Hardness and Approximation of Minimum Convex Partition

Nicolas Grelier ✉
Department of Computer Science, ETH Zürich, Switzerland

Abstract

We consider the Minimum Convex Partition problem: Given a set P of n points in the plane, draw a plane graph G on P, with positive minimum degree, such that G partitions the convex hull of P into a minimum number of convex faces. We show that Minimum Convex Partition is NP-hard, and we give several approximation algorithms, from an $\mathcal{O}(\log OPT)$-approximation running in $\mathcal{O}(n^8)$-time, where OPT denotes the minimum number of convex faces needed, to an $\mathcal{O}(\sqrt{n}\log n)$-approximation algorithm running in $\mathcal{O}(n^2)$-time. We say that a point set is k-directed if the (straight) lines containing at least three points have up to k directions. We present an $\mathcal{O}(k)$-approximation algorithm running in $n^{\mathcal{O}(k)}$-time. Those hardness and approximation results also holds for the Minimum Convex Tiling problem, defined similarly but allowing the use of Steiner points. The approximation results are obtained by relating the problem to the Covering Points with Non-Crossing Segments problem. We show that this problem is NP-hard, and present an FPT algorithm. This allows us to obtain a constant-approximation FPT algorithm for the Minimum Convex Partition Problem where the parameter is the number of faces.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases degenerate point sets, point cover, non-crossing segments, approximation algorithm, complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.45

Related Version *Full Version*: https://arxiv.org/abs/1911.07697

Funding Research supported by the Swiss National Science Foundation within the collaborative DACH project *Arrangements and Drawings* as SNSF Project 200021E-171681.

Acknowledgements The author thanks Michael Hoffmann for his helpful advice.

1 Introduction

The CG Challenge 2020 organised by Demaine, Fekete, Keldenich, Krupke and Mitchell [5], was about solving instances of *Minimum Convex Partition* (MCP).

▶ **Definition 1** (Demaine et al. [5]: Minimum Convex Partition problem)**.** Given a set P of n points in the plane. The objective is to compute a plane graph with vertex set P (with each point in P having positive degree) that partitions the convex hull of P into the smallest possible number of convex faces. Note that collinear points are allowed on face boundaries, so all internal angles of a face are at most π.

As explained by Bose et al., this problem has applications in routing [3]. They show that a routing algorithm named *Random-Compass* that works for triangulations can be extended to convex partitions. Having a convex partition with few faces reduces the amount of data to store. From now on, we denote by P a set of n points in the plane.

In this paper, we present several approximation algorithms for MCP. We obtain those approximation algorithms by relating the MCP problem to the *Covering Points with Non-Crossing Segments* (CPNCS) problem. First, we define what *non-crossing segments* are.

38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 45; pp. 45:1–45:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Definition 2** (Non-Crossing Segments)**.** We call a part of a (straight) line bounded by two points a *segment*. The two points are referred to as *endpoints* of the segment. Note that we do not force the endpoints to be distinct, therefore we consider a point p as being a segment. The endpoint of p is p itself. Two segments are *non-crossing* if the intersection of their relative interior is empty.

▶ **Definition 3** (Covering Points with Non-Crossing Segments)**.** Given a set P of n points, find a minimum number of non-crossing segments whose endpoints are in P such that each point of P is contained in at least one segment.

The condition that the endpoints of the segments must be in P has no effect on the number of segments required. We add it as it simplifies some arguments. Note that CPNCS is not a so-called *set cover problem* nor an *exact cover problem*. We believe that CPNCS is interesting in itself. Even though it is a very natural problem, to the best of our knowledge it had not been introduced before.

1.1 NP-hardness results

Fevens, Meijer and Rappaport first considered the MCP problem in 2001 [7], and its complexity was explicitly asked about by Knauer and Spillner in 2006 [12]. It has remained open since then [2, 5]. We show in Section 5 that MCP is NP-hard. To do this, we use the decision version of the problem, as stated below:

▶ **Definition 4** (MCP - decision version)**.** Given a set P of points in the plane and a natural number k, is it possible to find at most k closed convex polygons whose vertices are points of P, with the following properties: a) The union of the polygons is the convex hull of P, b) the interiors of the polygons are pairwise disjoint, and c) no polygon contains a point of P in its interior.

We also show NP-hardness of a similar problem, which we call *Minimum Convex Tiling* problem (MCT). The problem is exactly as in Definition 4, but the constraint about the vertices of the polygons is removed (i.e. they need not be points of P). This can make a difference as shown in Figure 1. Equivalently, the MCT problem corresponds to the MCP problem when Steiner points are allowed. A *Steiner point* is a point that does not belong to the point set given as input, and which can be used as a vertex of some polygons. The MCT problem has been studied in 2012 by Dumitrescu, Har-Peled and Tóth, who asked about the complexity of the problem [6]. We answer their question, and our proofs are very similar for MCP and MCT. We also show in the full version of the paper [10] that CPNCS is NP-hard, even for some constrained point sets.

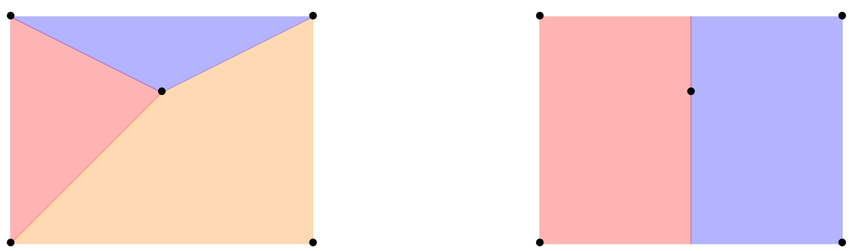

Figure 1 A minimum partition with three convex polygons and a tiling with two.

1.2 Approximation algorithms

For the related problem *Minimum Convex Partition of Polygons with Holes*, Bandyapadhyay, Bhowmick and Varadarajan showed the existence of a $(1+\varepsilon)$-approximation algorithm running in time $n^{\mathcal{O}((\log n/\varepsilon)^4)}$ [1]. Although they only consider holes with non empty interior, one can observe that their proof extends to the case of point holes. This is an even more general setting than MCP for point sets, so their algorithm also applies in our setting. This implies that MCP is not APX-hard unless $NP \subseteq DTIME(2^{\text{polylog n}})$.

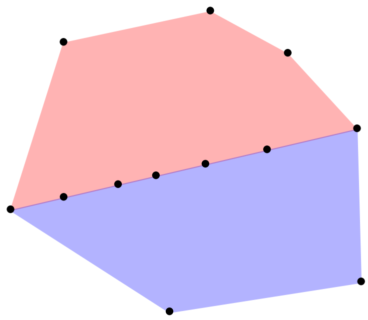

Figure 2 The number of inner points can be arbitrarily much larger than the number of convex faces required.

Under the assumption that no three points are collinear, Knauer and Spillner have shown the existence of a $\frac{30}{11}$-approximation algorithm for MCP in 2006 [12]. As a lower bound on the number of convex faces for one particular point set, they rely on the observation that each inner point has degree at least 3. The *inner points* of P are the points not on the boundary of the convex hull. This gives a lower bound on the number of edges, and therefore on the number of faces, by Euler's formula. Note that the restriction that no three points are on a line is necessary, as shown in Figure 2. There are only two faces in a minimum convex partition of this point set, and all the inner points have degree 2.

Additionally, Knauer and Spillner showed how to adapt any constructive upper bound on the number of faces into an approximation algorithm. More explicitly, they showed that if one can compute in polynomial time a convex partition with at most λn convex faces, then there exists a 2λ-approximation algorithm running in polynomial time. The best result to date is a proof by Sakai and Urrutia that one can partition a point set in quadratic time using at most $\frac{4}{3}n$ convex faces (the result was presented at the 7th JCCGG in 2009, the paper appeared on arXiv in 2019) [19]. Although they do not mention it, combining this result with the one by Knauer and Spillner gives a quadratic time $\frac{8}{3}$-approximation algorithm.

Concerning previous upper bounds, Neumann-Lara, Rivero-Campo and Urrutia first showed in 2004 how to construct in quadratic time a partition of any point set with at most $\frac{10}{7}n$ convex faces [17]. In 2006, Knauer and Spillner improved this to $\frac{15}{11}n$ convex faces [12]. As said above, the best known upper bound is $\frac{4}{3}n$, as proven by Sakai and Urrutia in 2009.

Relatedly for lower bounds, García-Lopez and Nicolás have given in 2013 a construction of point sets for which any convex partition has at least $\frac{35}{32}n - \frac{3}{2}$ faces [8].

All these results concerning upper bounds hold for all point sets, even where many points are on a line. Indeed, slightly shifting the points so that no three points are on a line can only increase the number of convex faces needed. So an upper bound for point sets where no three points are on a line also holds for all point sets. However, as mentioned above, the lower bound used by Knauer and Spillner does not extend to our setting, where we consider all point sets. They say that a constant-approximation algorithm would be desirable for unrestricted point sets, but so far not even an $\mathcal{O}(n^{1-\varepsilon})$-approximation is known. For the MCT problem, Dumitrescu, Har-Peled and Tóth showed the existence of a 3-approximation algorithm for point sets with no three collinear points [6]. They also ask whether a constant-approximation algorithm exists when this constraint is removed. However, so far no $\mathcal{O}(n^{1-\varepsilon})$-approximation algorithm is known. In Section 3, we prove the following:

► **Theorem 5.** *There exists $\mathcal{O}(\log OPT)$-approximation algorithms for MCP, MCT and CPNCS running in $\mathcal{O}(n^8)$-time.*

Allowing several points to be on a line does not simply create tedious technicalities to deal with. The crux of the matter is to find, for a fixed point set, an exploitable lower bound on the number of faces in a minimum convex partition. When no three points are on a line, the number of inner points in P gives a linear lower bound on the number of faces in a convex partition [12], and in a convex tiling [6]. In this paper, we consider point sets with no restriction. We introduce the CPNCS problem as it pinpoints where the difficulty of finding a constant-approximation algorithm for MCP is and makes the problem easier to study. We show in Section 2 the following theorem, which is used to prove Theorem 5:

► **Theorem 6.** *Let P be a set of n points with at least one inner point, and let $\lambda \geq 1$ be a real number. Let f_m denote the minimum number of faces in a convex partition of P. Let s_m denote the minimum number of non-crossing segments in a covering of the inner points of P, denoted by P_i.*

1. *It holds that $\frac{s_m}{6} \leq f_m \leq 8s_m$.*
2. *Given a covering of P_i with $s \leq \lambda s_m$ non-crossing segments, it is possible to compute in $\mathcal{O}(n^2)$-time a convex partition of P with at most $24\lambda f_m$ convex faces.*
3. *Given a convex partition of P with $f \leq \lambda f_m$ convex faces, it is possible to compute in $\mathcal{O}(n)$-time a covering of P_i with at most $44\lambda s_m$ non-crossing segments.*

The theorem also holds when considering convex tilings instead of convex partitions.

The idea behind the similarity of MCP, MCT and CPNCS is that they are all about maximizing the number of vertices of degree 2 with incident edges being aligned in a plane straight-line drawing of a graph on a point set. We show in the full version of the paper [10] that MCP and CPNCS are however not equivalent, meaning that one cannot use an optimal solution for one to deduce an optimal solution for the other.

1.3 Exact algorithms, FPT algorithms

Under the assumptions that the points lie on the boundaries of a fixed number h of nested convex hulls, and that no three points lie on a line, Fevens, Meijer and Rappaport gave an algorithm for solving MCP in time $\mathcal{O}(n^{3h+3})$ [7]. Observe that this is not an FPT algorithm. Some integer linear programming formulations of the problem have been recently introduced [2, 20, 4].

A first FPT algorithm with respect to the number k of inner points was introduced by Grantson and Levcopoulos, with running time $\mathcal{O}(2^{16k}k^{6k-5}n)$ [9]. The idea of the algorithm is to enumerate all plane graphs on the inner points, and then for each to them to guess how to connect the inner points to points on the boundary of the convex hull. Another FPT algorithm with respect to the number of inner points was later found by Spillner, with running time $\mathcal{O}(2^k k^4 n^3 + n \log n)$ [21].

We show in Section 4 the existence of an FPT algorithm that checks whether there is a solution for CPNCS with at most k non-crossing segments, running in time $\mathcal{O}(2^{k^2}k^{7k} + n^4 \log n)$. By Theorem 6, this gives us a constant-approximation FPT algorithm for MCP and MCT, where the parameter is the number of convex faces needed. Under the assumption that no three points are on a line, the number of faces in a minimum convex partition or in a minimum convex tiling is the same as the number of inner points, up to a constant multiplicative factor [12, 6]. However, without this assumption the number of inner points can be arbitrarily much larger than the minimum number of convex faces, as shown in Figure 2.

2 The relation between MCP, MCT and CPNCS

Throughout this section, we denote by P a point set in the plane. We denote by P_i the set of inner points of P. Let p be in P. If P and $P \setminus \{p\}$ do not have the same convex hull, we say that p is an *extreme point*. We denote by $P'_i \subseteq P_i$ the extreme points in P_i, where P_i denotes the inner points in P. Note that a point might lie on the boundary of the convex hull of a point set without being an extreme point. We say that P is *special* if $|P'_i| \leq 2$. Recall that for a given covering of a point set Q with non-crossing segments, we always assume that the endpoints of the segments are in Q.

▶ **Lemma 7.** *Let P be a set of n points that is not special. Given a covering K of P_i with s non-crossing segments, one can compute in $\mathcal{O}(n^2)$-time a convex partition Σ of P with at most $4s + 2|P'_i|$ faces. Moreover every segment in K is the union of some edges in Σ.*

Due to space constraints, we postpone the proof of Lemma 7 to the full version of the paper [10]. The idea of the proof is to compute a constrained triangulation with respect to the segments of the covering. This gives us a convex partition of the inner points, and it remains to connect the points in P'_i to points on the boundary of the convex hull.

▶ **Lemma 8.** *Let P be a set of n points. Given a convex tiling Σ of P with f faces, one can compute in $\mathcal{O}(n)$-time a covering K of P_i with at most $6f - 2|P'_i|$ non-crossing segments. Moreover every segment in K is the union of some edges in Σ.*

Proof. The proof is illustrated in Figure 3. Let us denote by $G_0 = (V_0, E_0)$ the plane graph corresponding to the convex tiling, where a point in V_0 is extreme or has degree at least 3. Observe that some points in V_0 might not be in P. Also, the relative interior of an edge in E_0 might overlap with points in P. We assume that G_0 is given with a doubly connected edge list (DCEL) structure. If there is an edge between two points on the boundary of the convex hull of V_0, but not consecutive, we remove this edge. Note that this decreases the number of faces by 1, and does not break the convexity property. We denote by m the number of such edges that we have removed. We also remove from P all points contained in the relative interior of an edge between two points on the boundary of the convex hull. We denote by P''_i the extreme points in P_i that we have not removed. As an edge contains at most two points in P'_i, we have $|P''_i| \geq |P'_i| - 2m$. Using the DCEL structure, this can be done in $\mathcal{O}(n)$-time. We have obtained a new graph $G = (V, E)$, and there are $f - m$ convex faces in G. We denote by Q the set of inner points that are of degree at least 3 in G. We set $k := |Q|$. Now observe that for each point p in P''_i, there exists at least one edge e in E with one endpoint in Q, one endpoint on the boundary of the convex hull, such that e overlaps with a point in P''_i. This is because if we consider p and the two lines going through p and one of the two consecutive vertices in P''_i (the one before p and the one after p when going around P''_i in clockwise order), they define a wedge that one edge must intersect because of convexity. The point in P''_i can be an endpoint of e or in its relative interior. If for a point $p \in P''_i$ there are several edges that satisfy the conditions, we choose one

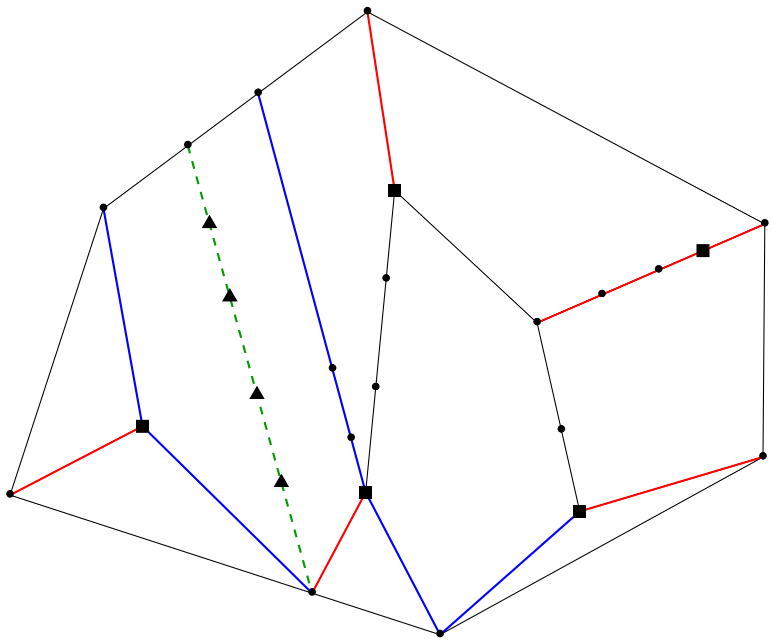

Figure 3 Illustration of Lemma 8. The green dashed edge and the triangle points are removed at the beginning for the analysis, and added back at the end. The extreme points in P_i'' are represented as square points. The edges in E' are in red. The other edges from P_i'' to the boundary of the convex hull are in blue.

arbitrarily. We denote these edges by E'. An edge in E' overlaps with exactly one point in P_i'', thus $|E'| = |P_i''|$. We denote by E_b the edges not in E' that have a point on the boundary of the convex hull and the other in Q, and we denote $|E_b|$ by m'. The vertices on the boundary of the convex hull are adjacent to two other vertices on the boundary of the convex hull. Moreover, those vertices are incident to $|P_i''| + m'$ additional edges. We have $2|E| = \sum_{v \in V} \deg(v) \geq 3k + 2(n-k) + |P_i''| + m' = k + 2n + |P_i''| + m'$. By Euler's formula, we have $f - m = |E| - n + 1 \geq \frac{k+|P_i''|+m'}{2} + 1$.

Now, the solution consists of the union of all edges in E incident to two points in Q, with the m edges in E_0 that we have removed, and with the $|P_i''| + m'$ edges in $E' \cup E_b$. We may need those edges as they might overlap with points in P_i. Note that there are at most $3k$ edges in E incident to two points in Q as G is plane. Moreover, all points in P_i are indeed covered by the edges in our solution. Thus, we obtain a covering of P_i with s segments, where $s \leq 3k + m + m' + |P_i''| \leq 3(2(f-m) - |P_i''| - m') + m + m' + |P_i''| \leq 6f - 5m - 2|P_i''| \leq 6f - 5m - 2(|P_i'| - 2m) \leq 6f - 2|P_i'|$. ◀

It is now possible to combine Lemmas 7 and 8 to prove Theorem 6. The proof can be found in the full version of the paper [10].

3 Approximation algorithms for CPNCS

We present several approximation algorithms for CPNCS. Let us first consider the ones whose approximation ratio is not output-dependent. The best algorithms in terms of approximation ratio are constant-approximation algorithms. The fastest algorithms take quadratic time. Therefore by 2. of Theorem 6, all the algorithms we present for CPNCS can be used to obtain approximation algorithms for MCP and MCT with the same order of approximation ratio, and the same order of running time. We have also one algorithm for CPNCS which realises an $\mathcal{O}(\log OPT)$-approximation in time $\mathcal{O}(n^8)$, where OPT denotes the minimum number of segments needed. Using 1. and 2. of Theorem 6, we also derive from it the $\mathcal{O}(\log OPT)$-approximation algorithm for MCP and MCT running in time $\mathcal{O}(n^8)$, where now OPT denotes the minimum number of faces needed in a convex partition, or in a convex tiling, respectively. This is how we prove Theorem 5. We first mention an easy approximation algorithm running relatively fast, at the cost of a high approximation ratio. The proof can be found in the full version of the paper [10]. The idea is to use the greedy algorithm to solve *Covering Points with lines* on P (a set cover problem), and then to split the lines into non-crossing segments.

▶ **Theorem 9.** *There exists an $\sqrt{n}\log(n)$-approximation algorithm for CPNCS running in $\mathcal{O}(n^2)$-time.*

Mitchell presented in a technical report some approximation algorithms for the problem of covering a point set with a minimum number of pairwise-disjoint triangles [16]. In his problem, the triangles of the covering must be subtriangles of some triangles given as input, for otherwise the problem would be trivial. He makes the assumption that no three points are on a line. We adapt his algorithms to our setting of CPNCS for point sets with no constraint. Let P be a set of n points. By doing a rotation if necessary, we can assume that no two points in P have the same x-coordinate. We say that a trapezoid is *constrained* if 1) it has two disjoint vertical sides, each lying on a line that contains a point in P, and 2) the two remaining sides are lying on lines that contain each at least two points in P. Note that there are $\mathcal{O}(n^6)$ constrained trapezoids.

We also allow for some degeneracies. Let us consider a triangle with vertices a, b and c, not all three on a line. If a is in P, the segment with endpoints b, c is vertical and lies on a line that contains a point in P, and the segments with endpoints a, b and a, c respectively are contained in some lines ℓ and ℓ' such that ℓ and ℓ' contains at least two points in P, then we say that the triangle is a constrained trapezoid. If a constrained trapezoid is split into two halves by a vertical line ℓ going through its interior, with ℓ containing a point in P, we obtain two constrained trapezoids. Likewise, if a segment s is in a constrained trapezoid τ, such that s lies on a line that contains at least two points in P, s intersects the interior of τ, and the endpoints of s are contained in the vertical sides of τ, then s splits τ into two constrained trapezoids.

For a set of points P where no two points have the same x-coordinate, we define the *enclosing trapezoid* as follows. Let ℓ_1 be the vertical line that contains the leftmost point in P, and let ℓ_2 be the vertical line that contains the rightmost point in P. Let L be the set of all lines containing at least two points in P. Observe that no line in L is vertical. We denote by a the highest intersection point between ℓ_1 and a line in L. We denote by b the lowest point intersection point between ℓ_1 and a line in L. Similarly, we denote by c and d, respectively, the highest intersection point, respectively the lowest intersection point, between ℓ_2 and a line in L. We denote by ℓ_3 the line containing a and c, and by ℓ_4 the line containing b and d. The *enclosing trapezoid* of P is the constrained trapezoid of $P \cup \{a, b, c, d\}$ defined by ℓ_1, ℓ_2, ℓ_3 and ℓ_4. It is denoted by $\mathcal{T}_P$.

We define the *strong guillotine property* in the special case of segments. We show that if there is a covering of P with s non-crossing segments, then there is a covering of S with $\mathcal{O}(s \log s)$ non-crossing segments having the strong guillotine property. We then present an algorithm that outputs an optimal solution among all the coverings with non-crossing segments having the strong guillotine property. Let S be a set of non-crossing segments covering P. We assume that the endpoints of the segments in S are in P. We say that S has the strong guillotine property with respect to a constrained trapezoid $\mathcal{T}$ that contains all segments in S if *a*) S contains at most one segment, or if *b*) there exists a partitioning line ℓ containing at least two points in P and at least one segment in S, such that for any segment $s \in S$, ℓ either contains s or does not intersect the relative interior of s, and ℓ splits $\mathcal{T}$ into two constrained trapezoids $\mathcal{T}_1$ and $\mathcal{T}_2$, such that the segments in $\mathcal{T}_1$, respectively $\mathcal{T}_2$, have the strong guillotine property with respect to $\mathcal{T}_1$, respectively $\mathcal{T}_2$, or if *c*) there exists a vertical line not intersecting with the relative interior of any segment in S, that splits $\mathcal{T}$ into two constrained trapezoids $\mathcal{T}_1$ and $\mathcal{T}_2$, such that the segments in $\mathcal{T}_1$, respectively $\mathcal{T}_2$, have the strong guillotine property with respect to $\mathcal{T}_1$, respectively $\mathcal{T}_2$. Observe that the line ℓ in case *b*) only intersects the vertical sides of $\mathcal{T}$, for otherwise ℓ would not split $\mathcal{T}$ into constrained trapezoids. We simply say that S has the strong guillotine property if it has the strong guillotine property with respect to the enclosing trapezoid $\mathcal{T}_P$.

▶ **Lemma 10.** *If there exists a covering of P with s non-crossing segments, then there exists a covering of P with $\mathcal{O}(s\log(s))$ non-crossing segments with the strong guillotine property.*

Proof. Recall that we assume that the endpoints of the segments are in P, by cropping them if need be. We can even crop some segments further such that they are pairwise-disjoint (it may be that now some segments are reduced to points). Consider the endpoints of the segments in that covering, that we denote by P'. We denote $|P'|$ by n', and we have $n' \leq 2s$. Note that no two points in P' have the same x-coordinate. We denote by X the set of x-coordinates of the points in P'. We now consider the segment tree based on X, as defined in [18]. The segment tree defines some canonical intervals. Each interval, whose endpoints are in X, is partitioned into $\mathcal{O}(\log s)$ canonical intervals. We partition each segment in the covering, such that the projection on the x-axis of each new segment is a canonical interval. Therefore we obtain a covering of P with $\mathcal{O}(s\log(s))$ non-crossing segments. We claim that this family of segments has the strong guillotine property. Let us denote by x_i, $1 \leq i \leq n'$ the elements in X, ordered by increasing value. We distinguish two cases. If there exists a segment σ whose projection on the x-axis is equal to the interval $[x_1, x_{n'}]$, then we recurse on the parts above and below σ which contain some segments. Observe that if $n' = 2$ we are done. If there is no such segment, then by definition of a segment tree, there is no segment in the covering whose relative interior intersects the vertical line ℓ with x-coordinate equal to $x_{\lfloor(1+n')/2\rfloor}$. Thus we can recurse on the left and right side of ℓ. ◀

▶ **Theorem 11.** *There exists an $\mathcal{O}(\log(OPT))$-approximation algorithm running in $\mathcal{O}(n^8)$-time for CPNCS.*

Proof. We explain how to recursively compute a minimum covering of P with non-crossing segments under the constraint that the solution has the strong guillotine property. The approximation ratio for the CPNCS problem when this additional constraint is removed follows from Lemma 10. If P is empty, we return no segment, which is a valid solution. If P can be covered with a single segment, we return that segment. This can be tested in $\mathcal{O}(n^2)$ time using duality. Now let us assume that not all points in P are on a line. We compute the enclosing trapezoid $\mathcal{T}_P$ of P. We consider the four vertices a, b, c, d of $\mathcal{T}_\mathcal{P}$. We start by adding the segment with endpoints a, c, and the segment with endpoints b, d. Now all the points to cover are within the enclosing trapezoid $\mathcal{T}_P$. We distinguish two cases, according to whether a segment with endpoints on the vertical sides of $\mathcal{T}_P$ is in a minimum covering with non-crossing segments having the strong guillotine property. If it is, we can add it to the solution and recurse on the two new constrained trapezoids. If no such segment is part of a minimum solution, then there exists a vertical line ℓ that splits a minimum solution into two parts, such that ℓ does not intersect the relative interior of any segment in that minimum solution. We can recurse on the $\mathcal{O}(n)$ choices of splitting vertically the constrained trapezoid into two constrained trapezoids. For each of the $\mathcal{O}(n^2)$ recursions, we compute the number of segments corresponding to that solution, and we output the solution corresponding to the one that minimises the number of segments.

To optimise we can do dynamic programming, and solve first the thinnest constrained trapezoids (in terms of width on the x-axis). There are $\mathcal{O}(n^6)$ constrained trapezoids, and we take quadratic time for each of them, so the total running time is $\mathcal{O}(n^8)$. ◀

We prove the following theorem in the full version of the paper [10].

▶ **Theorem 12.** *There exists an $\mathcal{O}(\log(n))$-approximation algorithm running in $\mathcal{O}(n^7)$-time for CPNCS.*

We say that a point set P is *k-directed* if there exists a set D of k directions, such that for any line ℓ that contains at least three points in P, the direction of ℓ is in D. For convenience, for any set of directions D and any segment s reduced to a point, we say that the direction of s is in D. We say that a set of segments S has the *autopartition property* if $|S| \leq 1$, or if there exists a line ℓ which contains at least one segment in S, and splits S into two sets that have the autopartition property. The relative interior of a segment in S is either contained in ℓ or does not intersect ℓ. Tóth has shown that any set of s' disjoint segments having up to k directions have an autopartition of size $\mathcal{O}(s'k)$ [22]. Using this result and similar techniques to the ones of Theorem 11, we show in the full version of the paper [10] the following:

▶ **Theorem 13.** *There exists an $\mathcal{O}(k)$-approximation algorithm for CPNCS in k-directed sets running in $n^{\mathcal{O}(k)}$. Furthermore, there exists a 4-approximation algorithm for CPNCS in 2-directed sets running in time $\mathcal{O}(n^5)$.*

4 Fixed-parameter algorithm for CPNCS

As mentioned in the introduction, there are known fixed-parameter algorithms for MCP, where the parameter is the number of inner points. We present here a fixed-parameter constant-approximation algorithm for MCP and MCT, where the parameter is the number of faces in a minimum convex partition or a minimum convex tiling, respectively. For point sets where no three points are on a line, the minimum number of convex faces is at least half the number of inner points [12], and the number of convex tiles is at list a sixth of the number of inner points [6]. However, as shown in Figure 2, when we allow for several points to be on a line, the number of inner points can be arbitrarily much larger than the number of convex faces in a minimum convex partition. If the number of inner points is significantly larger than the number of convex faces needed, our algorithm has a lower running time. We first show that CPNCS is in FPT.

▶ **Theorem 14.** *We can compute in time $\mathcal{O}(2^{k^2}k^{7k} + n^4 \log n)$ whether a point set P can be covered with at most k non-crossing segments, and to output such a covering if it exists.*

The proof uses a kernelisation technique presented by Langerman and Morin for *Covering Points with Lines* [13]. Assume there is a line ℓ that contains at least $k+1$ points in P. Then in any covering of P with at most k lines, ℓ must be in the covering. Otherwise, we would need at least $k+1$ lines to cover the points contained in ℓ. Now one can compute all of these lines that contain at least $k+1$ points, dismiss all of the covered points, until no line covers more than k of the remaining points. If there remains more than k^2 points, then there is no covering of the point set with at most k lines. Otherwise, one can compute every way of covering the $\mathcal{O}(k^2)$ remaining points, and check whether there is one that uses in total at most k lines. In our setting, we are looking for a covering with non-crossing segments, which makes it more difficult. Indeed, if a line ℓ contains at least $k+1$ points, we only know that ℓ must contain at least one segment of the covering. This means that we cannot simply dismiss the points covered by such a line. Also, we have to be careful about crossings. To prove Theorem 14, we need several lemmas. For a point set P, we say that a segment s is a *P-segment* if its endpoints are in P. Recall that we only consider coverings of a point set P with non-crossing P-segments.

▶ **Definition 15.** Let P be a point set, and let s and t be two crossing P-segments. We denote by p the intersection of s and t. We determine four points in P, that we call the *points enclosing* p. There are two points on $s \cap P$ and two points on $t \cap P$. The two points on $s \cap P$, denoted by u and v, are such that the segment with endpoints u and v, which we denote by uv, is the shortest P-segment contained in s whose relative interior contains p. Likewise, the two points u' and v' are such that $u'v'$ is the shortest P-segment contained in t whose relative interior contains p. The points u, v, u' and v' are the points enclosing p.

▶ **Lemma 16.** *Given a set P of n points, it is possible to compute in time $\mathcal{O}(n^4 \log n)$ the pairs of crossing P-segments, to find whether their intersection p is in P, and to store the points enclosing p. Additionally, we can also store for each P-segment how many points in P they contain, and the list of those points.*

The proof of Lemma 16 can be found in the full version of the paper [10].

▶ **Lemma 17.** *Given a set P of n points, and a natural number k, it is possible to find in time $\mathcal{O}(2^{k^2} + n^4 \log n)$ either a certificate that there is no covering of P with at most k non-crossing segments, or to output a family $\mathcal{F}$ of $\mathcal{O}(2^{k^2})$ sets S containing at most k non-crossing P-segments, with the following properties: For any fixed covering of P with at most k non-crossing P-segments, there exists a set S in $\mathcal{F}$ such that a) any segment $s \in S$ contains at least $k+1$ points in P, b) for each segment t of the covering, if $|P \cap t \cap s| \geq 2$ for some $s \in S$, then t is contained in s, and c) if a segment of the covering contains at least $k+1$ points in P, then it is contained in a segment in S.*

Let P be a point set and let k be a natural number. Observe that if a set S of segments satisfies property a), then in a covering with at most k segments of P, each segment s in S contains at least one segment t of the covering, such that $|P \cap t| \geq 2$. Indeed if there exists a segment $s \in S$ such that for any segment t in the covering, we have that $s \cap t$ contains at most one point in P, then at least $k+1$ segments are needed to cover the points in $P \cap s$. This implies that if S consists of m segments and satisfies properties a) and b), then there are at least m segments in the considered covering of P with non-crossing segments.

Proof of Lemma 17. We first do some preprocessing by using the algorithm of Lemma 16. This takes $\mathcal{O}(n^4 \log n)$ time. We create a list L of segments, which at the beginning is empty, and will contain the segments in S when we are done. For each line ℓ that contains at least $k+1$ points, we find the extremal points p and q of P contained in ℓ in time $\mathcal{O}(n)$. Then we add the line segment with endpoints p and q to L. Using the algorithm presented by Guibas et al. [11], we can compute all lines containing more than k points in time $\mathcal{O}(\frac{n^2}{k} \log(\frac{n}{k}))$. If there are more than k of such lines, we already know that there is no covering of P with at most k non-crossing segments of P. Indeed such a covering can only exist if there exists a covering of P with at most k lines. Let us now assume that there are at most k such lines. We add all corresponding segments to L in total time $\mathcal{O}(kn + \frac{n^2}{k}\log(\frac{n}{k}))$. Let us show that the segments in L satisfy properties a), b) and c), although they might still be crossing. First, property a) holds by definition. Moreover property b) holds for all coverings of P with at most k segments because a segment in L containing points p and q also contains all points on the line (p, q). Finally, property c) also holds trivially for all coverings of P with at most k segments.

We are now going to modify L and make copies of it while maintaining the fact that properties a), b) and c) hold. Our aim is that no two segments in L cross. Let us consider one segment s in L which is crossed by another segment s' in L. We denote by p the intersection of s and s'. We retrieve the points u and v such that uv is the shortest P-segment in s whose relative interior contains p. We do likewise with u' and v' in s'. Observe that not both uv and $u'v'$ can be in a covering of P with non-crossing segments. More generally, in a valid covering, at least one of uv and $u'v'$ is not contained in any segment of the covering. We create one copy of L, and recurse on two cases, one where we assume that uv is not contained in a segment of the covering, and one where we assume that $u'v'$ is not contained in a segment of the covering. Let us assume for now that uv is not contained in a segment of the covering. We keep s' in L, and s' might still be removed at a later step. We remove s from L. The segment s' splits s at p into two sides. Let us denote by x and y the endpoints of s, with u being closer to x than v is. If p is not in P, we consider the segments xu and vy. If p is in P, we consider the segments xp and py. Any of the two new segments that contains more than k points in P is added to L. Indeed property a) holds by definition. Moreover property b) holds because s was in L, and we are assuming that the segment uv is not contained in a segment of the covering. If a segment contains at most k points, we do not add it to L. We claim that property c) still holds. This is because if a point $q \in P$ which lies on a line that contains more than k points is not contained in some segment in L, that means that if a segment t contains q as well as at least k other points in P, then t also contains some segment which we are assuming not to be contained in the covering.

If we obtain more than k segments in L, we stop this branch of the recursion, as we already know that there is no valid covering of P with at most k segments, assuming that uv is not contained in a segment of the covering. We now iterate over all crossing segments in L. We obtain $O(k)$ segments in L, which are by construction non-crossing. As the depth of the recursion tree is in $\mathcal{O}(k^2)$, the number of leaves is in $\mathcal{O}(2^{k^2})$. We would like to say that each recursion implies the existence of one more segment in a covering with non-crossing segments, but this is a priori not the case. Therefore, if the number of lines containing more than k points is in $\Omega(k)$, we might have to do $\Omega(k^2)$ recursions. We can do the computation in total time $\mathcal{O}(2^{k^2} + kn + \frac{n^2}{k}\log(\frac{n}{k}))$, using the information we preprocessed. If we add to it the running time of the preprocessing, the total running time of the algorithm is in $\mathcal{O}(2^{k^2} + n^4 \log n)$. ◀

The proof of Theorem 14 appears in the full version of the paper [10]. The idea is to fix one valid covering K if it exists, and then to guess in time $\mathcal{O}(2^{k^2})$ the set $S \in \mathcal{F}$ of segments which corresponds to this covering K. Then we can argue by property c) that there are at most k^2 points in P not contained in some segments in S. It remains to guess what are the segments in K covering those points. If some of these segments split a segment in S, then we simply do as in the proof of Lemma 17 and update the set S of segments.

▶ **Theorem 18.** *It is possible to compute in time $\mathcal{O}(2^{36f^2} f^{42f+1} + n^4 \log n)$ a convex partition of a point set P with at most $24f$ convex faces, where f denotes the minimum number of convex faces required. The same holds when considering convex tilings.*

Proof. We first compute a minimum covering of the inner points in time $\mathcal{O}(2^{s^2}s^{7s+1}+n^4\log n)$ by applying the algorithm of Theorem 14 for $k=1,2,\ldots,s$, where s denotes the minimum number of segments required in a covering of the inner points. Then, by 2. of Theorem 6, we obtain in $\mathcal{O}(n^2)$-time a convex partition with at most $24f$ convex faces. The same holds with convex tilings for the same arguments. As by 1. of Theorem 6, we have $s \le 6f$, the total running time of the algorithm is as stated. ◀

We discuss in the full version of the paper [10] why the membership of CPNCS in FPT does not contradict the W[1]-hardness of Maximum Independent Set in Segment Intersection Graphs shown by Marx [15]. We also discuss why our techniques are not sufficient to obtain an exact FPT algorithm for MCP.

5 NP-hardness of MCP and MCT

Our proof of NP-hardness of MCP and MCT builds upon gadgets introduced by Lingas [14]. He used them to prove NP-hardness of several decision problems, including *Minimum Convex Partition for Polygons with Holes* and *Minimum Rectangular Partition for Rectilinear Polygons with Holes*. The entire proof appears in the full version of the paper [10]. The idea is to first mimic Lingas' proof. We show how we can embed the rectilinear polygon with holes into a grid $\mathit{\Lambda}$ of polynomial size. Then we add all edges of the grid outside of the polygon and inside of the holes to the drawing. This gives us a set $\mathit{\Phi}$ of unit length segments. We finally replace each unit segment by K collinear points, where K depends polynomially on the size of the input, and obtain a point set P. We show that the convex faces in a minimum convex partition of P have large area, and that if the interior of a convex face F in a convex partition of P intersects a segment in $\mathit{\Phi}$, then the area of F, denoted by $A(F)$, is not large enough. Therefore, the interior of a convex face F in a minimum convex partition of P does not intersect a segment in $\mathit{\Phi}$. From this we can conclude that convex partitions on P behave as if the segments of the polygon where there as constraints. Thus, the reduction works similarly as Lingas'. We present here our key lemma in the proof. It states that if the interior of a convex face F intersects a segment σ in $\mathit{\Phi}$, then F cannot have large area within two cells of $\mathit{\Lambda}$ on different sides of σ, where large area means larger than $1/K$.

▶ **Lemma 19.** *Let L and L' be two unit cells in $\mathit{\Lambda}$, and let F be a convex polygon whose interior does not contain any point in P. If $\mathrm{A}(F\cap L) > 1/K$, and the boundary of F crosses a segment of $\mathit{\Phi}$ between L and L', then $\mathrm{A}(F\cap L') < 1/K$.*

Proof. The proof is illustrated in Figure 4. By assumption, F intersects a line segment whose endpoints p and q are at distance $1/K$. Let us consider the two line segments s and s' of the boundary of S that intersect the line ℓ which contains p and q. Assume for contradiction that the lines containing respectively s and s' do not intersect, or intersect on the side of ℓ where L lies. This implies that $F\cap L$ is contained in a parallelogram that has area $1/K$, as illustrated in Figure 5. Indeed such a parallelogram has base $1/K$ and height 1, therefore $\mathrm{A}(F\cap L) \le 1/K$. This shows that the lines containing respectively s and s' intersect on the side of ℓ where L' lies. Using the same arguments as above, this implies $\mathrm{A}(F\cap L') < 1/K$. ◀

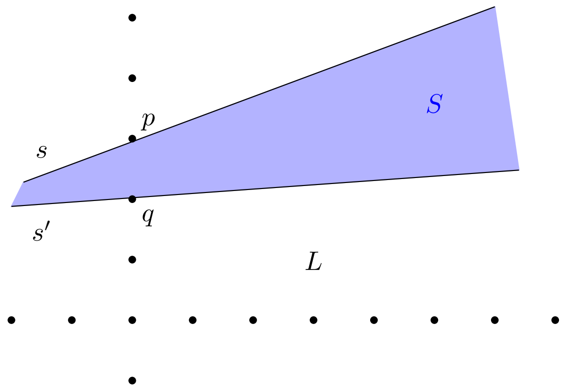

Figure 4 If $A(S \cap L) > 1/K$, the two lines containing s and s' intersect on the left side.

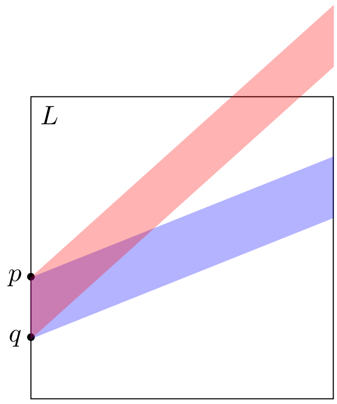

Figure 5 The area of the parallelograms is $1/K$.

6 Open problems

It would be interesting to have approximation algorithms for MCP, MCT and CPNCS with better ratio than $\mathcal{O}(\log OPT)$. As MCP is not APX-hard unless $NP \subseteq DTIME(2^{polylog\ n})$ [1], we expect that some improvement can be achieved.

A natural question is to ask whether MCP is FPT with respect to the number of faces in an optimal convex partition, as we have only shown a constant-approximation FPT algorithm. This question is open when having several points on a line is allowed, since otherwise the minimum number of convex faces is linear in the number of inner points.

We have shown that the decision versions of MCP and CPNCS are NP-complete, and that the one of MCT is NP-hard, but the question whether the decision version of MCT is in NP remains open. We also do not know the complexity of MCP and MCT when it is assumed that no three points are collinear.

References

1 Sayan Bandyapadhyay, Santanu Bhowmick, and Kasturi Varadarajan. Approximation schemes for partitioning: Convex decomposition and surface approximation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1457–1470. SIAM, 2014. doi:10.1137/1.9781611973730.96.

2 Allan S. Barboza, Cid C. de Souza, and Pedro J. de Rezende. Minimum convex partition of point sets. In *Proceedings of International Conference on Algorithms and Complexity*, pages 25–37. Springer, 2019. doi:/10.1007/978-3-030-17402-6_3.

3 Prosenjit Bose, Andrej Brodnik, Svante Carlsson, Erik D Demaine, Rudolf Fleischer, Alejandro López-Ortiz, Pat Morin, and J Ian Munro. Online routing in convex subdivisions. *International Journal of Computational Geometry & Applications*, 12(04):283–295, 2002. doi:10.1142/S021819590200089X.

4 Hadrien Cambazard and Nicolas Catusse. An integer programming formulation using convex polygons for the convex partition problem. In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.SoCG.2021.20.

5 Erik Demaine, Sándor Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. CG:SHOP 2020. https://cgshop.ibr.cs.tu-bs.de/competition/cg-shop-2020. Accessed: 12/02/2020.

6 Adrian Dumitrescu, Sariel Har-Peled, and Csaba D. Tóth. Minimum convex partitions and maximum empty polytopes. In *Proceedings of Scandinavian Workshop on Algorithm Theory*, pages 213–224. Springer, 2012. doi:10.1007/978-3-642-31155-0_19.

7 Thomas Fevens, Henk Meijer, and David Rappaport. Minimum convex partition of a constrained point set. *Discrete Applied Mathematics*, 109(1-2):95–107, 2001. doi:10.1016/S0166-218X(00)00237-7.

8 Jesús García-López and Carlos M. Nicolás. Planar point sets with large minimum convex decompositions. *Graphs and Combinatorics*, 29(5):1347–1353, 2013. doi:10.1007/s00373-012-1181-z.

9 Magdalene Grantson and Christos Levcopoulos. A fixed parameter algorithm for the minimum number convex partition problem. In *Japanese Conference on Discrete and Computational Geometry*, pages 83–94. Springer, 2004. doi:10.1007/11589440_9.

10 Nicolas Grelier. Hardness and approximation of minimum convex partition. *arXiv preprint*, 2019. arXiv:1911.07697.

11 Leonidas J. Guibas, Mark H. Overmars, and Jean-Marc Robert. The exact fitting problem in higher dimensions. *Computational geometry*, 6(4):215–230, 1996. doi:10.1016/0925-7721(95)00020-8.

12 Christian Knauer and Andreas Spillner. Approximation algorithms for the minimum convex partition problem. In *Proceedings of Scandinavian Workshop on Algorithm Theory*, pages 232–241. Springer, 2006. doi:10.1007/11785293_23.

13 Stefan Langerman and Pat Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005. doi:10.1007/s00454-004-1108-4.

14 Andrzej Lingas. The power of non-rectilinear holes. In *Proceedings of International Colloquium on Automata, Languages, and Programming*, pages 369–383. Springer, 1982. doi:10.1007/BFb0012784.

15 Dániel Marx. Parameterized complexity of independence and domination on geometric graphs. In *International Workshop on Parameterized and Exact Computation*, pages 154–165. Springer, 2006. doi:10.1007/11847250_14.

16 Joseph S. B. Mitchell. Approximation algorithms for geometric separation problems. *Technical report, Dept. of Applied Math. and Statistics, State U. of New York at Stony Brook*, 1993. Available at https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.7089&rep=rep1&type=pdf.

17 Víctor Neumann-Lara, Eduardo Rivera-Campo, and Jorge Urrutia. A note on convex decompositions of a set of points in the plane. *Graphs and Combinatorics*, 20(2):223–231, 2004. doi:10.1007/s00373-004-0555-2.

18 Franco P. Preparata and Michael I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985. doi:10.1007/978-1-4612-1098-6.

19 Toshinori Sakai and Jorge Urrutia. Convex decompositions of point sets in the plane. *arXiv preprint*, 2019. `arXiv:1909.06105`.

20 Allan Sapucaia, Pedro J. de Rezende, and Cid C. de Souza. Solving the minimum convex partition of point sets with integer programming. *Computational Geometry*, page 101794, 2021. `doi:10.1016/j.comgeo.2021.101794`.

21 Andreas Spillner. A fixed parameter algorithm for optimal convex partitions. *Journal of Discrete Algorithms*, 6(4):561–569, 2008. `doi:10.1016/j.jda.2008.07.002`.

22 Csaba D. Tóth. Binary space partitions for line segments with a limited number of directions. *SIAM Journal on Computing*, 32(2):307–325, 2003. `doi:10.1137/S0097539702403785`.