

Abstract

We explore two closely related visibility problems: the maximum hidden set and the minimum convex cover. It is known that the convex cover number ($cc(P)$) for any simple polygon is greater than or equal to the hidden set number ($hs(P)$). We define a new subclass of polygons, homestead polygons, as those which have $hs(P) = cc(P)$. We present subclasses whose members are homestead polygons and whose members can be homesteads or not homesteads. For histograms and spirals, we give linear time algorithms which find both a hidden set and a convex cover of the same size, improving on observations from Bajuelos et al. (2008)[2] for finding the maximum hidden set when restricted to vertices. We also show that in general, deciding if a polygon is a homestead polygon is NP-hard.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Graph theory

Keywords and phrases Convex Cover, Clique Cover, Hidden set, Independent Set

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Acknowledgements We want to thank Joseph Mitchell, Valentin Polishchuk, Christiane Schmidt, Kien Huynh and the rest of CG Group for discussing these problems with us.

1 Introduction

Across various fields and applications, polygons are used for modeling environments, molds, and physical objects in general and often it is more advantageous to break such a polygon into simpler pieces for increased efficiency of processing. There are several methods of decomposing polygons, but the version that motivates our work is the notion of convex cover. This version of decomposition first studied by Pavlidis [9] for its practical applications in pattern recognition. It differs from other forms of decomposition in that all the pieces must be convex polygons and pieces are allowed to overlap. O'Rourke and Supowit [8] showed that the problem of determining the minimum convex cover is NP-hard for polygons with or without holes, with more recent work by Abrahamsen [1] in 2021 showing that the problem is $\exists\mathbb{R}$-complete.

Our work focuses on how minimum convex cover is related to a visibility problem called maximum hidden set. Shermer [10] introduced the maximum hidden set problem and proved that it is NP-hard. Intuitively, a maximum hidden set in a polygon is a maximum cardinality set of points where no pair of points sees each other. Eidenbenz [3] subsequently showed that finding a maximum hidden set was APX-hard for simple polygons. Shermer [10] observed that the size of a maximum hidden set must be less than or equal to the size of a minimum convex cover, since no two hidden points could exist in the same convex piece. Shermer [10] also observed that hidden set is essentially the same as the graph theoretical notion of independent set, with hidden points being the indepdendent set of an infinite graph on all the points called a point visibility graph. The size of a minimum convex cover of a polygon is similarly equivalent to the clique cover number of the point visibility graph.

In the graph theory setting, when the independence number and clique cover number

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

are known to be equal, they can both be found in polynomial time. This is because another quantity, the Lovasv number, was shown by Knuth [7] to be "sandwiched" between the complement problems of independence number and clique cover number. Hence, when they coincide, the Lovasv number is equivalent to both and can be approximated in polynomial time due to work by Grötschel, Lovàsz, and Schrijver [5]. While a polynomial time algorithm in the number of nodes of a graph does not directly give us insight into how to develop a polynomial time algorithm for an infinite graph, studying subclasses of polygons where they coincide has allowed us to develop linear time algorithms for those subclasses.

We define the problems and concepts used for this discussion more rigorously, and introduce our own classification of polygon, homestead polygons.

▶ **Definition 1.** *[[10]] Given a polygon P, the **point-visibility graph of P**, $PVG(P) = (V, E)$ where $V = \{p \mid p \in P\}$ and $E = \{(x, y) \mid \overline{xy} \subset P\}$.*

▶ **Definition 2.** *For a polygon P, the **hidden set number of P**, hs(P), is the independence number of PVG(P).*

▶ **Definition 3.** *For a polygon P, the **convex cover number of P**, cc(P), is the minimum number of convex pieces needed to cover P. It is also the clique covering number of PVG(P).*

▶ **Definition 4.** *A polygon, P is a homestead polygon if $hs(P) = cc(P)$.*

2 Homestead Polygons

2.1 Combinatorial classes

From Shermer [10], we know that for a polygon P with r reflex vertices, the following inequality holds:

$$1 \leq hs(P) \leq cc(P) \leq r + 1$$

This implies the following two theorems and corollary:

▶ **Theorem 5.** *All convex polygons are homestead polygons.*

Proof. By definition, convex polygons has convex cover number 1. Hence, for convex polygon P, $1 \leq hs(P) \leq 1$, therefore $hs(P) = cc(P) = 1$. Since $hs(P) = cc(P)$, we know that P is a homestead, thus all convex polygons are homestead polygons. ◀

▶ **Theorem 6.** *Any polygon P with $hs(P) = r + 1$ is a homestead polygon.*

Proof. For any polygon P with $hs(P) = r + 1$, from the above inequality we know that $r + 1 \leq cc(P) \leq r + 1$, thus $hs(P) = cc(P) = r + 1$. Since $hs(P) = cc(P)$, we know that P is a homestead. ◀

▶ **Corollary 7.** *All spiral polygons are homestead polygons.*

Proof. From Bajuelos et al [2], we know that for any spiral polygon P, $hs(P) = r + 1$. Therefore all spiral polygons are homestead polygons. ◀

2.2 Spiral Polygon Algorithm

A spiral polygon is a simple polygon formed by two chains, one composed of entirely convex vertices and the other of entirely reflex vertices (excluding the two end-points that connect it to the convex chain). From the previous discussion, we know that spiral polygons have a convex cover and a hidden set of size $r+1$. Bajuelos et al. [2] note that a hidden set of size $r+1$ can be obtained by simply placing a point at the midpoint of each edge along the reflex chain. Thus, algorithmically, finding a hidden set is extremely trivial. However, finding a convex cover requires a small trick to achieve linear time. See Figure 1 for a demonstration of this algorithm.

We will consider a spiral polygon as a reflex chain and a convex chain. We iterate through the reflex chain, adding a pair of a hidden set and a convex piece to a running set of them. The hidden points are the aforementioned midpoints, and the convex pieces are formed by extending the edge that the new hidden point lies on. Since all the vertices in the reflex chain (except the two on the ends) are reflex, we know this extension will intersect the convex chain, and thus form a convex piece. We only need to extend out one way, since the very first edge of the reflex chain only extends one way and from there we can just extend past the vertex which was not part of the previous edge. This is because taking out the previous convex piece produces a spiral with one less reflex vertex and whose reflex chain starts with the edge now being considered.

Data: P, a spiral polygon (vertices in counterclockwise order)
Result: C, a set of pairs made of a hidden point and convex piece
$c_0 \leftarrow$ the convex vertex in P which has a reflex vertex preceding it;
$r \leftarrow c_0.prev$;
$c_1 \leftarrow c_0$ $c_2 \leftarrow c_0$ $C = \{\}$;
while *r is a reflex vertex* **do**
  $p \leftarrow ((r.x + r.prev.x)/2, (r.y + r.prev.y)/2)$;
  while *$c_1.next$ is not left of $\overrightarrow{r, r.prev}$* **do**
    $c_1 \leftarrow c_1.next$
  end
  $x \leftarrow$ intersection of $\overrightarrow{r, r.prev}$ and $\overrightarrow{c_1, c_1.next}$;
  $L \leftarrow \{r.prev, r, c_0\}$;
  while $c_2 \neq c_1$ **do**
    $L.add(c_2)$;
    $c_2 \leftarrow c_2.next$;
  end
  $m \leftarrow$ midpoint of $\overline{r.prev, r}$;
  $C.add(\{L, m\})$;
  $c_0 \leftarrow x$;
  $c_2 \leftarrow c_2.prev$;
  $r \leftarrow r.prev$;
end

Algorithm 1 An algorithm for finding both $hs(P)$ and $cc(P)$ in spiral polygons

To form these convex pieces using the edge extensions, we keep track of the intersection. We can just keep a pointer to a vertex in the convex chain and move forward through it until the edge corresponding to the vertex is intersected by the extension of the current reflex edge. From there, we simply list out the previous convex vertices, the previous intersection, the reflex vertex not being extended and the new intersection to get the convex piece. Since the extensions are all going to intersect the convex chain in order, we can simply keep moving this pointer forward throughout and only incur $O(n)$ work.

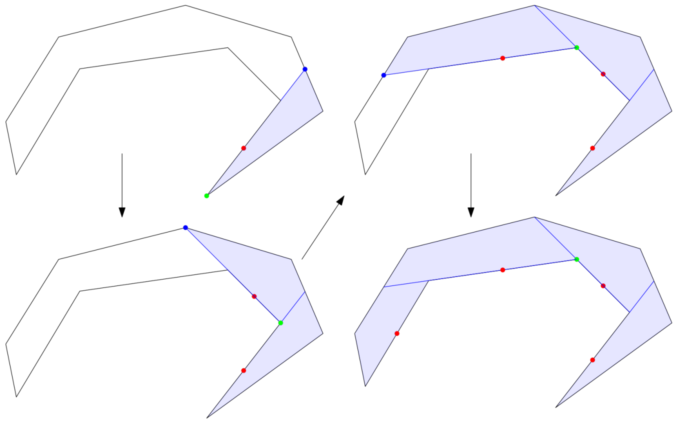

Figure 1 Demonstrating our algorithm for spiral polygons.

2.3 Histogram Polygons

A histogram polygon is a simple polygon formed by two chains whose x-coordinates increase monotonically, one of which has only one edge, and where all angles between edges of the polygon are orthogonal. Bajuelos et al.[2] presented a formula for a maximum hidden *vertex* set of a histogram polygon in general position given the number of "bottom sides". Counting these bottom sides takes $O(n)$ time, implying a linear time algorithm. Our algorithm improves upon that result by finding a maximum hidden set (no vertex or position constraint) and a minimum convex cover in $O(n)$. Also, Hoorfar and Bagheri[6] present an $O(n)$ time algorithm for the related problem of finding minimum hidden guards in histogram polygons under orthogonal vision, meaning 3 of the 5 hiding problems presented by Shermer [10] can be solved in $O(n)$ time for histogram polygons for at least one form of visibility.

We will consider the histogram polygon as an ordered list of “bars” from right to left. Each bar is the rectangle formed under each horizontal edge until the base is reached (a convex piece), paired with the midpoint of its top edge (a hidden point). We can decompose the polygon into these bars, creating an overestimate on convex cover and hidden set. To lower this, we merge bars of the same height without a boundary in between, which discards the newer hidden point and combines the rectangles.

Moving right to left, keep track of a set and a stack. The set will be our answer and the stack will keep track of all the bars which are candidates for merging. When considering a bar, we compare its height to the bars on the stack. If the bar is higher than the top of the stack or the stack is empty, we add it to the stack and the set. If the bar has equal height, we merge it with the top bar of the stack. If the bar has a lower height, then we pop the top bar off the stack and compare again. Continue until all bars have been considered and return the set of merged bars. See Figure 5 for an example run of the algorithm.

The result is a linear time algorithm (using the same stack operations argument as the Graham scan) which solves both maximum hidden set and minimum convex cover for histogram polygons. Since the algorithm always returns a hidden set and convex cover of the same size, histogram polygons must be homestead polygons. The pseudocode is given in Algorithm 2

Data: M, a histogram polygon
Result: C, a set of pairs made of a hidden point and convex piece
$p \leftarrow$ the rightmost vertex of the base;
$S \leftarrow$ an empty stack of pairs;
$C \leftarrow$ and empty set of pairs;
$yMin \leftarrow p.y$;
$p \leftarrow p.next$;
while $p.y > yMin$ **do**
  $q1 \leftarrow (p.next.x, minY)$; $q2 \leftarrow (p.x, minY)$;
  $R \leftarrow [p, p.next, q1, q2]$; $m \leftarrow ((p.next.x + p.x)/2, p.y)$;
  $pair \leftarrow [m, R]$;
  if S is empty **then** $S.push(pair)$; $C.add(pair)$;
  $possibleMatch \leftarrow S.pop()$;
  while *S is not empty* ***and*** $possibleMatch[1].y \leq p.y$ **do**
    $possibleMatch \leftarrow S.pop()$;
  end
  $T \leftarrow possibleMatch[0]$;
  $S.push(possibleMatch)$;
  if $T[0].y = p.y$ **then**
  $T[1] \leftarrow p.next$; $T[2] \leftarrow q1$;
  If $T[0].y > p.y$ **then** $S.push(pair)$; $C.add(pair)$;
end
return C;

Algorithm 2 An algorithm for finding both $hs(M)$ and $cc(M)$ for histogram polygon M

3 Nonhomestead Polygons

We present several simple polygons for which $hs(P) < cc(P)$. These polygons each have different

▶ **Theorem 8.** *There exists an orthogonal polygon which is not a homestead polygon*

Proof. Observe the example M in Figure 3. This polygon is clearly orthogonal. First we will show that M does not admit a hidden set larger than size 3 and then show that the minimum convex cover must be greater than or equal to 4 by using a theorem from graph theory.

We start by finding a convex cover of size 4. We can remove out the points that are in the intersection of 2 or more convex pieces since their use can only admit a hidden set of size 3 or smaller. This is because if among the two remaining convex pieces, there can only be an additional two points that can be hidden in them, by definition of convexity. See Figure 4, the regions outlined in purple are the remaining points that have not been eliminated.

Now we show that for each purple region R_i, after taking out the strong visibility region of R_i, the rest of M can be covered with just 2 convex pieces. By definition of strong visibility, all the points in R_i see all the points in the strong visibility region. Therefore, using any point in R_i disqualifies usage of the strong visibility region, hence covering the rest with 2 convex pieces means at most 2 additional hidden points can be used.

For some of the purple regions, we split R_i into 2 regions and analyze them separately because the strong visibility region of the whole of R_i is just a convex piece and thus we needed 3 to complete that cover, but when the region is split both pieces can have the

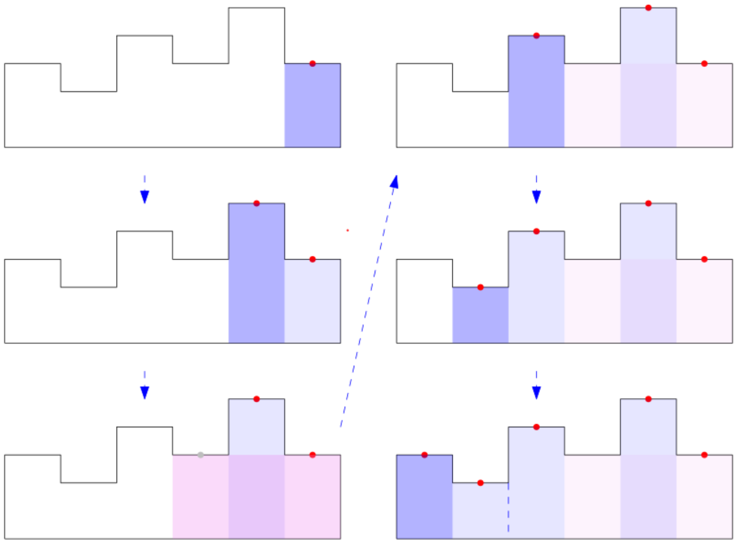

Figure 2 Demonstrating our algorithm for histogram polygons.

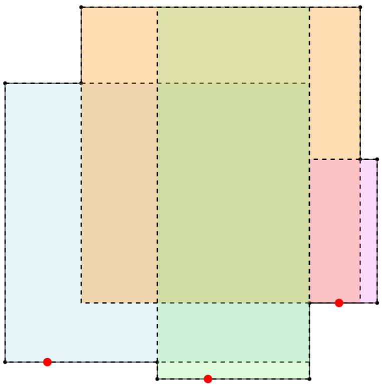

Figure 3 Orthogonal and x-monotone polygon, M which is not a homestead polygon.

remaining portion covered with just 2. This full analysis is shown in Figure 5, with the specific region being analyzed in each iteration shaded in purple.

Since, as is shown in Figure 5, including the strongly visible region of any purple region allows the rest of M to be covered with just 2 added convex pieces, we know that M admits hidden sets of size at most 3.

For convex cover, we find an indueced subgraph of the PVG of M with clique cover number 4. From Gella and Artes [4], we know that for graph G and induced subgraph H, the following inequality holds:

$$cc(G) \geq cc(H) \tag{1}$$

Therefore, since convex cover is simply the clique cover of the point visibility graph, finding that clique cover number of the point visbility graph in M restricted to a small set of points is 4 will suffice to show that $cc(M) \geq 4$.

We show this through finding the chromatic number of an isomorphism of the complement

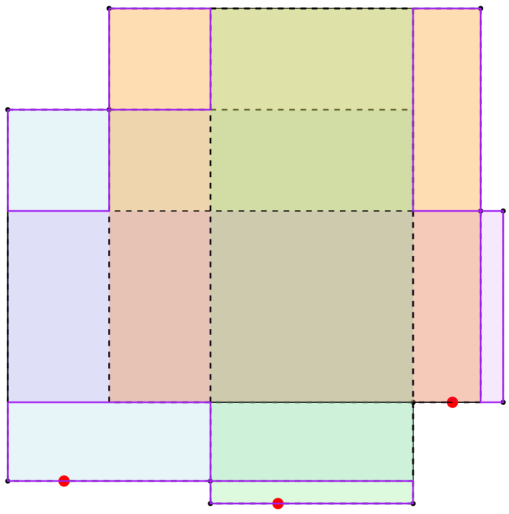

Figure 4 Showing the remaining regions of M that haven't been ruled out by a starting convex cover.

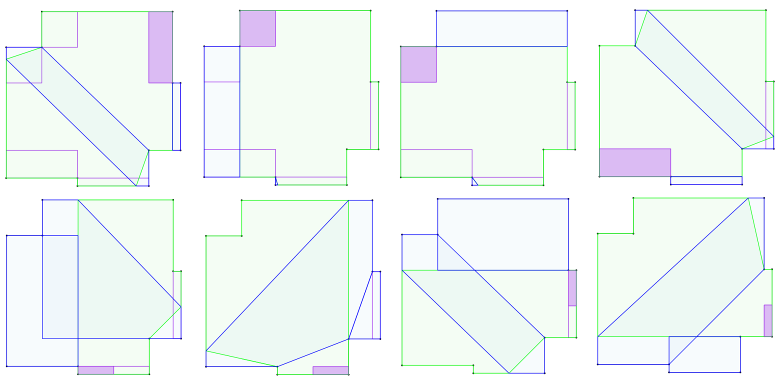

Figure 5 Showing the size 2 convex covers of the remaining region after taking out the strong visibility regions of the purple regions.

graph in Figure 6. Start by observing that nodes 1,2,3 form a clique in the complement, thus must all have different colors, say red, blue and green respectively. Node 6 is adjacent to both red (1) and blue (2), it must be green. Node 5 is adjacent to green (6) and red (1) and thus must be blue. Node 7 is adjacent to green (3) and blue (5) so it must be red. Since 4 is adjacent to green (3), red (7), and blue (5), it must be a fourth color. Therefore the chromatic number of G' must be 4. This in turn proves that $cc(G) = 4$ and in turn $cc(P) \geq 4$. Therefore M, an orthogonal polygon, is not a homestead. ◀

▶ **Corollary 9.** *There exists an x-monotone polygon which is not a homestead polygon.*

Proof. The polygon M is also clearly x-monotone in addition to being orthogonal. Therefore, this corollary is true from the previous proof that M is not a homestead. ◀

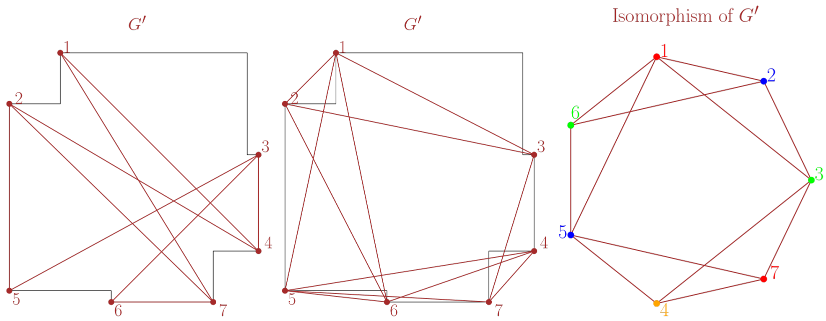

Figure 6 Showing an induced subgraph of $PVG(M)$ with $cc(G) = 4$ from chromatic number of compliment.

3.1 General position

Another subclass of polygon that we had conjectured was comprised only of homestead polygons was polygons whose vertices were in general position. A set of vertices is said to be in general position if no 3 vertices are collinear and no 4 vertices are said to be cocircular.

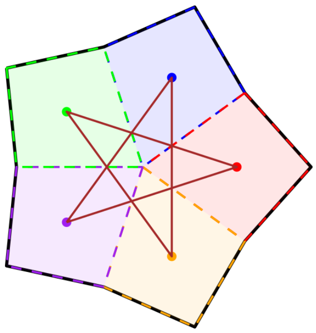

Figure 7 A polygon W which has no three collinear vertices and is not a homestead.

▶ **Theorem 10.** *There exists a polygon which is not a homestead polygon which has vertices in general position*

Proof. First we show a polygon W (in Figure 7 which is a polygon where no 3 of the vertices are collinear, but does have cocircular points, and is not a homestead polygon. The polygon exhibits rotational symmetry which allows us to simplify our analysis. First we show that the hidden set number of W is 2, then that its convex cover number is 3 and then give a modification of W' that does not change any of the properties which make W not a homestead.

For hidden set, we analyze the green region of W as, due to rotational symmetry, it is equivalent to analyzing all the other regions. Each of the colored regions have nodes in their centers. These nodes are connected if their regions are strongly visible to each other (their strong visibility regions contain all of the other). Any point in the green region sees all the points in the red and orange regions, meaning we need to cover the blue and purple regions with just one convex polygon. Since blue and purple have an edge, this implies that they see each other, so simply taking their convex hull will produce the convex piece. Because only 1 additional convex piece is needed after using the strong visibility region of the green region, and the green region is, by rotational symmetry, equivalent to all the other regions which total all of W, W admits hidden sets of size at most 2.

For convex cover, we take the convex vertices as our subgraph ("points of the star"). It is clear that the graph induced by these vertices is the same as that of the regions from the hidden set discussion, ie the convex vertex in the green region sees the vertices in the red and orange regions but not the blue and purple. This graph is C_5, a well known graph which has clique cover number 3.

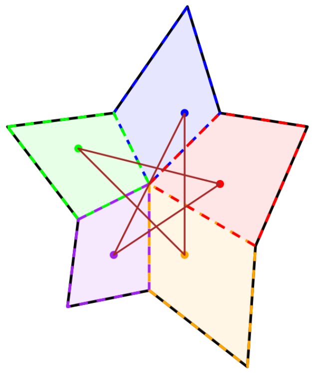

Figure 8 A polygon W' whose vertices are not in general position and which is not a homestead polygon.

Observing W' (see Figure 8), it is clear that the colored regions are related just as those of W. The convex vertices also are related in the same way as those of W. Therefore, W', a polygon which has vertices in general position, must have hidden set number of at most 2 and convex convex number of at least 3 and must not be a homestead polygon.

◀

▶ **Corollary 11.** *There exists a star-shaped polygon which is not a homestead polygon.*

Proof. A star shaped polygon is a polygon for which there is some point or set of points that see the entirety of the polygon. The center point of W is in all of the colored regions, therefore must see the entirety of W. Since W is ◀

3.2 NP-hardness of Homestead Decision Problem

We can define a decision problem with respect to homestead polygons by taking a polygon as an input and requiring an output of true or false indicating whether the polygon is a homestead polygon or not. The key component in the two NP-hardness reductions (both from 3-SAT, the variant of the Boolean satisfiability problem where all clauses have 3 literals) for hidden set and convex cover in simple polygons is the construction of a simple polygon which is a homestead if the 3-SAT instance is satisfiable and a nonhomestead if it is not satisfiable.

▶ **Theorem 12.** *Deciding if a simple polygon is a homestead polygon or not is NP-hard.*

Proof. It is sufficient to show that for any instance of 3-SAT, the corresponding construction from Shermer's [10] NP-hardness reduction for hidden set has exactly the convex cover number k which, if equal to the hidden set number, indicates that the instance is satisfied. This is because we can simply apply any algorithm for deciding if a polygon is a homestead polygon to this construction in the same way that Shermer takes the value k as input for a maximum hidden set decision algorithm. The full details of the construction are provided in Shermer's paper.

For a 3-SAT instance with m clauses and n variables, $k = 2mn + 8n + m + 1$. We show that there is a subgraph of every construction that has clique cover number equal to K. Shermer shows a set of convex pieces which cover the construction of size k, so we just need to prove that no cover with less pieces can exist using the subgraph argument. First we show the subgraph in Figure 9, with dotted edges signifying that the edge connects to a point in a different component. We use the same labels as in Shermer's proof.

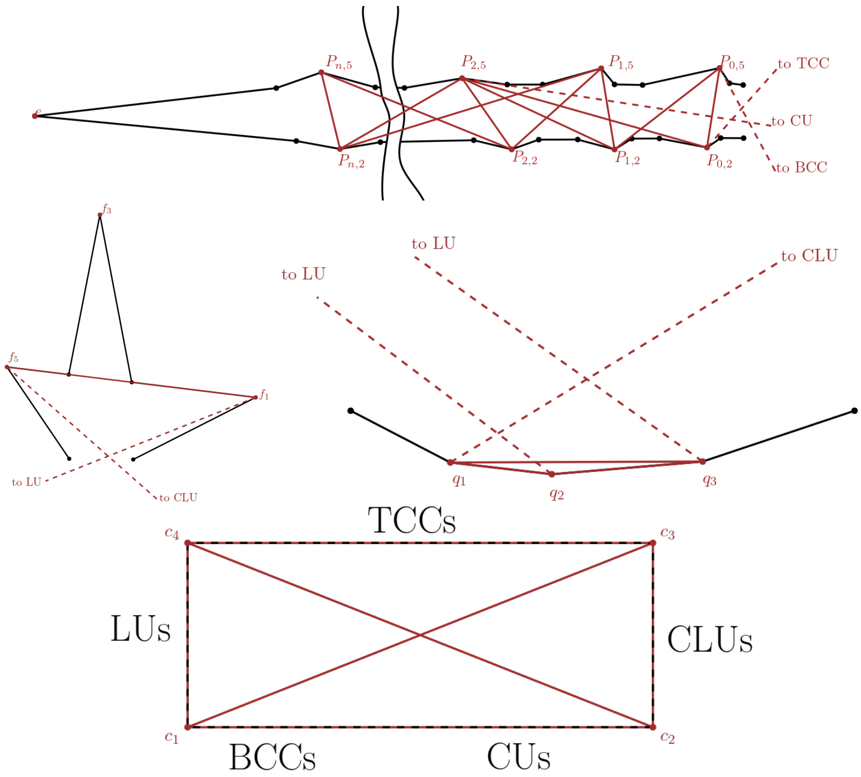

Figure 9 The subgraphs in each of the shermer construction components. Top is a literal unit, left is a consistency checker, right is a clause unit and bottom is the overall structure, indicating we are including the corners.

Just as in the previous examples, we start by finding a large clique in the complement graph since the chromatic number of the complement must be at least the size of that clique. A clique in the complement graph is an independent set in the graph, so we will select a set of points identified in Figure 8 such that no two share an edge.

For each literal unit, we take all the e points and the $P_{i,2}$ points for $0 \leq i \leq n$, amounting to $2mn + 4n$ points. For each consistency checker, we take all the f for an additional $2n$ points. For each clause checker, we take all the q_0 points for an additional m points. And from the overall box structure, we take c_1. The total size of this clique is $2mn + 6n + m + 1$, meaning we will have to show that an additional $2n$ colors are needed to color the complement graph.

For a start, we can label c_2, c_3, c_4 the same color as c_1 without any issue. For each $P_{i,5}$ in each literal unit, we can label $P_{i,5}$ with the same color. For each clause unit, q_2, q_3, can be labeled with the same color as the q_1. This leaves only the f_1, f_5 from each literal unit uncolored. The f_1 can be colored the same as the $P_{0,2}$ (or $P_{0,5}$ if its a bottom clause unit), but if we colored $P_{0,5}$ ($P_{0,2}$ for bottom) the same as $P_{0,2}$ ($P_{0,5}$ for bottom), then we can't. The same applies for f_5, but with the complement literal units (CLUs). However, f_1 and f_5 can be colored with the same color. Therefore we need only an additional $2n$ colors to color f_1 and f_5. Therefore, the chromatic number of the complement of the subgraph is $2mn + 6n + m + 1 = k$

Since the chromatic number of the complement of an induced subgraph of the PVG of the construction is equal to k, this means that the convex cover of the construction must be at least k. Since Shermer shows such a cover, this implies that the convex cover number of the construction is exactly k.

Since the convex cover number of the Shermer construction I' for any 3-SAT instance I is equal to k, and the hidden set number of I' is strictly less than k if and only if I is unsatisfiable, 3-SAT reduces to deciding if a simple polygon is a homestead polygon or not. Thus, since 3-SAT is NP-hard, deciding if a simple polygon is a homestead polygon or not must also be NP-hard.

◀

▶ **Corollary 13.** *Deciding if a polygon with or without holes is a homestead polygon or not is NP-hard.*

Proof. Since deciding if a simple polygon is a homestead polygon is NP-hard, this naturally implies that the more general case of a polygon with or without holes is also NP-hard. ◀

4 Conclusion

Polygon Subclass	Homesteadness	Deciding Homesteadness
Polygons with or without holes	Some	NP-hard
Simple polygons	Some	NP-hard
Star-shaped polygons	Some	?
Monotone polygons	Some	?
Orthogonal polygons	Some	?
Histogram polygons	All	Always true, $O(n)$ for values
Spiral polygons	All	Always true, $O(n)$ for values
Convex polygons	All	Always true, $O(1)$ for values

Table 1 Summary of results. (Note: "values" refers to finding either a maximum hidden set or minimum convex cover)

We presented two subclasses of polygons whose members are all homestead polygons and for which a maximum hidden set and a minimum convex cover can be found in $O(n)$ time. We also presented several subclasses for which there exists example polygons that are not homestead polygons. Lastly, we showed that for simple polygons, deciding if a polygon is a homestead polygon or not is NP-hard.

Additionally, we would like to pose the following open problems regarding maximum hidden set and minimum convex cover. We suspect that $cc(P) = O(hs(P))$, as all of the examples that we have observed have $cc(P) \leq 3/2hs(P)$. We also suspect that monotone mountains are homestead polygons and have a large part of the proof of this complete.

Finally, we believe that it is possible that finding the hidden set number and convex cover number when restricted to homestead polygons is no longer NP-hard and that there exists some value analogous to the Lovasv number which can be computed in polynomial time.

References

1 Mikkel Abrahamsen. Covering polygons is even harder. *CoRR*, abs/2106.02335, 2021. URL: `https://arxiv.org/abs/2106.02335`, `arXiv:2106.02335`.

2 Antonio Leslie Bajuelos Domínguez, Santiago Canales Cano, Gregorio Hernández Peñalver, and Ana Mafalda Martins. Escondiendo puntos en espirales e histogramas. In *Actas de las VI Jornadas de Matemática Discreta y Algorítmica, JMDA'08*, España, 2008. Universidad de Lleida. URL: `https://oa.upm.es/4564/`.

3 Stephan Eidenbenz. Inapproximability of finding maximum hidden sets on polygons and terrains, 2002.

4 Frederick S. Gella and Rosalio G. Artes. Clique cover of graphs. *Applied mathematical sciences*, 8:4301–4307, 2014.

5 Martin Grötschel, Lovász László, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 06 1981. `doi:10.1007/BF02579273`.

6 Hamid Hoorfar and Alireza Bagheri. Minimum hidden guarding of histogram polygons, 2017. `arXiv:1708.05815`.

7 Donald E. Knuth. The sandwich theorem. 1993. URL: `https://arxiv.org/abs/math/9312214`, `doi:10.48550/ARXIV.MATH/9312214`.

8 J. O'Rourke and K. Supowit. Some np-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–190, 1983. `doi:10.1109/TIT.1983.1056648`.

9 T. Pavlidis. Analysis of set patterns. *Pattern Recognition*, 1(2):165–178, 1968. URL: `https://www.sciencedirect.com/science/article/pii/003132036890006X`, `doi:https://doi.org/10.1016/0031-3203(68)90006-X`.

10 T. Shermer. Hiding people in polygons. *Computing*, 42(2-3):109–131, 1989.

A Algorithm 2 Proof of Correctness

▶ **Theorem 14.** *For any histogram polygon P, $hs(P) = cc(P)$.*

Proof.

▶ **Lemma 15.** *Algorithm 2 produces a hidden set.*

Proof. Assume that the first elements in each pair do not comprise a hidden set, H. This implies there are two vertices, $v1$ and $v2$, in H which see each other.

Case 1: $v1.y \neq v2.y$

Without loss of generality, we can assume $v1.y < v2.y$. From our algorithm, every point in H must be the midpoint of some horizontal edge in M which is not the base. Since every vertical line passing through the base can only pass through one more boundary (definition of monotonicity), we know that the region above each midpoint must be outside the polygon. Therefore, any line segment from $v1$ that connects to a point above $v1$ must go outside the polygon, which means $v1$ cannot see $v2$.

Case 2: $v1.y = v2.y$

Without loss of generality, we can assume $v1$ was added to H before $v2$. Since $v2$ was added into H, we know that $v1$ could not have been in the stack when $v2$ was being processed. Therefore a midpoint lower than $v1$ must've been added to H after $v1$ was added and before

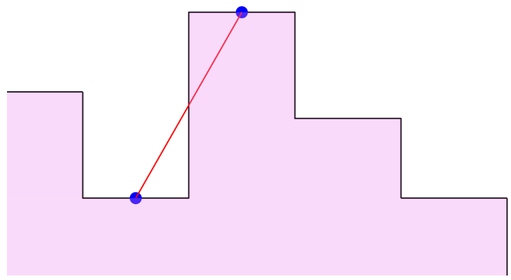

Figure 10 Example of Case 1 for Lemma 15.

$v2$ was added which popped $v1$ from the stack. The line segment $v1v2$ passes over this midpoint, which means $v1v2$ leaves the interior of the polygon, hence $v1$ does not see $v2$. ◀

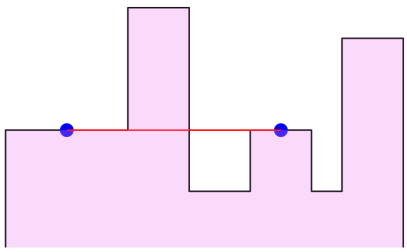

Figure 11 Example of Case 2 for Lemma 15.

Thus, since no two points in our hidden set see each other, Lemma 15 holds.

▶ **Lemma 16.** *Algorithm 2 produces a convex cover.*

Proof. Assume that the second elements in each pair do not comprise a convex cover, C. Since every element of C must be a rectangle since each vertex in each rectangle shares one coordinate with each of its neighbors and there are no 3 collinear points in each, we know that every piece of C must be convex. Therefore, according to our assumption, C must not be a cover.

Case 1: A point not in P is covered

Our algorithm considers every horizontal edge and constructs a rectangle down to the base. If any of these individual rectangles covers some point that is not part of the polygon, a vertical line through that point shows that it is not monotone, since it will have entered and left the polygon more than once.

If a point is covered by a rectangle obtained from merging, then this point must lie above some horizontal edge in between the left and right sides of the rectangle. This would imply there was some edge processed between two of the rectangles in the merge that was shorter than both, but if this was the case, the earlier of the two individual rectangles from the merge would've been popped from the stack before processing the second one, therefore it would not be a candidate for merging. Therefore, our algorithm does not cover any region outside of M.

Case 2: A point in P is not covered

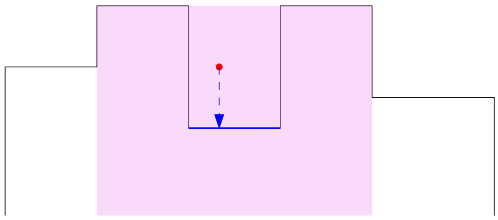

Figure 12 Example of Case 1 for Lemma A.

The only other condition is that the union must cover all regions. Assume there is some point which is not covered by any of the individual rectangles. Since M is monotone with respect to the horizontal, we know that a vertical line through each point must pass through a horizontal edge or through one of its endpoints. This vertical line must be a subset of a rectangle by definition. Since the merged rectangles do not get rid of any section, there cannot be any points that are not covered. Therefore, Lemma 14 holds by contradiction. ◀

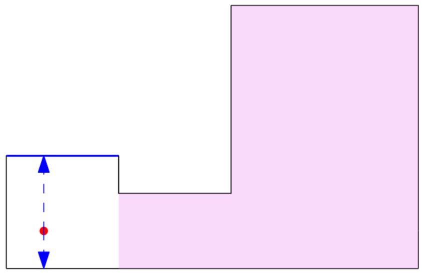

Figure 13 Example of Case 2 for Lemma A.

Since we know that Algorithm 2 produces a hidden set, H, and a convex cover, C, we can construct the following inequality.

$$|H| \leq hs(M) \leq cc(M) \leq |C|$$

and since $|H| = |C|$, this inequality collapses giving us:

$$hs(M) = cc(M) = |C|$$

Thus proving our theorem, for any histogram polygon M, the hidden set number and convex cover number are equal, making M a homestead by definition. ◀

B Algorithm 2 Proof of Optimality

▶ **Theorem 17.** *Algorithm 2 has optimal worst-case time complexity for both maximum hidden set and minimum convex cover.*

An algorithm has optimal worst-case time complexity if there is some lower bound on worst-case time complexity for that problem which the time complexity of that algorithm is asymptotically equivalent to.

▶ **Lemma 18.** *Finding the maximum hidden set in a histogram polygon takes* $\Omega(n)$ *time complexity.*

Proof. Shermer shows that for orthogonal polygons, the size of the maximum hidden set can be as large as, but not exceed, $(n-2)/2$[10]. His example of orthogonal staircases are monotone mountains, which means that this tight bound also applies to histogram polygons. Simply reporting $(n-2)/2$ hidden points takes $\Theta(n)$ time, therefore a lower bound for any

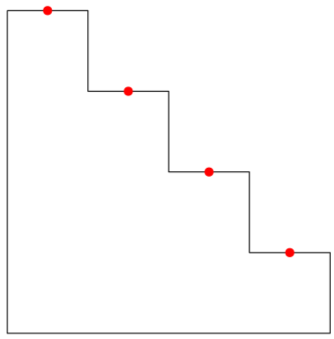

Figure 14 Shermer's Staircase example, $n = 10$.

algorithm that reports maximum hidden points for histogram polygons must have worst-case time complexity $\Omega(n)$. Thus Lemma B holds. ◀

▶ **Lemma 19.** *Finding the minimum convex cover in a histogram polygon takes* $\Omega(n)$ *time complexity.*

Since we know that there is at most $(n-2)/2$ hidden points in a histogram polygon and that this is a tight bound, there must also be some histogram polygon for which at least $(n-2)/2$ convex pieces are required to cover it. This follows from the fact that $hs(M) \leq cc(M)$.

Simply reporting at least $(n-2)/2$ convex pieces would take $\Omega(n)$ time, thus any algorithm that reports the minimum convex cover for histogram polygons must have worst-case time complexity $\Omega(n)$. Thus Lemma B holds.

▶ **Lemma 20.** *Algorithm 2 has* $O(n)$ *worst-case time complexity.*

Proof. All operations before the while loop can be completed in $O(n)$.

The rotation can be done as follows: first find the two points on the base by finding the maximum and minimum points in the direction of monotonicity. If we don't already know the direction of monotonocity, we can check this in linear time with a march around M to determine whether the direction of the vertical edges increases/decreases monotonically (allowing for one exception, the base) or if its the horizontal. If both work choose the horizontal. After finding these points, rotate it so that the orientation of their edge points to the right. Apply this rotation to all of M. This all runs in $O(n)$ time.

Since we already found the base, finding p is $O(1)$ and so is instantiating the set and the stack. Thus the total of the first section is $O(n)$ worst-case.

Our outer while loop iterates over every horizontal edge in M except the base, which means it runs $(n-2)/2$ times. All operations excluding the inner while loop are $O(1)$,

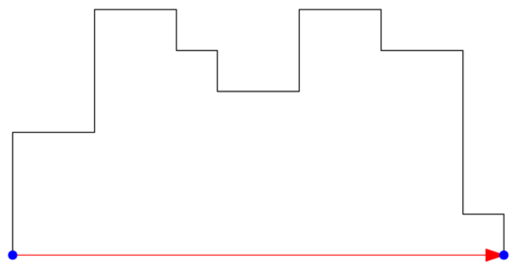

Figure 15 Example of a histogram polygon after rotation.

therefore excluding the inner loop, the outer while loop contributes $(n-2)/2 * O(1) = O(n)$ to time complexity.

For the inner while loop, we use the same argument as the Graham Scan. The inner while loop can iterate at most $(n-2)/2 - 1$ times if we need to pop every pair from the stack when considering the last edge. However, once an element is popped from the stack it is either put back on by that edge, which that edge can only do to 1 element besides its own pair, or it is discarded. Therefore we can assign all discarded pops with the edges of their initial pushes and any reinsertions with the edge that reinserted it. This results in $O(n)$ time complexity from the stack operations, which means across all iterations the inner loop contributes $O(n)$ to time complexity.

The total time complexity $O(n) + O(n) + O(n) = O(n)$, therefore Lemma B holds. ◀

Since both problems have lower bounds of $\Omega(n)$ for their worst-case time complexities and our algorithm runs in $O(n)$, we know that our algorithm must be optimal, thus proving our final theorem:

▶ **Theorem 21.** *For any histogram polygon P, Algorithm 2 finds both a maximum hidden set and minimum convex cover in optimal $(O(n))$ time.*

C Aside on convex cover and clique cover

A point which we have made but which we haven't found anywhere else in the literature is that the minimum convex cover of a polygon P is the same as minimum clique cover on the point visibility graph. Traditionally, a convex cover of a polygon P is a set of convex shapes whose union is P. It differs from the convex decomposition in that overlaps are allowed. Clique cover is traditionally a partition of a graph into cliques. For any convex cover, we can assign the overlaps to one of the pieces in the overlap, which would maintain that each piece is a clique in the PVG and it would be a clean partition, such as in Figure 6. The PVG does not require cliques to be geometrically contiguous, so these remain cliques. Therefore it follows that convex cover and clique cover are equivalent notions.

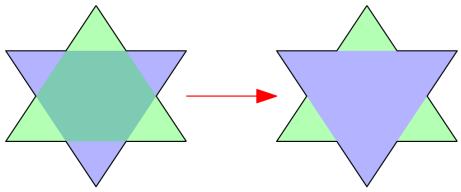

Figure 16 Showing that we can assign intersections to specific cliques to be a "partition" of the PVG, such as for the Star of David.