Packing Segments in a Simple Polygon is APX-hard

Heuna Kim [*]   Tillmann Miltzow [†]

Abstract For a given set of line segments and a polygon P in the plane, we want to find the maximum number of segments that can be disjointly embedded by translation into P. We show APX-hardness and discuss variations.

This problem can be considered in two respects : as a variant of the Kakeya problem and as a maximum-packing problem for line segments.

1 Introduction

The Kakeya Problem. The famous Kakeya problem asks for the region R in the plane with minimum-area such that a unit-length line segment can continuously rotate by π within R. One variant of the Kakeya problem relaxes the continuous rotation and tries to find a planar region R' with the minimum area such that translates of all the unit-length line segments in the plane can be placed in R'. The segments may intersect. This region R' is called a *minimum area translation cover*.

Pál [5, 4] solved these two problems, and many other interesting variations about the minimum-area translation cover have been studied (refer [3, 6] for surveys).

A Minimum-Container Problem and a 3-approximation Algorithm. Finding a minimum-area translation cover can be considered as a minimum-container problem if we want to *disjointly* embed line segments. The following question arises naturally in this context; given a set of line segments $\mathcal{S}$, what is the minimum-area convex body R such that translates of segments in $\mathcal{S}$ can be disjointly embedded in R?

We suspect this problem is computationally intractable, but not much is known about this problem except for a 3-approximation algorithm by Sang Won Bae (by private communication).

The 3-approximation algorithm is as follows. Using the algorithm by Ahn et al. [1], we compute the triangle T which is the minimum-area convex translation cover of the given set of line segments $\mathcal{S}$. Then, we construct a convex trapezoid Q as follows. First translate two copies T_1, T_2 of T so that one side of
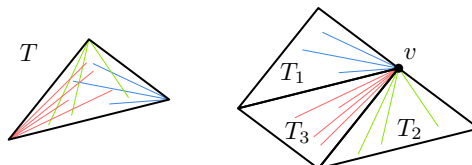

Figure 1: The minimum-area convex translation cover T and the trapezoid $Q = T_1 \cup T_2 \cup T_3$.

each copy is aligned on a line and T_1 and T_2 share one vertex v. We obtain the third copy T_3 by rotating T by π and translate it so that the three copies form the trapezoid $Q = T_1 \cup T_2 \cup T_3$, see Figure 1.

Then all segments in $\mathcal{S}$ can be disjointly embedded in Q; every segment s in $\mathcal{S}$ can be translated in a way that one of its endpoints lies on v and s still lies inside Q. Since the optimal area is at least the area of T, the obtained trapezoid gives a 3-approximation.

Problem Definition and Summary of Results. To solve a minimum-container problem it is natural to consider its dual, that is, a maximum-packing problem. We consider the maximum-packing problem in this abstract. We show hardness results for simple polygons and a simple approximation algorithm for convex polygons.

As in [2], we define MaxSegPack$d$ for a class $\mathcal{R}$ of regions in $\mathbb{R}^d$ as the following problem; given a collection of (open) segments and a region $R \in \mathcal{R}$, what is the maximum number of segments that can be disjointly embedded in R by translation?

This problem is known to be NP-hard when $\mathcal{R}$ is a convex 3-polytope of general regions in the plane [2]. We state the result for a convex 3-polytope.

Theorem 1 ([2]) MaxSegPack3 *for a convex 3-polytope is NP-hard.*

We state the main results as the following theorem.

Theorem 2 MaxSegPack2 *for a simple polygon and a set of unit segments $\mathcal{U}$ is strongly NP-complete. Also, approximating an optimal solution of MaxSegPack2 for a simple polygon and a set of unit segments with an approximation ratio $15/16 + \varepsilon$ is NP-hard for any $\varepsilon > 0$.*

We could also find a simple approximation algorithm.

---

[*]Institut für Informatik, Freie Universität Berlin, `heunak@mi.fu-berlin.de`. This research was supported by the Deutsche Forschungsgemeinschaft within the research training group Methods for Discrete Structures (GRK 1408).

[†]Institut für Informatik, Freie Universität Berlin, `t.m@fu-berlin.de`

Theorem 3 *There exists a k-approximation algorithm for* MaxSegPack2 *for a convex k-gon.*

By inspecting the proof from Theorem 1 in [2], we could easily conclude NP-hardness for high-dimensional cases.

We extend MaxSegPack$d$ to the following problem MaxPack(d_K, d_S) ; given a collection of (open) d_S-simplices and region R in d_K-space, what is the maximum number of simplices that can be disjointly embedded in R by translation?

Theorem 4 MaxPack(d_K, d_S) *for a convex d_K-polytope is NP-hard for all $d_K \geq 3, d_S \geq 1$.*

Remark. When a line segment s can be embedded in some region R, we say s *fits* in R. Also, if a set of line segments $\mathcal{S}$ can be disjointly embedded in R, we say $\mathcal{S}$ *can be packed* in R.

We regard two line segments of the same lengths and the same slopes as the same line segment since if two line segments have the same lengths and the same slopes we can overlap them completely by translation.

2 Proof of Theorem 2

We first show that MaxSegPack2 for a simple polygon P is in NP and then show that it is NP-hard. A natural candidate for a certificate of this problem is the set of the coordinates of the endpoints of the line segments. We can check whether the line segments are inside a given simple polygon P and whether they have no intersections by using linear inequalities.

We claim that those coordinates and the coefficients of linear inequalities can be described with polynomial precision. To this end, it is enough to show that the coordinates correspond to a feasible solution of conjunctions and disjunctions of a polynomial number of linear inequalities with coefficients of bounded precision.

To specify the linear inequalities, we first triangulate the given simple polygon arbitrarily. Three inequalities suffice to describe if each endpoint lies in one of the triangles. This gives us $6n$ inequalities, where n specifies the number of line segments we want to pack. A pair of line segments is crossing free if and only if at least one of them is completely to the left or completely to the right of the supporting line of the other. Since two linear inequalities suffice to describe if a line segment is to the left of another, this gives us $2\binom{n}{2}$ linear inequalities. Lastly, we need to specify two equalities per line segment to define the slope and the length of line segments (relative positions of two endpoints). In total, this gives us $6n + 2\binom{n}{2}$ inequalities and $2n$ equalities with coefficients of bounded precision. Hence, we can verify any certificate in a polynomial time.

Before describing the reduction from MAX-3-SAT, we state the following two lemmas for constructing gadgets. Lemma 5 will be used for the clause gadgets and Lemma 6 for the variable gadgets.
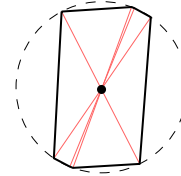

Figure 2: Four segments and a polygon such that exactly one of the segments fits but no two of them can be packed.

Lemma 5 *Let S be a set of unit-length line segments with distinct slopes. We construct a convex polygon $Q = Q(S)$ with the following properties:*
  1. *any segment $s \in S$ fits in Q;*
  2. *no two segments in S can be packed in Q; and*
  3. *no unit-length line segment $s \notin S$ fits in Q.*

Proof. Translate all the segments of S so that their midpoints lie at the origin. Now define $Q(S)$ as the convex hull of all those segments; see Figure 2 for an illustration.

The diameter of Q is 1 and the diameter is attained only for pairs of opposite extreme points of Q. Therefore, a unit-length line segment s fits in Q if and only if s can be translated in a way that its endpoints lie at opposite extreme points of Q. This implies the first and the third property.

Each segment s that fits in Q has a unique position in Q and this unique position always goes through the origin. Thus, no two segments of unit length can be packed in Q. This implies the second property.   □

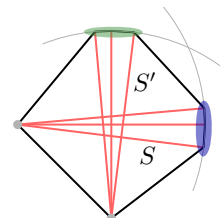

Figure 3: Sets S and S' and the convex polygon $R(S, S')$ constructed from them.

Lemma 6 *Let S be a set of unit length line segments such that the angle with the x-axis is within ± 0.1 radian, and let S' be a set of unit-length line segments such that the angle with the y-axis is within ± 0.1 radian.*

There exists a convex polygon $R = R(S, S')$ with the following properties:

1. segments in S can be packed in R;
2. the set S' can be packed in R;
3. no two segments $s \in S$ and $s' \in S'$ can be packed in R; and
4. no unit segment $s \notin S \cup S'$ fits into R.

Proof. Translate the left endpoint of every line segment $s \in S$ to the point $(-0.5, 0)$ and the bottom endpoint of every line segment $s' \in S'$ to the point $(0, -0.5)$. The convex hull of those segments define $R = R(S, S')$. See Figure 3.

The diameter of Q is 1 and the diameter is attained only for pairs of points (p, q) such that either 1) $p = (-0.5, 0)$ and q is one of right extreme points (marked blue in Figure 3). or 2) $p = (0, -0.5)$ and q is one of top extreme points (marked green in Figure 3). These are exactly the endpoints of segments in $S \cup S'$ after we moved the segments of S. By the same argument as in Lemma 5, any unit-length line segment s fits in R if and only if $s \in S \cup S'$. Each segment s that fits in R has a unique position $p(s)$ in R. Observe that $p(s_1)$ and $p(s_2)$ are disjoint if either $s_1, s_2 \in S$ or $s_1, s_2 \in S'$ and $p(s_1)$ and $p(s_2)$ intersect otherwise. Thus, any two segments s_1 and s_2 can be packed in R if and only if either $s_1, s_2 \in S$ or $s_1, s_2 \in S'$. Altogether these arguments imply the above four properties. $\square$

Given a 3-CNF formula ϕ with m clauses and n variables, we construct a simple polygon P and a set of $2m$ unit segments $\mathcal{U}$ that satisfy the following property; there exists an assignments that satisfies t clauses of ϕ if and only if $t + m$ elements of $\mathcal{U}$ can be disjointly embedded in P.

We begin by defining the line segments. Then we describe clause and variable polygons and finally we describe how to join everything to one big polygon.

For each clause C_i, $i = 1, \ldots, m$ of ϕ we construct two unit segments s_i and s'_i. The line segment s_i forms an angle $\alpha_i = \frac{i}{100m}$ with the x-axis and s'_i forms an angle $\alpha'_i = \frac{i}{100m}$ with the y-axis. [1] Note that all s_i's can be regarded as slight perturbations of a horizontal unit segment, and all s'_i as a slight perturbation of a vertical unit segment.

For each clause C_i we define the *clause polygon*

$$Q_i = Q(\{s_i, s'_i\})$$

according to Lemma 5.

For each variable x_j with $j = 1, \ldots, n$, we define

$$S_j = \{\, s_i \mid \text{the literal } x_j \text{ is contained in } C_i \,\} \text{ and}$$

$$S'_j = \{\, s'_i \mid \text{the literal } \overline{x_j} \text{ is contained in } C_i \,\}.$$

---

[1]To compute the endpoints of the segments we need sine and cosine operations, but it is not necessary since the construction does not depend on the exact values of the angles. We also could define the angles as rational values.
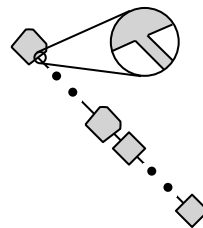

Figure 4: Joining polygons together without new segments fitting in.

For each variable x_j we define the *variable polygon*

$$R_j = R(S_j, S'_j)$$

according to Lemma 6. Note that each segment $s \in \mathcal{U}$ fits in at most four polygons: one clause polygon and at most three variable polygons.

The polygon P is defined by joining all the polygons $Q_1, \ldots, Q_m, R_1, \ldots, R_n$. In order to join these polygons, add a narrow diagonal tunnel from one polygon to the next; see Figure 4 for an illustration. Since every segment in $\mathcal{U}$ is either almost horizontal or vertical, none of them fits into the tunnel.

It is clear that this construction can be done within a polynomial time. For this polygon P and this set of line segments $\mathcal{U}$, we claim that there exists an assignment that satisfies t clauses of ϕ if and only if $t + m$ elements of $\mathcal{U}$ can be packed in P.

First suppose that we are given an assignment A that satisfies t clauses of ϕ. We will describe how to embed $t + m$ segments in the polygon P. There are some segments that fit in P not uniquely but in several possible variable polygons. In this case, we make an arbitrary choice. If x_j is true in A, place segments in S_j in the variable polygon R_j and if x_j is false in A, place segments in S'_j in R_j unless the segments are already placed in some other variable polygon. We also place all remaining segments into their corresponding clause polygon Q_i if possible.

If C_i is satisfied by A, both segments s_i and s'_i are placed in P for the following reason. Either s_i or s'_i is placed in R_j for some j since at least one variable x_j in C_i makes C_i satisfied. We placed the other to Q_i unless it is already contained in a different variable polygon.

Otherwise, only one of the segments s_i or s'_i fits in P, since neither s_i nor s'_i are contained in any variable polygon R_j and both segments cannot fit in Q_i. Since t clauses are satisfied, the first case happens t times and the second case appears $m - t$ times. Hence, $t + m$ segments can be packed into P.

For the other direction, suppose $t + m$ segments in $\mathcal{U}$ can be packed in P. We assume this packing is maximal. We define an assignment A by checking which segments are placed in R_j. If R_j contains a segment of S_j then we set x_j to true and otherwise

we set x_j to false. We can repeat the same argument in the other direction. For each clause C_i, if s_i and s_i' are both packed, then either s_i or s_i' is in some R_j, which implies that the clause C_i is satisfied by the variable x_j. Otherwise, one of s_i and s_i' is packed, but none of s_i and s_i' is placed in a variable polygon, and this implies that C_i cannot be satisfied by A. Then $2 \times n_S + n_N = t + m$ and $n_S + n_N = m$ where n_S is the number of satisfied clauses and n_N is the number of non-satisfied clauses. Then the number of satisfied clause in A is t. This shows the problem is NP-hard.

Finally we show that no approximation algorithm exists with an approximation ratio $15/16 + \varepsilon$ for any $\varepsilon > 0$. Suppose there exists an approximation algorithm for MaxSegPack2 for a simple polygon and a set of unit length segments with an approximation ratio $15/16 + \varepsilon/2$ for some $\varepsilon > 0$. By using the previous construction for any CNF formula ϕ of m clauses, we can find an assignment A that satisfies t clauses where $\frac{t+m}{2m} \geq 15/16 + \varepsilon/2$; that is, we have an approximation algorithm for MAX-3-SAT with an approximation ratio $t/m \geq 7/8 + \varepsilon$.

Since there is no approximation algorithm for MAX-3-SAT with the approximation ratio $7/8 + \varepsilon$ for any $\varepsilon > 0$ unless P=NP, there exists no approximation algorithm for MaxSegPack2 for a simple polygon and a set of unit segments with an approximation ratio $15/16 + \varepsilon/2$ for any $\varepsilon/2 > 0$ unless P=NP.

3   Approximation Algorithm for a Convex k-gon

The following algorithm gives a k-approximation for MaxSegPack2 for a convex polygon.

   Input: a set of line segments $\mathcal{S}$; convex k-gon P
   Output: $\mathcal{T} \subseteq \mathcal{S}$; a k-approximated solution
   for all $v \in$ vertices of P **do**
       $S_v := \{s \in \mathcal{S} : s$ can be placed on v inside $P\}$
   end for
   return the largest set S_v

Any segment $s \in \mathcal{S}$ that fits in P can be translated so that one of endpoints of v is on a vertex of P and v still lies in P. For each vertex v of P, all the elements S_v can be packed in P. Since

$$\bigcup_{p:\text{vertices of } P} S_v$$

is at least the optimal solution, the largest set S_v has the cardinality at least $1/k$ of the optimal solution.

4   Hardness for d-space

Theorem 1 in [2] states MaxSegPack3 for a convex 3-polytope is NP-hard; that is, MaxPack$(3, 1)$ is NP-hard. In the proof, all line segments were constructed in a way that they are uniquely embeddable in a convex 3-polytope for the reduction. We can prove
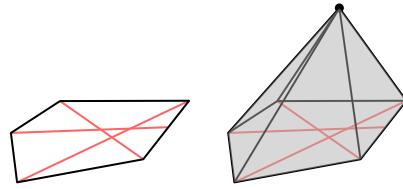

Figure 5: Visualization of constructing a pyramid, in dimension three.

that MaxPack(d_K, d_S) for a convex d_K-polytope is NP-hard inductively by reducing (1) an instance of MaxPack$(d_K, 1)$ to an instance of MaxPack$(d_K + 1, 1)$ and (2) an instance of MaxPack(d_K, d_S) to an instance MaxPack$(d_K + 1, d_S + 1)$.

Let $(K, \mathcal{S})$ be any instance of MaxPack$(d_K, 1)$ where K is a convex d_K-polytope and $\mathcal{S}$ a set of line segments that can be uniquely embedded in K. We construct K' by taking a pyramid whose base is K. Then K' is convex $(d_K + 1)$-polytope. Then, $(K', \mathcal{S})$ is an instance of MaxPack$(d_K + 1, 1)$ whose solution corresponds to a solution of $(K, \mathcal{S})$ for MaxPack$(d_K, 1)$, since all line segments $s \in \mathcal{S}$ can be embedded in K uniquely and s cannot be embedded in any smaller homothetic copies of K. This is the reduction for (1), and the reduction for (2) is quite similar; we replace $s \in \mathcal{S}$ by the convex hull s' of s and the apex of K'. Therefore, MaxPack(d_K, d_S) is NP-hard for all $d_K \geq 3, d_S \geq 1$.

References

[1] H.-K. Ahn, S. W. Bae, O. Cheong, J. Gudmundsson, T. Tokuyama, and A. Vigneron. A generalization of the convex Kakeya problem. In *LATIN 2012: Theoretical Informatics*, pages 1–12. Springer, 2012.

[2] M. G. Dobbins and H. Kim. Packing segments in a convex 3-polytope is np-hard. *30th European Workshop on Computational Geometry (EuroCG)*, 2014.

[3] I. Łaba. From harmonic analysis to arithmetic combinatorics. *Bulletin (New Series) of the American Mathematical Society*, 45(1):77–115, 2008.

[4] J. Pál. Ein Minimumproblem für Ovale. *Mathematische Annalen*, 83(3):311–319, 1921.

[5] J. Pál. *Ueber ein elementares Variationsproblem*, volume 3. AF Host, 1921.

[6] T. Tao. From rotating needles to stability of waves: Emerging connections between. *Notices of the AMS*, 48(3), 2001.