Convex partial transversals of planar regions

Vahideh Keikha
Dept. of Mathematics and Computer Science, University of Sistan and Baluchestan, Zahedan, Iran
va.keikha@gmail.com

Mees van de Kerkhof[1]
Dept. of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
m.a.vandekerkhof@uu.nl

Marc van Kreveld[2]
Dept. of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
m.j.vankreveld@uu.nl

Irina Kostitsyna
Dept. of Mathematics and Computer Science, TU Eindhoven,
Eindhoven, The Netherlands
i.kostistyna@tue.nl

Maarten Löffler[3]
Dept. of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
m.loffler@uu.nl

Frank Staals[4]
Dept. of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
f.staals@uu.nl

Jérôme Urhausen[5]
Dept. of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
j.e.urhausen@uu.nl

Jordi L. Vermeulen[6]
Dept. of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
j.l.vermeulen@uu.nl

Lionov Wiratma[7]
Dept. of Information and Computing Sciences, Utrecht University
Utrecht, The Netherlands
Dept. of Informatics, Parahyangan Catholic University
Bandung, Indonesia
l.wiratma@uu.nl;lionov@unpar.ac.id

---

[1] M.v.d.K. supported by the Netherlands Organisation for Scientific Research under proj. 628.011.005.
[2] M.v.K. supported by the Netherlands Organisation for Scientific Research under proj. 612.001.651.
[3] M.L. supported by the Netherlands Organisation for Scientific Research under proj. 614.001.504.
[4] F.S. supported by the Netherlands Organisation for Scientific Research under proj. 612.001.651.
[5] J.U. supported by the Netherlands Organisation for Scientific Research under proj. 612.001.651.
[6] J.V. supported by the Netherlands Organisation for Scientific Research under proj. 612.001.651.
[7] L.W. supported by the Mnst. of Research, Techn. and High. Ed. of Indonesia (No. 138.41/E4.4/2015)

29th International Symposium on Algorithms and Computation (ISAAC 2018).
Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 187; pp. 187:1–187:12
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

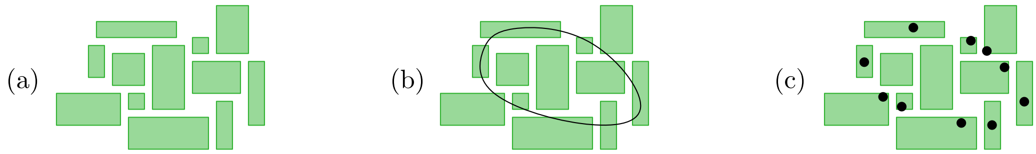


Figure 1 (a) A set of 12 regions. (b, c) A convex partial transversal of size 10.

Abstract

We consider the problem of testing, for a given set of planar regions $\mathcal{R}$ and an integer k, whether there exists a convex shape whose boundary intersects at least k regions of $\mathcal{R}$. We provide polynomial-time algorithms for the case where the regions are disjoint axis-aligned rectangles or disjoint line segments with a constant number of orientations. On the other hand, we show that the problem is NP-hard when the regions are intersecting axis-aligned rectangles or 3-oriented line segments. For several natural intermediate classes of shapes (arbitrary disjoint segments, intersecting 2-oriented segments) the problem remains open.

2012 ACM Subject Classification Theory of computation → Computational Geometry

Keywords and phrases computational geometry, algorithms, NP-hardness, convex transversals

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.187

1 Introduction

A set of points Q in the plane is said to be in *convex position* if for every point $q \in Q$ there is a halfplane containing Q that has q on its boundary. Now, let $\mathcal{R}$ be a set of n *regions* in the plane. We say that Q is a *partial transversal* of $\mathcal{R}$ if there exists an injective map $f : Q \to \mathcal{R}$ such that $q \in f(q)$ for all $q \in Q$; if f is a bijection we call Q a *full transversal*. In this paper, we are concerned with the question whether a given set of regions $\mathcal{R}$ admits a convex partial transversal Q of a given cardinality $|Q| = k$. Figure 1 shows an example.

The study of convex transversals was initiated by Arik Tamir at the Fourth NYU Computational Geometry Day in 1987, who asked "Given a collection of compact sets, can one decide in polynomial time whether there exists a convex body whose boundary intersects every set in the collection?" Note that this is equivalent to the question of whether a convex full transversal of the sets exists: given the convex body, we can place a point of its boundary in every intersected region; conversely, the convex hull of a convex transversal forms a convex body whose boundary intersects every set. In 2010, Arkin et al. [2] answered Tamir's original question in the negative (assuming P $\neq$ NP): they prove that the problem is NP-hard, even when the regions are (potentially intersecting) line segments in the plane, regular polygons in the plane, or balls in $\mathbb{R}^3$. On the other hand, they show that Tamir's problem can be solved in polynomial time when the regions are *disjoint* segments in the plane and the convex body is restricted to be a polygon whose vertices are chosen from a given discrete set of (polynomially many) candidate locations. Goodrich and Snoeyink [6] show that for a set of *parallel* line segments, the existence of a convex transversal can be tested in $O(n \log n)$ time. Schlipf [13] further proves that the problem of finding a convex stabber for a set of disjoint *bends* (that is, shapes consisting of two segments joined at one endpoint) is also NP-hard. She also studies the optimisation version of maximising the number of regions stabbed by a convex shape; we may re-interpret this question as finding the largest k such

Table 1 New and known results. The arrows indicate that one result is implied by another.

		disjoint	intersecting
line segments:	parallel	$O(n^6)$ (upper hull only: $O(n^2)$)	N/A
	2-oriented	$\downarrow$	open
	3-oriented	$\downarrow$	NP-hard
	ρ-oriented	polynomial	$\uparrow$
	arbitrary	open	NP-hard [2]
rectangles:	squares	$\downarrow$	open
	rectangles	polynomial	NP-hard
other:	bends	NP-hard [13]	$\leftarrow$

that a convex partial transversal of cardinality k exists. She shows that this problem is also NP-hard for a set of (potentially intersecting) line segments in the plane.

Related work. Computing a partial transversal of maximum size arises in wire layout applications [14]. When each region in $\mathcal{R}$ is a single point, our problem reduces to determining whether a point set P has a subset of cardinality k in convex position. Eppstein et al. [4] solve this in $O(kn^3)$ time and $O(kn^2)$ space using dynamic programming; the total number of convex k-gons can also be tabulated in $O(kn^3)$ time [12, 10].

If we allow reusing elements, our problem becomes equivalent to so-called *covering color classes* introduced by Arkin et al. [1]. Arkin et al. show that for a set of regions $\mathcal{R}$ where each region is a set of two or three points, computing a convex partial transversal of $\mathcal{R}$ of maximum cardinality is NP-hard. Conflict-free coloring has been studied extensively, and has applications in, for instance, cellular networks [5, 7, 8].

Results. Despite the large body of work on convex transversals and natural extensions of partial transversals that are often mentioned in the literature, surprisingly, no positive results were known. We present the first positive results: in Section 2 we show how to test whether a set of parallel line segments admits a convex transversal of size k in polynomial time; we extend this result to disjoint segments of a fixed number of orientations and to disjoint axis-aligned rectangles in Section 3. Although the hardness proofs of Arkin et al. and Schlipf do extend to partial convex transversals, we strengthen these results by showing that the problem is already hard when the regions are 3-oriented segments or axis-aligned rectangles (Section 4). Our results are summarized in Table 1.

For ease of terminology, in the remainder of this paper, we will drop the qualifier "partial" and simply use "convex transversal" to mean "partial convex transversal". Also, for ease of argument, in all our results we test for *weakly convex* transversals. This means that the transversal may contain three or more colinear points. Missing proofs can be found in the full version of this paper [9].

2 Parallel disjoint line segments

Let $\mathcal{R}$ be a set of n vertical line segments in $\mathbb{R}^2$. We assume that no three endpoints are aligned. Let $\Uparrow\mathcal{R}$ and $\Downarrow\mathcal{R}$ denote the sets of upper and lower endpoints of the regions in $\mathcal{R}$, respectively, and let $\Updownarrow\mathcal{R} = \Uparrow\mathcal{R} \cup \Downarrow\mathcal{R}$. In Section 2.1 we focus on computing an *upper convex transversal* –a convex transversal Q in which all points appear on the upper hull of Q– that maximizes the number of regions visited. We show that there is an optimal transversal whose strictly convex vertices lie only on bottom endpoints in $\Downarrow\mathcal{R}$. This allows us to develop a dynamic programming algorithm that computes such an optimal upper convex transversal in $O(n^2)$ time. In Section 2.2 we prove that there exists an optimal convex transversal whose strictly convex vertices are taken from the set of all endpoints $\Updownarrow\mathcal{R}$, and whose leftmost and rightmost vertices are taken from a discrete set of points. This leads to an $O(n^6)$ time dynamic programming algorithm to compute such a transversal.

2.1 Computing an upper convex transversal

Let k^* be the maximum number of regions visitable by an upper convex transversal of $\mathcal{R}$.

▶ **Lemma 1.** *Let U be an upper convex transversal of $\mathcal{R}$ that visits k regions. There exists an upper convex transversal U' of $\mathcal{R}$, that visits the same k regions as U, and such that the leftmost vertex, the rightmost vertex, and all strictly convex vertices of U' lie on the bottom endpoints of the regions in $\mathcal{R}$.*

Proof. Let $\mathcal{U}$ be the set of all upper convex transversals with k vertices. Let $U' \in \mathcal{U}$ be an upper convex transversal such that the sum of the y-coordinates of its vertices is minimal. Assume, by contradiction, that U' has a vertex v that is neither on the lower endpoint of its respective segment nor aligned with its adjacent vertices. Then we can move v down without making the upper hull non-convex. This is a contradiction. Therefore, all vertices in U' are either aligned with their neighbors (and thus not strictly convex), or at the bottom endpoint of a region. ◀

Let $\Lambda(v, w)$ denote the set of bottom endpoints of regions in $\mathcal{R}$ that lie left of v and below the line through v and w. See Fig. 2(a). Let $slope(\overline{uv})$ denote the slope of the supporting line of $\overline{uv}$, and observe that $slope(\overline{uv}) = slope(\overline{vu})$.

By Lemma 1 there is an optimal upper convex transversal of $\mathcal{R}$ in which all strictly convex vertices lie on bottom endpoints of the segments. Let $K[v, w]$ be the maximum number of regions visitable by an upper convex transversal that ends at a bottom endpoint v, and has an incoming slope at v of *at least* $slope(\overline{vw})$. Note that the second argument w is used only to specify the slope, and w may be left or right of v. We have that

$$K[v, w] = \max_{u \in \Lambda(v,w)} \max_{s \in \Lambda(u,v)} K[u, s] + I[u, v],$$

where $I[u, v]$ denotes the number of regions in $\mathcal{R}$ intersected by the segment $\overline{uv}$ (in which we treat the endpoint at u as open, and the endpoint at v as closed). See Fig. 2(a).

▶ **Observation** 2. Let v, s, and t be bottom endpoints of segments in $\mathcal{R}$ with $slope(\overline{sv}) > slope(\overline{tv})$. We have that $K[v, t] \geq K[v, s]$.

Fix a bottom endpoint v, and order the other bottom endpoints $w \in \Downarrow\mathcal{R}$ in decreasing order of slope $slope(\overline{wv})$. Let S_v denote the resulting order.

▶ **Lemma 3.** *Let v and w be bottom endpoints of regions in $\mathcal{R}$, and let u be the predecessor of w in S_v, if it exists (otherwise let $K[v, u] = -\infty$). We have that*

$$K[v, w] = \begin{cases} \max\{1, K[v, u], K[w, v] + I[w, v]\} & \textit{if } w_x < v_x\,, \\ \max\{1, K[v, u]\} & \textit{otherwise.} \end{cases}$$

Where v_x denotes the x-coordinate of a point v. Lemma 3 now suggests a dynamic programming approach to compute the $K[v, w]$ values for all pairs of bottom endpoints v, w: we process the endpoints v on increasing x-coordinate, and for each v, we compute all

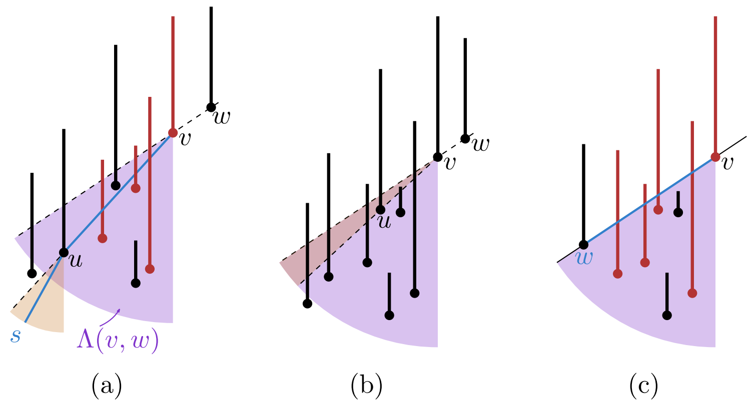


Figure 2 (a) The definition of $K[v,w]$. The region $\Lambda(v,w)$ is indicated in purple. The segments counted in $I[u,v]$ are shown in red. (b) The case that $K[v,w] = K[v,u]$, where u corresponds to the predecessor slope of $slope(\overline{vw})$. (c) The case that $K[v,w] = K[w,v] + I[w,v]$.

$K[v,w]$ values in the order of S_v. To this end, we need to compute (i) the (radial) orders S_v, for all bottom endpoints v, and (ii) the number of regions intersected by a line segment $\overline{uv}$, for all pairs of bottom endpoints u, v. We show that we can solve both these problems in $O(n^2)$ time. We then also obtain an $O(n^2)$ time algorithm to compute $k^* = \max_{v,w} K[v,w]$.

Computing predecessor slopes. For each bottom endpoint v, we simply sort the other bottom endpoints around v. This can be done in $O(n^2)$ time in total [11][8]. We can obtain S_v by splitting the resulting list into two lists, one with all endpoints left of v and one with the endpoints right of v, and merging these lists appropriately. This takes $O(n^2)$ time.

Computing the number of intersections. We use the standard duality transform [3] to map every point $p = (p_x, p_y)$ to a line $p^* : y = p_x x - p_y$, and every non-vertical line $\ell : y = ax + b$ to a point $\ell^* = (a, -b)$. Consider the arrangement $\mathcal{A}$ formed by the lines p^* dual to all endpoints p (both top and bottom) of all regions in $\mathcal{R}$. Observe that all our query segments $\overline{uv}$ with $u_x < v_x$ are defined by two bottom endpoints u and v, so the supporting line ℓ_{uv} of such a segment corresponds to a vertex ℓ^*_{uv} of the arrangement $\mathcal{A}$.

In the dual space, a vertical line segment $R = \overline{pq} \in \mathcal{R}$ corresponds to a strip R^* bounded by two parallel lines p^* and q^*. Let $\mathcal{R}^*$ denote this set of strips corresponding to $\mathcal{R}$. It follows that if we want to count the number of regions of $\mathcal{R}$ intersected by a query segment $\overline{uv}$ on line ℓ we have to count the number of strips in $\mathcal{R}^*$ containing the point ℓ^* and whose slope $slope(R)$ lies in the range $[u_x, v_x]$. See Fig. 3 for an illustration.

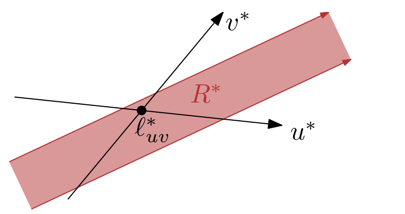


Figure 3 The strip R^* with a slope in the range $[u_x, v_x]$ containing ℓ^*_{uv} contributes one to $T_{u^*}(\ell^*_{uv})$ and zero to $T_{v^*}(\ell^*_{uv})$.

▶ **Observation 4.** Let p^* be a line, oriented from left to right, and let R^* be a strip. The line p^* intersects the bottom boundary of R^* before the top boundary of R^* if and only if $slope(p^*) > slope(R^*)$.

[8] Alternatively, we can dualize the points into lines and use the dual arrangement to obtain all radial orders in $O(n^2)$ time.

Consider traversing a line p^* of $\mathcal{A}$ (from left to right), and let $T_{p^*}(\ell^*)$ be the number of strips that contain the point ℓ^* and that we enter through the top boundary of the strip.

▶ **Lemma 5.** *Let ℓ^*_{uv}, with $u_x < v_x$, be a vertex of $\mathcal{A}$. The number of strips from $\mathcal{R}^*$ containing ℓ^*_{uv} with a slope in $[u_x, v_x]$ is $T_{u^*}(\ell^*_{uv}) - T_{v^*}(\ell^*_{uv})$.*

▶ **Corollary 6.** *Let $u, v \in \Downarrow\mathcal{R}$ be bottom endpoints. The number of regions of $\mathcal{R}$ intersected by $\overline{uv}$ is $T_{u^*}(\ell^*_{uv}) - T_{v^*}(\ell^*_{uv})$.*

We can easily compute the counts $T_{u^*}(\ell^*_{uv})$ for every vertex ℓ^*_{uv} on u^* by traversing the line u^*. Thus, we can compute the number of regions in $\mathcal{R}$ intersected by $\overline{uv}$, for all bottom endpoints u and v in a total of $O(n^2)$ time.

Together with our dynamic programming approach for computing k^* we then get:

▶ **Theorem 7.** *Given a set of n vertical line segments $\mathcal{R}$, we can compute the maximum number of regions k^* visitable by an upper convex transversal Q in $O(n^2)$ time.*

2.2 Computing a convex transversal

We now consider computing a convex transversal that maximizes the number of regions visited. We first prove some properties of an optimal convex transversal. We then use these properties to compute the maximum number of regions visitable by such a transversal using dynamic programming.

Canonical Transversals Like in the case of the upper hull, we first argue that we can discretize the problem. Similar to Lemma 1 we can argue that the strictly convex vertices in the upper and lower hulls must lie on endpoints of the segments in $\mathcal{R}$. We can then show that the leftmost and rightmost vertex must lie on the intersection point of a segment with a line that goes through at least two endpoints. Next, we give a more precise characterization of the type of transversals we have to consider.

A convex transversal Q' of $\mathcal{R}$ is a *lower canonical* transversal if and only if

- the strictly convex vertices on the upper hull of Q' lie on bottom endpoints in $\mathcal{R}$,
- the strictly convex vertices on the lower hull of Q' lie on bottom or top endpoints of regions in $\mathcal{R}$,
- the leftmost vertex ℓ of Q' lies on a line through w, where w is the leftmost strictly convex vertex of the lower hull of Q', and another endpoint.
- the rightmost vertex r of Q' lies on a line through z, where z is the rightmost strictly convex vertex of the lower hull of Q', and another endpoint.

Let $Q = \ell urv$ be a quadrilateral whose leftmost vertex is ℓ, whose *top* vertex is u, whose rightmost vertex is r, and whose *bottom* vertex is v. A quadrilateral Q is a *lower canonical* quadrilateral if and only if

- u and v lie on endpoints in $\Updownarrow\mathcal{R}$,
- ℓ lies on a line through v and another endpoint, and
- r lies on a line through v and another endpoint.

We define an *upper canonical transversal*, and an *upper canonical quadrilateral* analogously. In this case the points ℓ and r are defined by points on the upper hull.

Let k^u_2 be the maximal number of regions of $\mathcal{R}$ visitable by an upper convex transversal, let k^u_4 be the maximal number of regions of $\mathcal{R}$ visitable by a canonical upper quadrilateral, and let k^u denote the maximal number of regions of $\mathcal{R}$ visitable by a canonical upper transversal. We define k^b_2, k^b_4, and k^b, for the maximal number of regions of $\mathcal{R}$, visitable by a lower convex transversal, canonical lower quadrilateral, and canonical lower transversal, respectively.

▶ **Lemma 8.** *Let k^* be the maximal number of regions in $\mathcal{R}$ visitable by a convex transversal of $\mathcal{R}$. We have that $k^* = \max\{k_2^u, k_4^u, k^u, k_2^b, k_4^b, k^b\}$.*

By Lemma 8 we can restrict our attention to upper and lower convex transversals, canonical quadrilaterals, and canonical transversals. We can compute an optimal upper (lower) convex transversal in $O(n^2)$ time using the algorithm from the previous section. We now argue that we can compute an optimal canonical quadrilateral in $O(n^5)$ time, and an optimal canonical transversal in $O(n^6)$ time. Arkin et al. [2] describe an algorithm that given a discrete set of vertex locations can find a convex polygon (on these locations) that maximizes the number of regions stabbed. Note, however, that since a region contains multiple vertex locations —and we may use only one of them— we cannot directly apply their algorithm.

Computing the maximal number of regions intersected by a canonical quadrilateral Consider a canonical lower quadrilateral $Q = \ell urw$ with $u_x < w_x$. We explicitly compute the regions intersected by $\overline{u\ell} \cup \overline{\ell w}$ and set these aside. Using a rotational sweep we then compute how many of the remaining regions intersect $\overline{ur} \cup \overline{wr}$, for all candidate points r, and find the candidate point r that maximizes the total number of regions intersected by Q. If $u_x > w_x$, we use a symmetric procedure in which we count all regions intersected by $\overline{ur} \cup \overline{rw}$ first, and then the remaining regions intersected by $\overline{u\ell} \cup \overline{\ell w}$.

Since there are $O(n^4)$ candidate triples u, w, ℓ, naively computing the maximum as sketched above requires $O(n^6)$ time. We argue that we do not have to do this rotational sweep for every such triple. This reduces the running time to $O(n^5)$.

▶ **Lemma 9.** *Given a set of n vertical line segments $\mathcal{R}$, we can compute the maximum number of regions k^* visitable by a canonical quadrilateral Q in $O(n^5)$ time.*

Computing the maximal number of regions intersected by a canonical transversal We describe an algorithm to compute the maximal number of regions visitable by a lower canonical convex transversal. Our algorithm consists of three dynamic programming phases, in which we consider (partial) convex hulls of a particular "shape".

In the first phase we compute (and memorize) the maximal number of regions $B[w, u, v, \ell]$ visitable by a transversal that has $\overline{w\ell}$ as a segment in the lower hull, and a convex chain $\ell, \dots, u, v$ as upper hull. See Fig. 4(a).

In the second phase we compute the maximal number of regions $K[u, v, w, z]$ visitable by the canonical convex transversal whose rightmost top edge is $\overline{uv}$ and whose rightmost bottom edge is $\overline{wz}$. See Fig. 4(b) and (c). To make sure that we appropriately count segments that intersect both the upper and lower hull we have to distinguish between two cases, depending on whether $u_x \leq w_x$ or vice versa. Furthermore, we use that for all pairs of candidate edges $\overline{wz}$ and $\overline{uv}$ we can precompute the number of segments $I[w, z, u, v]$ intersected by $\overline{wz}$ that are *not* intersected by $\overline{uv}$.

In the third phase we compute the maximal number of regions visitable when we "close" the transversal using the rightmost vertex r. To this end, we define $R'[z, u, v, r]$ as the number of regions visitable by the canonical transversal whose rightmost upper segment is $\overline{uv}$ and whose rightmost lower segment is $\overline{wz}$ and r is defined by the strictly convex vertex z.

▶ **Theorem 10.** *Given a set of n vertical line segments $\mathcal{R}$, we can compute the maximum number of regions k^* visitable by a convex transversal Q in $O(n^6)$ time.*

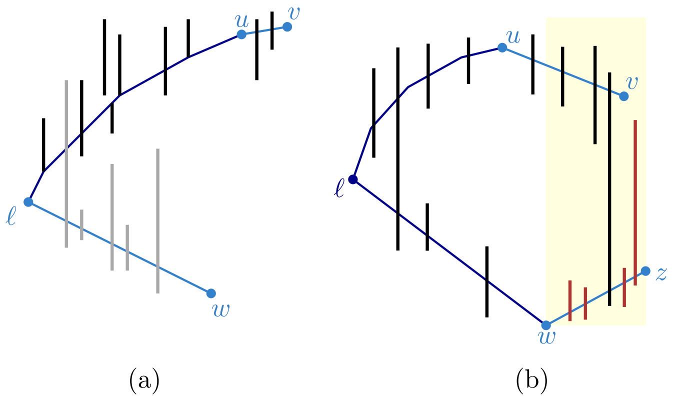


Figure 4 (a) $B[w,u,v,\ell]$ indicates the number of regions visited by a convex transversal that has $\overline{\ell w}$ as lower hull and the upper hull from ℓ to $\overline{uv}$. We can compute the $B[w,u,v,\ell]$ values for all u, v by explicitly setting aside the segments intersected by $\overline{\ell w}$ and then using the upper hull algorithm. (b) The base case of the recurrence when $u_x < w_x$. The regions counted by $I[w,z,u,v]$ are shown in red, whereas the regions counted by $B[w,u,v,\ell]$ are shown in black. (c) The inductive step when $u_x < w_x$.

3 2-oriented disjoint line segments

In this section we consider the case when $\mathcal{R}$ consists of vertical and horizontal disjoint segments. We will show how to apply similar ideas to those presented in the previous section to compute an optimal convex transversal Q of $\mathcal{R}$. As in the previous section, we will mostly restrict our search to canonical transversals. However, we will have one special case to consider when an optimal partial convex transversal has bends not necessarily belonging to a discrete set of points. In this section we will provide an overview of the ideas behind our approach; the reader is referred to the full version of this paper for the missing details [9].

We call the left-, right-, top- and bottommost vertices ℓ, r, u and b of Q the *extreme* vertices. They subdivide the transversal into four chains. Similarly to the 1-oriented case, we can move the non-extreme convex vertices to be on the endpoints of the segments (Lemma 11). In the 1-oriented case, the extreme vertices were restricted to being on intersections of lines through endpoints with segments in $\mathcal{R}$, which we will call a *1st-order fixed point*. For the 2-oriented case, we need to extend this notion: when one extreme vertex is on a 1st-order fixed point, the opposite extreme vertex might be on the intersection of a line through an endpoint and the 1st-order fixed point with a segment in $\mathcal{R}$ (these are *2nd-order fixed points*). The proof is analogous to that of Lemma 1.

▶ **Lemma 11.** *Let Q be a convex partial transversal of $\mathcal{R}$ with extreme vertices ℓ, r, t, and b. There exists a convex partial transversal Q' of $\mathcal{R}$ such that*

- *the two transversals have the same extreme vertices,*
- *all segments that are intersected by the upper-left, upper-right, lower-right, and lower-left hulls of Q are also intersected by the corresponding hulls of Q',*
- *all strictly convex vertices on the* ***upper-left*** *hull of Q' lie on* ***bottom*** *endpoints of vertical segments or on the* ***right*** *endpoints of horizontal segments of $\mathcal{R}$, and*
- *the convex vertices on the other hulls of Q' lie on analogous endpoints.*

Let Q be the maximum convex transversal. There are three cases to consider. (1) There exists a chain of the convex hull of Q containing at least two endpoints of segments, (2)

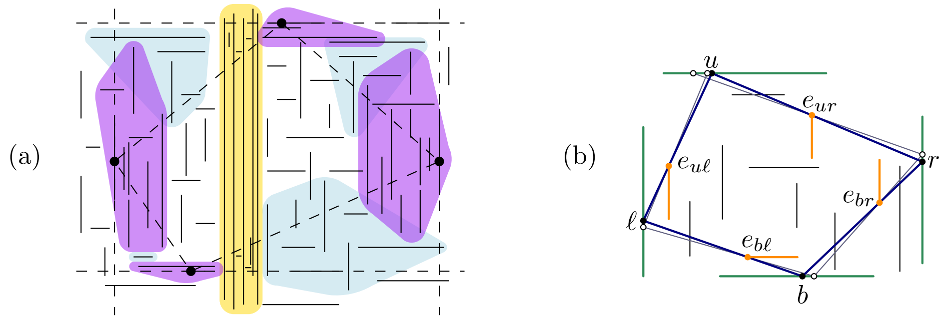


Figure 5 (a) The subdivision into cases for the canonical algorithm. (b) The construction in the special case.

there exists a chain of the convex hull of Q containing no endpoints, or (3) all four convex chains contain at most one endpoint. In case (1) we prove that one can move the endpoints around such that all points of the transversal are on a discrete set of points, allowing us to search for a canonical transversal (see below). In case (2) one can move the extreme points adjacent to that chain in such a way that the chain encounters an endpoint. In case (3) we can either move the points around such that one chain now contains two endpoints, putting us in case (1), or we are in the "special case" that is solved separately.

3.1 Calculating the canonical transversal

We subdivide our problem into subproblems (shown in Figure 5(a)) that can be solved using the algorithm for the 1-oriented case. We observe that if we fix the extreme vertices, we have a partial ordering of segments on each chain, defining the order in which they can be intersected. For each chain, we guess a point that will be a vertex. This gives us a subproblem for each extreme point: we need to find an "upper" and "lower" chain that links the extreme point to the guessed vertices. For this we can simply use the algorithm for the parallel case, except in the case where there are segments in $\mathcal{R}$ that could intersect two non-adjacent chains. We put such segments into a separate subproblem, of which there can be only one. We then need to examine all possible combinations of extreme points and guessed vertices, but as this is a constant number of points, and as we choose them out of a polynomial number of points, this gives a polynomial time algorithm. This algorithm extends to any constant number of orientations.

3.2 Special case

As mentioned above this case only occurs when the four hulls each contain exactly one endpoint. The construction can be seen in Figure 5(b). Let $e_{u\ell}$, e_{ur}, e_{br} and $e_{b\ell}$ be the endpoints on the upper-left, upper-right, lower-right and lower-left hull. Let further s_u, s_r, s_b and s_ℓ be the segments that contain the extreme points.

For two points a and b, let $l(a, b)$ be the line through a and b. For a given position of u we can place r on or below the line $l(u, e_{ur})$. Then we can place b on or left of the line $l(r, e_{br})$, ℓ on or above $l(b, e_{b\ell})$ and then test if u is on or to the right of $l(\ell, e_{u\ell})$. Placing r lower decreases the area where b can be placed and the same holds for the other extreme points. It follows that we place r on the intersection of $l(u, e_{ur})$ and s_r, we set $\{b\} = l(r, e_{br}) \cap s_b$ and $\{\ell\} = l(b, e_{b\ell}) \cap s_\ell$. Let then u' be the intersection of the line $l(\ell, e_{\ell u})$ and the upper segment s_u. In order to make the test if u' is left of u we first need the following lemma.

▶ **Lemma 12.** *Given a line ℓ, a point A, and a point $X(\tau)$ with coordinates $\left(\frac{P_1(\tau)}{Q(\tau)}, \frac{P_2(\tau)}{Q(\tau)}\right)$ where $P_1(\cdot)$, $P_2(\cdot)$, and $Q(\cdot)$ are linear functions. The intersection Y of ℓ and the line through the points X and A has coordinates $\left(\frac{P_1'(\tau)}{Q'(\tau)}, \frac{P_2'(\tau)}{Q'(\tau)}\right)$ where $P_1'(\cdot)$, $P_2'(\cdot)$ and $Q'(\cdot)$ are linear functions.*

Let (τ, c) be the coordinates of the point u for $\tau \in I$, where the constant c and the interval I are determined by the segment s_u. Then by Lemma 12 we have that the points r, b, ℓ, u' all have coordinates of the form specified in the lemma. First we have to check for which values of τ the point u is between $e_{u\ell}$ and e_{ur}, r is between e_{br} and e_{ur}, b is between $e_{b\ell}$ and e_{br} and ℓ is between $e_{b\ell}$ and $e_{u\ell}$. This results in a system of linear equations whose solution is an interval I'.

We then determine the values of $\tau \in I'$ where $u' = \left(\frac{P_1(\tau)}{Q(\tau)}, \frac{P_2(\tau)}{Q(\tau)}\right)$ is left of $u = (\tau, c)$ by considering the following quadratic inequality: $\frac{P_1(\tau)}{Q(\tau)} \leq \tau$. If there exists a τ satisfying all these constraints, then there exists a convex transversal such that the points u, r, b and ℓ are the top-, right-, bottom-, and leftmost points, and the points e_{jk} $(j, k = u, r, b, \ell)$ are the only endpoints contained in the hulls.

Combining this with the algorithm in the previous section, we obtain the following result:

▶ **Theorem 13.** *Given a set of 2-oriented line segments, we can compute the maximum number of regions visited by a convex partial transversal in polynomial time.*

Extensions. One should note that the concepts explained here generalize to more orientations. For each additional orientation there will be two more extreme points and therefore two more chains. It follows that for ρ orientations there might be ρth-order fixed points. This increases the running time, because more points need to be guessed and the pool of discrete points to choose from is bigger, but for a fixed number of orientations it is still polynomial in n. The special case generalizes as well, which means that the same case distinction can be used. We further know that when $\mathcal{R}$ is a set of non-intersecting connected regions, any transversal with size at least 2 intersects the boundary of each region containing a point of the transversal. It follows that the algorithm extends to disjoint convex polygons with limited edge orientations, e.g. disjoint axis-aligned rectangles.

4 3-oriented intersecting segments

We prove that the problem of finding a maximum convex partial transversal Q of a set of 3-oriented segments $\mathcal{R}$ is NP-hard using a reduction from Max-2-SAT.

▶ **Theorem 14.** *Let $\mathcal{R}$ be a set of segments that have three different orientations. The problem of finding a maximum convex partial transversal Q of $\mathcal{R}$ is NP-hard.*

First, note that we can choose the three orientations without loss of generality: any (non-degenerate) set of three orientations can be mapped to any other set using an affine transformation, which preserves convexity of transversals. We choose the three orientations in our construction to be vertical (|), the slope of 1 (⁄) and the slope of -1 (⧹).

Given an instance of Max-2-SAT we construct a set of segments $\mathcal{R}$ and then we prove that from a maximum convex partial transversal Q of $\mathcal{R}$ one can deduce the maximum number of clauses that can be made true in the instance.

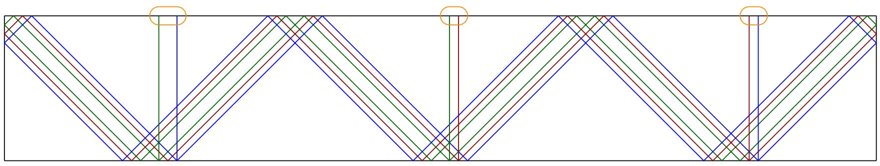
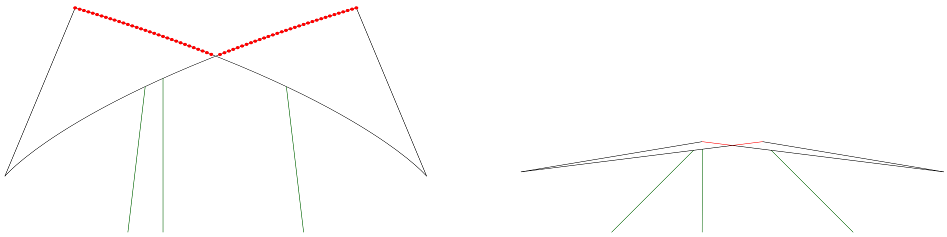


Figure 6 Overview of our construction. Each of the colored segment chains represents a variable. At each point where a chain bounces on the banana there is a fruit fly gadget. At each area marked orange there is a clause gadget. Each chain is only pictured once, but in actuality each chain is copied $m + 1$ times and placed at distance ϵ of each other. The distance between the different variables is exaggerated for clarity.



(a) Unswatted fly **(b)** Partially swatted fly

Figure 7 Sketch of a fly gadget. Endpoints of chain segments are divided over two implicit parabolic arcs together with some extra regions. To maximize our transversal, one of the two implicit arcs must be picked. This choice propagates through the rest of the construction. In our actual construction, the fly appears completely *swatted*: the aspect ratio of the fly approaches the local curvature of the banana, making it almost completely flat. The outer chain segments are then at an angle of 90°.

4.1 Overview of the construction

Our constructed set $\mathcal{R}$ consists of several different substructures. The construction is built inside a long and thin rectangle, referred to as the *crate*. The crate is not explicitly part of $\mathcal{R}$. Inside the crate, for each variable, there are several sets of segments that form chains. These chains alternate $\diagup$ and $\diagdown$ segments reflecting on the boundary of the crate. The idea is that an optimal solution must always place a point at (or close to) one of the endpoints of these segments, and furthermore, that two adjacent segments cannot both have their point at the reflection point. For each clause, there are vertical $|$ segments to transfer the state of a variable to the opposite side of the crate. Figure 6 shows this idea. However, the segments do not extend all the way to the boundary of the crate; instead they end on the boundary of a slightly smaller convex shape inside the crate, which we call the *banana*. By having all of the endpoints on the banana, the maximum partial transversal will be strictly convex. Aside from the chains associated with variables, $\mathcal{R}$ also contains segments that form gadgets to ensure that the variable chains have a consistent state, and gadgets to represent the clauses of our Max-2-SAT instance. Due to their winged shape, we refer to these gadgets by the name *fruit flies*. The idea is that an optimal solution must use one of two sequences of small points above the wings of the flies, and depending on this choice, can use only the endpoints of segments ending in one of the wings of the fly. See Figure 7 for a sketch of a fruit fly.

Our construction makes it so that we can always find a transversal that includes all of the chains, the maximum number of segments on the gadgets, and half of the $|$ segments.

For each clause of our MAX-2-SAT instance that can be satisfied, we can also include one of the remaining | segments. For the full construction and proof of correctness, see the full version of this paper [9].

Implications. Our construction strengthens the proof in [13] by showing that using only 3 orientations, the problem is already NP-hard. The machinery appears to be powerful: with a slight adaptation, we can also show that the problem is NP-hard for axis-aligned rectangles.

▶ **Theorem 15.** *Let $\mathcal{R}$ be a set of (potentially intersecting) axis-aligned rectangles. The problem of finding a maximum convex partial transversal Q of $\mathcal{R}$ is NP-hard.*

Proof. We build exactly the same construction, but afterwards we replace every vertical segment by a 45° rotated square and all other segments by arbitrarily thin rectangles. The points on the banana's boundary are opposite corners of the square, and the body of the square lies in the interior of the banana so placing points there is not helpful. ◀

References

1. E. M. Arkin, A. Banik, P. Carmi, G. Citovsky, M. J. Katz, J. S. B. Mitchell, and M. Simakov. Conflict-free covering. In *Proc. 27th Canadian Conference on Computational Geometry (CCCG)*, pages 17–23, 2015.
2. E. M. Arkin, C. Dieckmann, C. Knauer, J. S. B. Mitchell, V. Polishchuk, L. Schlipf, and S. Yang. Convex transversals. *Computational Geometry*, 47(2, Part B):224 – 239, 2014.
3. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 3rd edition, 2008.
4. D. Eppstein, M. Overmars, G. Rote, and G. Woeginger. Finding minimum area k-gons. *Discrete & Computational Geometry*, 7(1):45–58, 1992.
5. G. Even, Z. Lotker, D. Ron, and S. Smorodinsky. Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM Journal on Computing*, 33(1):94–136, 2003.
6. M. T. Goodrich and J. S. Snoeyink. Stabbing parallel segments with a convex polygon. *Computer Vision, Graphics, and Image Processing*, 49(2):152–170, 1990.
7. S. Har-Peled and S. Smorodinsky. Conflict-free coloring of points and simple regions in the plane. *Discrete & Computational Geometry*, 34(1):47–70, 2005.
8. M. J. Katz, N. Lev-Tov, and G. Morgenstern. Conflict-free coloring of points on a line with respect to a set of intervals. *Computational Geometry*, 45(9):508–514, 2012.
9. V. Keikha, M. van de Kerkhof, M. van Kreveld, I. Kostitsyna, M. Löffler, F. Staals, J. Urhausen, J. L. Vermeulen, and L. Wiratma. Convex partial transversals of planar regions. *arXiv preprint arXiv:1809.10078*, 2018.
10. J. S. B. Mitchell, G. Rote, G. Sundaram, and G. Woeginger. Counting convex polygons in planar point sets. *Information Processing Letters*, 56(1):45–49, 1995.
11. M. H. Overmars and E. Welzl. New methods for computing visibility graphs. In *Proc. 4th Annual Symposium on Computational Geometry (SCG)*, pages 164–171, 1988.
12. G. Rote, G. Woeginger, B. Zhu, and Z. Wang. Counting k-subsets and convex k-gons in the plane. *Information Processing Letters*, 38(3):149–151, 1991.
13. L. Schlipf. Notes on convex transversals. *arXiv preprint arXiv:1211.5107*, 2012.
14. M. Tompa. An optimal solution to a wire-routing problem. *Journal of Computer and System Sciences*, 23(2):127 – 150, 1981.