International Journal of Computational Geometry & Applications

CUTTING OUT POLYGONS WITH LINES AND RAYS *

OVIDIU DAESCU AND JUN LUO

Department of Computer Science,
University of Texas at Dallas
Richardson, TX 75080, USA
Email: {daescu,ljroger}@utdallas.edu

We present approximation algorithms for cutting out polygons with line cuts and ray cuts. Our results answer a number of open problems and are either the first solutions or significantly improve over previously known solutions. For the line cutting version, we prove a key property that leads to a simple, constant factor approximation algorithm. For the ray cutting version, we prove it is possible to compute in almost linear time a cutting sequence that is an $O(\log^2 n)$-factor approximation of an optimal cutting sequence. No algorithms were previously known for the ray cutting version.

Keywords: Polygon cutting; approximation algorithms; computational geometry.

1. Introduction

About two decades ago Overmars and Welzl [6] have first considered the problem of cutting out a polygon in the cheapest possible way. The problem falls in the general area of stock cutting, where a given shape needs to be cut out from a parent piece of material, and it is defined as follows: *Given a polygonal piece of material Q with a polygon P drawn on it, cut P out of Q by a sequence of "guillotine cuts" in the cheapest possible way.*

A *guillotine cut* is a line cut that does not cut through the interior of P and separates Q into a number of pieces, lying on both sides of the cut. A guillotine cut is an *edge cut* if it cuts along an edge of P. After a cut is made, Q is updated to that piece that still contains P. A *cutting sequence* is a sequence of cuts such that after the last cut in the sequence we have $P = Q$. The cost of a cut is the length of the intersection of the cut with Q and the goal is to find a cutting sequence that minimizes the total cost. See Figure 1 for an example.

From the definition of the problem it follows that P must be convex for a cutting sequence to exist. Overmars and Welzl [6] proved a number of properties for the case when both P and Q are convex polygons with n and m vertices, respectively, including: (i) There exists a finite optimal cutting sequence with at most $5n$ cuts, all touching P; (ii) There are cases in which there is no optimal cutting sequence with

*This research was partially supported by NSF grant CCF-0430366.

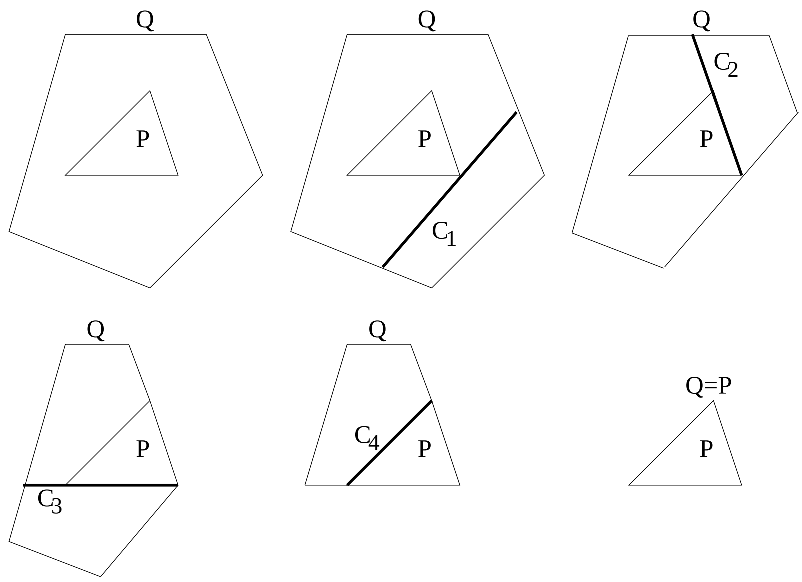

Fig. 1. A cutting sequence (bold lines) $\{C_1, C_2, C_3, C_4\}$ for cutting P out of Q.

all cuts along the edges of P and (iii) When only edge cuts are allowed, an optimal cutting sequence can be computed in $O(m + n^3)$ time by a dynamic programming algorithm. They further noted that when Q is not convex there are cases in which there is no optimal cutting sequence with all cuts touching P.

Overmars and Welzl [6] have also left a number of open problems including: (1) An algorithm for computing an optimal cutting sequence when Q is convex. Since the problem is not discrete, it might be possible that an optimal cutting sequence cannot be computed. (2) A proof of existence of a finite optimal cutting sequence when Q is not convex (regarding Q as an open object). (3) Algorithms to approximate an optimal cutting sequence. (4) The situation in which guillotine cuts are replaced by ray cuts, where a ray cut originates from infinity and ends at some point in Q. In this case, some non-convex polygons are *ray cuttable*. These polygons have the property that every edge of the polygon must be extendible to a ray. As observed in [2], this condition is more restrictive than *(weakly) external visibility*, which requires that every point on the boundary of P can "see" to infinity. The problem of computing or approximating optimal cutting sequences with rays has been left completely open.

About a decade later Bhadury and Chandrasekaran [1] have been able to answer the first open problem in a negative sense. Specifically, they showed that the problem has optimal solutions that lie in the algebraic extension of the field that the input data belongs to. Due to this algebraic nature of the problem, an approximation scheme is the best that one can achieve. They also provided an approximation scheme that, given an error range δ, is polynomial in δ and the encoding length of the input data in unary, and gives a cutting sequence $\mathcal{S}(\delta)$ with total cost at most δ more than that of an optimal cutting sequence $\mathcal{S}^*$.

For the fourth open problem, it has been proven in [2] that ray cuttability can be tested in linear time. However, the problem itself (computing or approximating an optimal sequence of ray cuts) has been again left open.

Very recently, Dumitrescu [3] has proved that there exists an $O(\log n)$-factor approximation algorithm which runs in $O(mn + n\log n)$ time for cutting out a convex polygon P with n vertices from a convex polygon Q with m vertices. Thus, he gives a first answer to the third open problem. He also raised the following two interesting questions: (1) If Q is the minimum axis-aligned rectangle enclosing P, is an optimal edge cutting sequence a constant factor approximation of an optimal cutting sequence? Answering this question in an affirmative sense would result in a constant factor approximation algorithm for cutting P out of Q. (2) Is it possible to extend the approximation results for guillotine cutting to ray cutting?

Some other related problems have been studied recently [2, 5, 7]. In [2], E. Demaine, M. Demaine and C. Kaplan present linear-time algorithms to recognize polygons cuttable by a (small) circular saw, and to recognize polygons cuttable by all (large) circular saws. In particular, for small saws, they prove that a polygon is cuttable by a circular saw precisely if it does not have two adjacent reflex vertices (vertices with interior angle larger than π; see Theorem 1 [2]) and give a linear time, 2.5-factor approximation algorithm. In [5], Jaromczyk and Kowaluk consider cutting polyhedral shapes with a hot wire cutter and give an $O(n^5)$ time algorithm that constructs a cutting path, if one exists. In [7], Pach and Tardos consider the problem of separating a large family of subsets from a family of pairwise disjoint compact convex sets drawn on a sheet of glass. They give a few results, including a proof that any family of n pairwise disjoint compact convex sets in the plane has at least $\Omega(n^{1/3})$ separable members. They also show that for any $\varepsilon > 0$, there exists a constant $c_\varepsilon > 0$ such that every ε-fat family of n pairwise disjoint compact convex sets in the plane has at least $c_\varepsilon n/\log n$ separable members.

Our results. In this paper we affirmatively answer the first open problem in [3], by proving there exists a sequence of edge cuts that is guaranteed to be a constant factor approximation of an optimal cutting sequence. We use this result to obtain a constant factor approximation algorithm for cutting P out of Q, when both P and Q are convex polygons, as follows. We first show how to cut out a triangle Q containing P of about the same size as P in $O((n+m)\log(n+m))$ time, with the cost of a cut sequence $\mathcal{S}$ only a constant factor more than the cost of an optimal cutting sequence $\mathcal{S}^*$. This part of the algorithm is based on simple observations and a mapping to a *(length,angle)* domain. Then, with an additional $O(n^3)$ time, we compute an optimal edge cutting sequence, resulting in an $O(n^3+(n+m)\log(n+m))$ time, constant factor approximation algorithm for cutting P out of Q. Alternatively, one can use a direct approach to obtain an $O(\log n)$-factor approximation algorithm that requires only $O((n+m)\log(n+m))$ time, an improvement over the solution in [3] by nearly a linear factor in some cases. We also address the ray cutting version of the problem and present an approximation algorithm that computes in almost linear time a finite sequence of ray cuts that is guaranteed to be an $O(\log^2 n)$-factor approximation of an optimal solution. For this, we first give an $O(\log n)$-approximation algorithm for cutting out an n-vertex convex polygon P from an m-vertex convex polygon Q by a sequence of ray cuts. The running time of this algorithm is $O(n + m)$. Then, we show that when P is ray cuttable and Q is the convex hull of P we can extend the approximation results for guillotine cutting to ray cutting, resulting in an $O(n \log n)$ time algorithm to compute an $O(\log^2 n)$-factor approximation of an optimal ray cutting sequence. Putting things together, when Q is convex, we answer the second question in [3] and the fourth question in [6] by combining the two algorithms above.

2. Preliminaries

Let P and Q be two polygons such that $P \subset Q$. A *guillotine cut* (or *line cut*) is a straight line cut that does not cut through the interior of P and separates Q into a number of pieces, lying on both sides of the cut. A guillotine cut is an *edge cut* if it cuts along an edge of P. After a cut is made, Q is updated to that piece that still contains P.

A *cutting sequence* $\mathcal{S}$ is a sequence of cuts such that after the last cut in the sequence we have $P = Q$ (see Figure 1). An *edge cutting sequence* $\mathcal{S}_e$ is a cutting sequence in which all cuts are edge cuts. For a cutting sequence to exist, P must be convex.

The cost of a cut is the length of the intersection of the cut with Q. An optimal cutting sequence $\mathcal{S}^*$ is a cutting sequence of minimum cost. An optimal edge cutting sequence is denotes as $\mathcal{S}_e^*$.

For a cut C in a cutting sequence $\mathcal{S}$, we denote the cost of C by $|C|$. The cost of $\mathcal{S}$ is denoted as $|\mathcal{S}|$. Let ab be a line segment with endpoints a and b. We use the notation $|ab|$ for the length of ab, and use $|P|$ for the perimeter of the polygon P. The boundary of P is denoted as ∂P and the diameter of P is denoted as $diam(P)$.

Let s_1, s_2 and s_3 be three line segments with pairwise disjoint intersections. Then, s_1, s_2 and s_3 define a unique triangle. We say that the triangle is obtuse if one of its angles is no smaller than $90°$ and we refer to the corresponding angle as an obtuse angle.

A *ray cut* is a cut that originates at infinity, ends in Q and does not cut through the interior of P. A *ray cutting sequence* is a sequence of ray cuts that cut P out of Q. The polygon P is *ray cuttable* if there exist a ray cutting sequence to cut P out of Q. A ray cutting sequence exists if every edge of P can be extended to a ray cut.

The convex hull $CH(P)$ of a polygon P is the smallest convex polygon that contains P. Let P be a simple polygon and let Q be the convex hull of P. Let e be an edge of Q such that $e \notin P$. Then, the edge e and the boundary of P between the endpoints of e and interior to Q define a *pocket* P' of $Q \setminus P$ (see Figure 2).

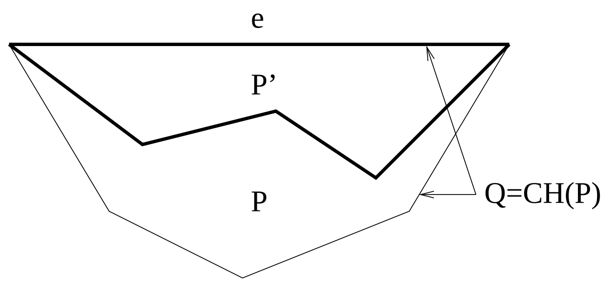

Fig. 2. A pocket P' (in bold lines) defined by ∂P and an edge $e \in Q = CH(P)$.

3. Cutting out polygons with guillotine cuts

Our approach for solving the problem resembles that in [3]. Specifically, the solution involves a separation phase and a carving phase. In the separation phase, three line cuts are used to obtain a triangle which encloses P and has the perimeter a constant factor more than the perimeter of P. In the carving phase, all the cuts are along the edges of P.

3.1. *Separation phase*

For a line cut C, let θ be the unique angle of C in the interval $I = [-90°, 90°]$. In [3], it has been proven that there exist two line cuts C_1 and C_2 with the property that (1) either the angle formed by C_1 and C_2 is such that $|\theta_1 - \theta_2| \in [20°, 160°]$, or C_1 and C_2 are almost parallel and tangent to P on opposite sides, (2) C_1 and C_2 are tangent to P and (3) $|C_1| + |C_2| = O(|\mathcal{S}^*|)$, where $\mathcal{S}^*$ is an optimal cutting sequence. Two such cuts can be found in $O(nm + n \log n)$ time [3].

In this section, we show how to find in $O((n+m)\log(n+m))$ time a pair of cuts C_1 and C_2 of minimum total length $|C_1|+|C_2|$ and satisfying the first two conditions above. Obviously, this implies the third condition, that is $|C_1| + |C_2| = O(|\mathcal{S}^*|)$. The third cut is chosen as in [3], resulting in three line cuts of total length $O(|\mathcal{S}^*|)$.

We reformulate the angle property above for the cuts C_1 and C_2 by changing the definition of the angle θ of a cut C as follows. The angle of the line cut that is parallel to the x-axis and tangent to P from bellow is $0°$. The angle increases gradually as the line cut rotates counter-clockwise along the boundary of P while being tangent to P. Thus, we have that $\theta \in [0°, 360°)$.

The problem now is to find two cuts C_1 and C_2 such that (1) C_1 and C_2 are tangent to P, (2) the angle between C_1 and C_2 is such that $|\theta_1 - \theta_2| \in [20°, 340°]$ and (3) $|C_1| + |C_2|$ is minimized.

Let C be a line cut tangent to P and along the edge $v_{i-1}v_i$ of P, with $1 \le i \le n$ and $v_0 = v_n$, and consider rotating C around v_i until it overlaps with the edge v_iv_{i+1} (v_nv_1 when $i = n$), or it touches a vertex of Q. Then, in this angle interval $\theta \in [\theta_i^1, \theta_i^2]$, the length $|C| = C(\theta)$ of a line cut C is a convex function [1].

We can compute the functions $C(\theta)$ defining the cut length $|C|$ for all the $O(n+m)$ angle intervals in $O(n+m)$ time by rotating C along the boundary of P (similar to the *rotating calipers* technique [8]). The diagram of $C(\theta)$ is a continuous function that consists of $O(n+m)$ convex curves, and it has $O(n+m)$ local minima.

Our algorithm is as follows:

1: Let $\mathcal{M} = \{m_1, m_2, \ldots, m_p\}$ be the list of local minima, where $p = O(m+n)$ is the number of local minima. Let the corresponding angles and cuts be $\mathcal{A} = \{a_1, a_2, \ldots, a_p\}$ and $\mathcal{C} = \{c_1, c_2, \ldots, c_p\}$, respectively. Assume the cuts are such that $a_1 < a_2 < \ldots < a_p$.
2: Find the two cuts $C_1, C_2 \in \mathcal{C}$ of smallest length, with $|C_1| \leq |C_2|$. Let θ_1 and θ_2 be the angles of C_1 and C_2, respectively.
3: Set $l = |C_1| + |C_2|$.
4: **if** $|\theta_1 - \theta_2| \notin [20°, 340°]$ **then**
5:     Set $l = \infty$.
6:     Find all the local minima with angle in $(\theta_1 - 20°, \theta_1 + 20°)$, by a plane sweep from θ_1. (Use $[0°, \theta_1 + 20°) \cup (340° + \theta_1, 360°]$ if $\theta_1 - 20° < 0$ and use $(\theta_1 - 20°, 360°] \cup [0°, \theta_1 - 340°)$ if $\theta_1 + 20° > 360$). We obtain a set $\{m'_1, m'_2, \ldots, m'_q\}$, where $q \leq p$, the corresponding angles are $\{a'_1, a'_2, \ldots, a'_q\}$, with $a'_1 < a'_2 < \ldots < a'_q$, and the cuts are $\{c'_1, c'_2, \ldots, c'_q\}$.
7:     **for** $i = 1$ to q **do**
8:         Find the cuts cb_1 and cb_2 with angles $a'_i - 20°$ and $a'_i + 20°$, respectively. (Use $340° + a'_i$ instead of $a'_i - 20°$ if $a'_i - 20° < 0$, and use $a'_i - 340°$ instead of $a'_i + 20°$ if $a'_i + 20° > 360$).
9:         Find the smallest value m_0 among $(\{m_1, m_2, ..., m_p\} \cup \{cb_1, cb_2\}) \setminus \{m''_1, m''_2, ..., m''_r\}$, where $r \leq p$ and $\{m''_1, m''_2, \ldots, m''_r\}$ are the local minima in $\{m_1, m_2, ..., m_p\}$ with angle in $(a'_i - 20°, a'_i + 20°)$ (resp. $[0°, a'_i + 20°) \cup (340° + a'_i, 360°]$ or $(a'_i - 20°, 360°] \cup [0°, a'_i - 340°)$).
10:         **if** $m_0 + m'_i < l$ **then**
11:             set $C_1 = c'_i$ and $C_2 = c_0$, where c_0 corresponds to the cut of length m_0 and c'_i corresponds to the cut of length m'_i.
12:             set $l = |C_1| + |C_2|$.
13:         **end if**
14:     **end for**
15: **end if**

Theorem 1. *Given two convex polygons P and Q, $P \subset Q$, with n and m vertices, respectively, we can find two cuts C_1 and C_2 tangent to P, with $|\theta_1 - \theta_2| \in [20°, 340°]$ and of minimum total length $l_{min} = |C_1| + |C_2|$, in $O((n+m)\log(n+m))$ time.*

Proof. Use the algorithm above. The total running time is $O((n+m)\log(n+m))$ since finding the two smallest cuts in $\mathcal{M}$ takes $O(n+m)$ time and the cost of the for loop is $O((n+m)\log(n+m))$ if we use plane sweep and a priority queue data structure. We next argue that $l_{min} = |C_1| + |C_2|$, as computed by the algorithm above, is of minimum length. Let θ_1 and θ_2 be the angles of the cuts C_1 and C_2, respectively. Let θ_{min} be the angle corresponding to the cut of smallest length. We have $\theta_1 \in (\theta_{min} - 20°, \theta_{min} + 20°)$ and $\theta_2 \notin (\theta_1 - 20°, \theta_1 + 20°)$. Note that θ_1 and θ_2 could only be one of the $O(n + m)$ local minima except that θ_2 could also be at $\theta_1 - 20°$ or $\theta_1 + 20°$, if there is no local minima outside $(\theta_1 - 20°, \theta_1 + 20°)$ or all the local minima outside $(\theta_1 - 20°, \theta_1 + 20°)$ are larger than the cut length at $\theta_1 - 20°$ or $\theta_1 + 20°$. (Above, if $\theta_1 - 20° < 0$ use $[0°, \theta_1 + 20°) \cup (340° + \theta_1, 360°]$ for $(\theta_1 - 20°, \theta_1 + 20°)$ and $340° + \theta_1$ for $\theta_1 - 20°$; if $\theta_1 + 20° > 360$ use $(\theta_1 - 20°, 360°] \cup [0°, \theta_1 - 340°)$ for $(\theta_1 - 20°, \theta_1 + 20°)$ and $\theta_1 - 340°$ for $\theta_1 + 20°$). We check all possible pairs (C_1, C_2) during the plane sweep and maintain the smallest cut length for $|C_1| + |C_2|$ over all such pairs. □

3.2. *Carving phase*

In this section, we affirmatively answer the conjecture in [3]. Let $\mathcal{S}^*$ denote an optimal cutting sequence and let $|\mathcal{S}^*|$ be its cost. Similarly, let $\mathcal{S}_e^*$ denote an optimal edge cutting sequence and let $|\mathcal{S}_e^*|$ be its cost. Let $|P|$ and $|Q|$ denote the perimeter of P and Q, respectively.

Theorem 2. *If P is enclosed in a minimum axis-aligned rectangle Q, an optimal edge cutting sequence $\mathcal{S}_e^*$ is a constant, $(2.5 + |Q|/|P|)$-factor approximation of an optimal cutting sequence $\mathcal{S}^*$ for cutting P out of Q.*

Proof. We construct an edge cutting sequence that is a $(2.5 + |Q|/|P|)$-factor approximation of $\mathcal{S}^*$ as follows. For every optimal cut $C^* \in \mathcal{S}^*$, in order, if C^* is an edge cut then we add it to the edge cutting sequence. Otherwise, C^* is tangent to a vertex v of P and we add to the edge cutting sequence two cuts that are along the two edges of P incident at v (see Figure 3).

Consider an optimal cut C^* at a vertex v of P. Then, the endpoints of C^* are either on the boundary of Q or they are on some previous cuts of $\mathcal{S}^*$. Let s_1 and s_2 be the two line segments (of Q or $\mathcal{S}^*$) on which the endpoints a and b of C^* lie. Then, C^* is opposite to an obtuse angle ($\geq 90°$) in the triangle defined by C^*, s_1 and s_2, as illustrated in Figure 3. Let C_1 and C_2 be the two edge cuts at v that replace the optimal vertex cut C^* at v. (The endpoints of C_1 and C_2 are on s_1 and s_2. It will become clear below that C_1 and C_2 give upper bounds for the actual edge cuts we construct.) Then, v splits C_1 into two line segments, cv and vd, where cv is inside the obtuse triangle defined by C^*, s_1 and s_2. Similarly, v splits C_2 into two line segments, ev and vf, where vf is inside the obtuse triangle defined by C^*, s_1 and s_2. Clearly, $|cv| + |vf| < |ab|$ so at least one of $|cv|$ and $|vf|$ is smaller than half the length of the optimal cut C^*. Assuming $|cv| \leq |vf|$ we have $|cv| < |C^*|/2$. Then, the edge cuts C_1 and C_2 are done such that C_1 precedes C_2 and the cost of these two edge cuts is $|cv| + |ev| + |vd| < |C^*|/2 + |ev| + |vd|$. Thus, from now on, we focus on bounding the lengths of ev and vd. These lengths give upper bounds (see below) on the components of the two edge cuts performed along the edges incident to v, that replace the vertex cut at v. Let Q_e be the piece of Q that contains P

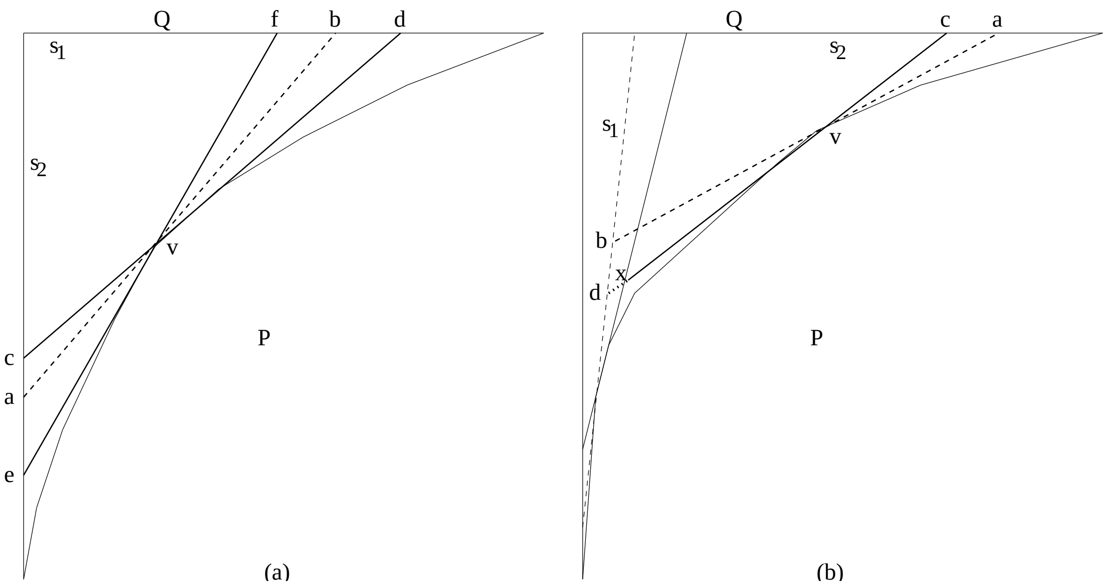

Fig. 3. Illustration of the extra cut cost, where dash lines are optimal cuts and continuous lines are edge cuts. In (b), the current cuts are drawn with bold lines.

after the two edge cuts are performed. Note that Q_e is included in the piece Q_v of Q that contains P after the optimal vertex cut C^* is performed. Since after a cut is performed the problem is divided into two independent subproblems, we need focus only on one of the two subproblems. Consider bounding the length of vd. We have two cases, depending on whether s_1 is on the boundary of Q (Figure 3 (a)) or it is part of some optimal cut preceding C^* in the optimal cutting sequence (Figure 3 (b)). Let the cost of the optimal cut vb which goes through the vertex v of P be L_v, and let the cost of the corresponding edge cut in the edge cutting sequence we construct be L_e. The difference of L_e and L_v is the extra cost for the edge cut over the vertex cut. In the first case, bd is on the boundary of Q. Then, $L_e - L_v < E_Q$, where $E_Q = |bd|$. In the second case, if d is on an optimal vertex cut C'^*, note that the line segment vd must intersect an edge cut in the edge cutting sequence constructed so far (that edge cut is associated with C'^*). Let x be the intersection point. Thus, $|vx| \leq |vd|$ and the edge cut adds length $|vx|$ to the edge cutting sequence. The line segment bd is on a previous optimal cut in $\mathcal{S}^*$ and $L_e - L_v = |vx| - |vb| \leq |vd| - |vb| \leq |bd| = E_v$.

Since subproblems are independent, it is easy to see that for any two line segments l_1 and l_2, which are on the boundary of Q or on an optimal cut and correspond to some values E_Q or E_v in the construction above, with $l_1 \neq l_2$, we have $l_1 \cap l_2 = \phi$, that is, they do not overlap. Then, the total extra cost is bounded by $|\mathcal{S}^*| + |Q|$.

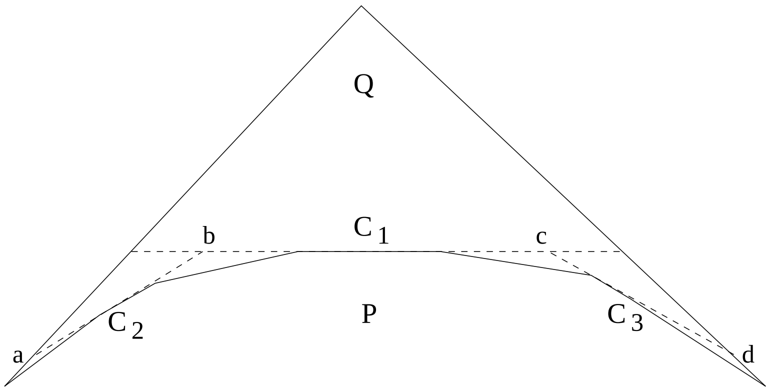

Fig. 4. Illustration of the upper bound in carving phase.

Let $\mathcal{S}_e$ be the edge cutting sequence constructed. We have:

$$|\mathcal{S}_e| < |\mathcal{S}^*|/2 + |\mathcal{S}^*| + (|\mathcal{S}^*| + |Q|) = 2.5|\mathcal{S}^*| + |Q|$$

and thus

$$|\mathcal{S}_e|/|\mathcal{S}^*| = 2.5 + |Q|/|\mathcal{S}^*| < 2.5 + |Q|/|P|$$

To end the proof we note that the cost of an optimal edge cutting sequence $\mathcal{S}_e^*$ satisfies $|\mathcal{S}_e^*| \leq |\mathcal{S}_e|$. □

Since an optimal edge cutting sequence $\mathcal{S}_e^*$ can be computed in $O(n^3 + m)$ time, by combining Theorem 1 and Theorem 2 we obtain the following result.

Theorem 3. *Given two convex polygons P and Q, $P \subset Q$, with n and m vertices, respectively, an $O(1)$-factor approximation of an optimal cutting sequence for cutting P out of Q can be computed in $O(n^3 + (n+m)\log(n+m))$ time.*

Lemma 1. *Given two convex polygons P and Q, $P \subset Q$, with n and m vertices, respectively, an $O(\log n)$-factor approximation of an optimal cutting sequence can be computed in $O((n+m)\log(n+m))$ time.*

Proof. We first proceed with the separation phase, which takes $O((n+m)\log(n+m))$ time. Next, assume that Q is the minimum axis-aligned rectangle enclosing P. For this part, our approach is similar to that in [3]. However, unlike in [3], where this step takes $O(n \log n)$ time, we perform this step in $O(n)$ time. Consider one of the four subproblems (quadrants) defined by the tangency points of Q to P (see Figure 4). For the other ones the analysis is similar. We refer to the boundary of P for this subproblem as P. We use $O(\log n)$ recursive steps. In the first step, we find the median edge of the edges of P and make a line cut C_1 along that edge. Thus, we have $|C_1| \leq |Q|$. The cut C_1 defines two independent subproblems. In the next step, we find the median edge in each of the two subproblems and make two

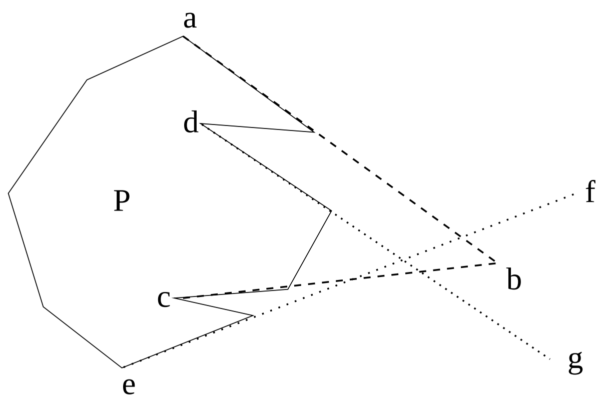

Fig. 5. Illustrating that dynamic programming fails on non-convex polygons.

line cuts C_2 and C_3. Clearly, $|C_2| + |C_3| \leq |Q|$ (they are part of a convex chain between P and Q, e.g. chain *abcd* in Figure 4). The same method is performed until all edges are cut. In each recursive step the cutting cost is no more than $|Q|$. There are $O(\log n)$ steps, so the total cost is $O(|Q| \cdot \log n) = O(|P| \cdot \log n)$. The edges of P are in sorted order and the median edge of P can be found in $O(1)$ time. Thus, the total time for the carving phase is $O(n)$. □

4. Cutting out polygons with ray cuts

In this section we consider the ray cutting version of the problem: Given a convex polygonal piece of material Q with a ray cuttable polygon P drawn on it, cut P out of Q by a sequence of ray cuts in the cheapest possible way.

We first observe that the dynamic programming algorithm in [6] for computing an optimal edge cutting sequence does not work in the case of ray cuts. The dynamic programming algorithm is based on the following fact: if we consider a pair of cuts C_1 and C_2 then, after these two cuts are made, the cuts on the boundary of P between C_1 and C_2, in clockwise order, are independent of the cuts between C_1 and C_2 taken in counter-clockwise order. However, in general this is not true for a ray cutting sequence. Consider Figure 5 and suppose we have two cuts ab and bc followed by other two cuts dg and ef. Then, the length of the cut dg, which is on the boundary of P between ab and bc in clockwise order, depends on whether ef is made before or after dg, where the cut ef is between the cuts ab and bc in counter-clockwise order.

Our solution for approximating an optimal ray cutting sequence has two key steps. In the first step we show how to cut out a convex polygon (i.e., the convex hull of P) by a sequence of ray cuts that is a good approximation of an optimal ray cutting sequence. As with the line cutting problem, this step has two phases: a separation phase and a carving phase. In the second step, Q is the convex hull of P

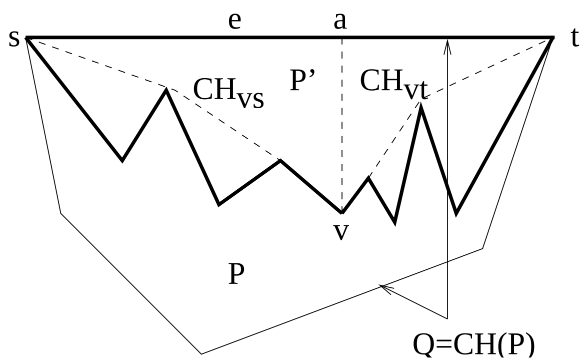

Fig. 6. A pocket P' of P, CH_{vs} and CH_{vt}.

and we show how to cut out a pocket P' of P, where a pocket is defined by an edge e of Q, $e \notin P$, and the boundary of P interior to Q and between the endpoints of e (see Figure 6).

We mention that we do not address the existence of a finite optimal ray cutting sequence. This remains an open problem.

4.1. *Cutting out convex polygons with rays*

We first give a number of properties of an optimal ray cutting sequence when Q and P are convex polygons. Then, we show how to cut out P from Q efficiently. As in the line cutting problem, we use a two phase algorithm for cutting out a convex polygon P from a convex polygon Q: a *separation phase* and a *carving phase*. In the separation phase, three ray cuts are made. After those cuts, P is enclosed by a triangle Q' such that $diam(Q')/diam(P) = O(1)$, where $diam(P)$ (resp., $diam(Q')$) denotes the diameter of P (resp., Q'). In the carving phase, we cut P out of Q' by a sequence of cuts along the edges of P. Thus, we only need to discuss the separation phase.

4.1.1. *General Properties*

Assume that Q and P are both convex polygons. Then, we have the following lemmas.

Lemma 2. *All ray cuts in an optimal ray cutting sequence must end on ∂P or on another ray cut.*

Proof. The proof is by contradiction. Suppose an optimal ray cut r_1 does not end on ∂P, nor it ends on another optimal ray cut. There are two possible cases for r_1.

Case 1. The ray r_1 does not intersect with other ray cuts (see Figure 7 (a)). Obviously we can discard r_1, since r_1 does not cut off any part of Q and it has no contribution in cutting P out of Q.

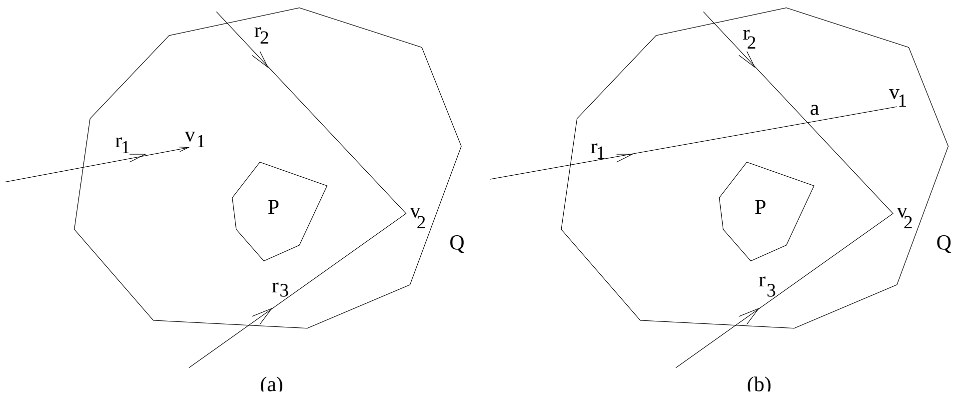

Fig. 7. Illustrating that a ray cut r_1 must end on ∂P or on another ray cut.

Case 2. The ray r_1 intersects with other optimal ray cuts but its endpoint v_1 does not lie on any other optimal ray cut (see Figure 7 (b)). Let r_2 be the last ray intersected by r_1 and let a be the intersection point of r_1 and r_2. Then, we can discard the line segment av_1, since av_1 does not cut off any part of Q and it has no contribution in cutting P out of Q. □

Lemma 3. *All ray cuts in an optimal ray cutting sequence touch P.*

Proof. We make the proof by contradiction. Suppose that there is an optimal ray cut r_1 that does not touch P. From Lemma 2, we know that r_1 ends on another optimal ray cut (denote it by r_2). Let $v_1 = r_1 \cap r_2$. Consider another optimal ray cut r_4 that ends on r_1 (see Figure 8). We can move r_1 parallel to itself either towards P or away from P and at the same time keep the end point v_1 on r_2. For the ray cut r_4, which ends on r_1 at v_3, we let v_3 move with r_1. The function that gives the total change in length for $|r_1| + |r_4|$ is a linear function. Then, this situation is the same as in the proof that an optimal line cut must touch P [1], and it follows that r_1 either is not necessary or it must touch P. □

4.1.2. *Separation phase*

Let ∂P (resp. ∂Q) denote the boundary of P (resp. Q). Let n be the number of vertices of P and let m be the number of vertices of Q.

Lemma 4. *There exists an $O(n+m)$ time algorithm that finds the closest distance from ∂P to ∂Q. Alternatively, the closest distance between ∂P and ∂Q can be found in $O(m \log n)$ time.*

Proof. Simple and omitted. □

Let pq be the segment that gives the closest distance, where p is a vertex of P and q is on some edge $e \in \partial Q$. Let l_q be the supporting line of e. Note that the angle

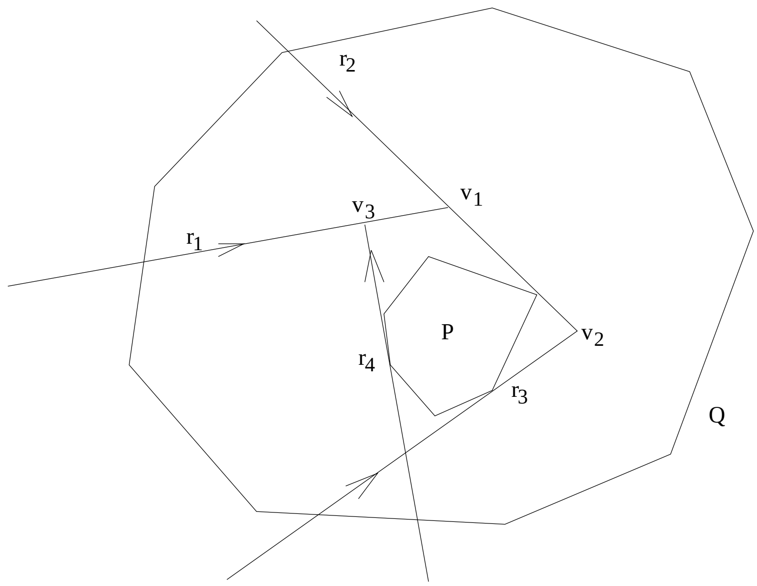

Fig. 8. Illustrating that the ray cut r_1 either touches P or it is not needed.

between pq and l_q is a right angle (see Figure 9). Let l_s be the semi-line originating from q and going through p, splitting P into two sides (left side and right side). Consider two ray cuts bj and bo, tangent to P at the vertices d and e, respectively, and ending at the same point b on l_s, where j and o are the originating points of the two ray cuts on l_q. Assume that point b moves along l_s and let $C_b = |bj| + |bo|$. Then, we can compute the minimum value C_{bmin} of C_b in $O(n)$ time.

Lemma 5. *Let C_{bmin} be defined as above and let $\mathcal{S}^*$ be an optimal ray cutting sequence. Then, C_{bmin} is a constant factor approximation of $|\mathcal{S}^*|$, with $C_{bmin} \leq 9 \cdot |\mathcal{S}^*|$.*

Proof. Let l'_q be the line tangent to P and parallel to l_q, such that P is between l_q and l'_q, and refer to Figure 9. The line l'_q is also orthogonal to l_s. Let a, h and i be the intersection points of l'_q with l_s, bj and bo, respectively. Let $D = diam(P)$, let $|ah| = x$ and let $D' = |hi|$. Then, $|ai| = D' - x$ and $D' \leq D$. Let $|aq| = H$, $|ab| = y$, $l_1 = |bh|$, $l_2 = |hj|$, $l_3 = |bi|$ and $l_4 = |io|$. We then have:

$$\frac{l_1 + l_2}{H + y} = \frac{l_1}{y}$$

$$l_1 + l_2 = (H + y)\frac{l_1}{y}$$

$$\frac{l_3 + l_4}{H + y} = \frac{l_3}{y}$$

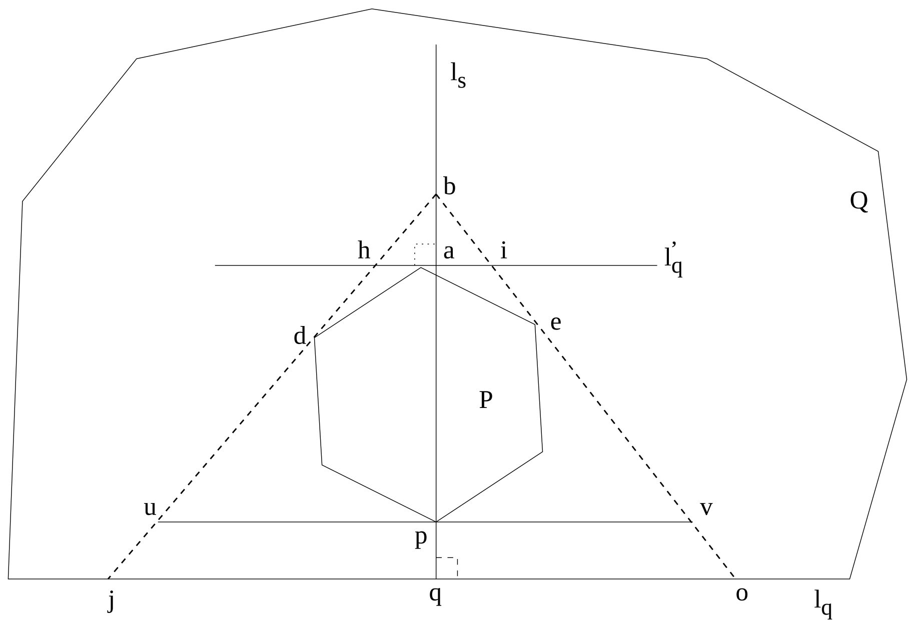

Fig. 9. Illustration of the ray cut construction.

$$l_3 + l_4 = (H + y)\frac{l_3}{y}$$

$$C_{bmin} \leq C_b = l_1 + l_2 + l_3 + l_4 = \frac{H + y}{y}(l_1 + l_3) = (\frac{H}{y} + 1)(l_1 + l_3).$$

$C_{bmin} \leq (\frac{H}{y} + 1)(2y + D') \leq (\frac{H}{y} + 1)(2y + D)$, since $l_1 \leq x + y$, $l_3 \leq y + D' - x$ and $D' \leq D$. Let

$$f(y) = (\frac{H}{y} + 1)(2y + D).$$

Then, $f(y)$ is convex and to minimize it we set $\frac{\partial f}{\partial y} = 0$ and obtain $y = \sqrt{\frac{HD}{2}}$. We then have

$$C_{bmin} \leq (\frac{H}{\sqrt{\frac{HD}{2}}} + 1)(2\sqrt{\frac{HD}{2}} + D)$$

$$= (\sqrt{\frac{2H}{D}} + 1)(\sqrt{2HD} + D)$$

Since $H = |aq| = |ap| + |pq|$, $|pq| \leq |\mathcal{S}^*|$ and $|ap| \leq D \leq |\mathcal{S}^*|$, we have $H \leq 2 \cdot |\mathcal{S}^*|$. Then,

$$C_{bmin} \leq (\sqrt{\frac{4 \cdot |\mathcal{S}^*|}{D}} + 1)(\sqrt{4 \cdot |\mathcal{S}^*| \cdot D} + D)$$

$$= 4 \cdot |\mathcal{S}^*| + 4\sqrt{|\mathcal{S}^*| \cdot D} + D \leq 9 \cdot |\mathcal{S}^*|$$

as $D \leq |\mathcal{S}^*|$. □

Consider the line tangent to p and parallel to l_q, and let u and v be the two intersection points of this line with bj and bo. After the cuts bj and bo for C_{bmin} are made, to cut out P we cut along the line segment uv Obviously, $|uv| \leq |bj| + |bo| \leq 9 \cdot |\mathcal{S}^*|$. Thus, the total cost for the separating phase is $|uv| + |bj| + |bo| \leq 18 \cdot |\mathcal{S}^*|$, which is a constant factor from the optimal solution. We then have the following theorem.

Theorem 4. *Given two convex polygons P and Q, $P \subset Q$, with n and m vertices, respectively, a cutting sequence $\mathcal{S}$ of cost $|\mathcal{S}| \leq 18 \cdot |\mathcal{S}^*|$, where $\mathcal{S}^*$ is an optimal ray cutting sequence, to cut out a triangle Q' such that $P \subset Q'$ and $|Q'|/|P| = O(1)$ can be found in $O(m+n)$ time.*

4.2. *Cutting out pockets by rays*

Let $CH(P)$ be the convex hull of P. Let s and t be the end vertices of an edge $e \in CH(P)$ defining a pocket P' of P. For a vertex $v \in P$ that is inside the pocket P', let CH_{vs} (resp. CH_{vt}) be the portion of the boundary of the convex hull of the vertices of P' between v and s (resp., between v and t) that lies in the pocket P' (see Figure 6). Note that the shortest paths from v to s and t are the same as CH_{vs} and CH_{vt}.

Observation 1. Assume that the shortest paths CH_{vs} and CH_{vt} for some vertex v in P' are available and refer to Figure 10. Let l be a vertex of P between v and s and let r be a vertex of P between v and t. Then, the shortest path CH_{rs} (resp., CH_{lt}) does not intersect with the shortest path CH_{vs} (resp., CH_{vt}), except for some overlap. Consequently, the subproblems on the pockets defined by CH_{vs} are independent of the subproblems on the pockets defined by CH_{vt} (when CH_{vs} and CH_{vt} are seen as the boundary of Q for the subproblems).

Let P_{st} denote the boundary of P inside the pocket P'. The observation below can be easily obtained from Theorem 1 in [2] and the property that, for a ray cuttable polygon P, every edge of P is extendible to a ray.

Observation 2. If P is ray cuttable then the boundary of P inside a pocket P' has a (multi) funnel-like structure (see Figure 11).

For a vertex v of P_{st}, with CH_{vs} and CH_{vt} available, we can use the simplified edge cutting algorithm for the carving phase in Section 3.2, with some modifications, to cut out CH_{vs} and CH_{vt} from the pocket of v (refer to Figure 12). We first make a ray cut va, where va is the shortest straight line segment from v to the segment st that has empty intersection with P. Such a ray cut always exists since the polygon P is ray cuttable, and it can be found in constant time. After the cut va, the ray cuts we perform for CH_{vs} or CH_{vt} originate on st and do not cross va. Consider

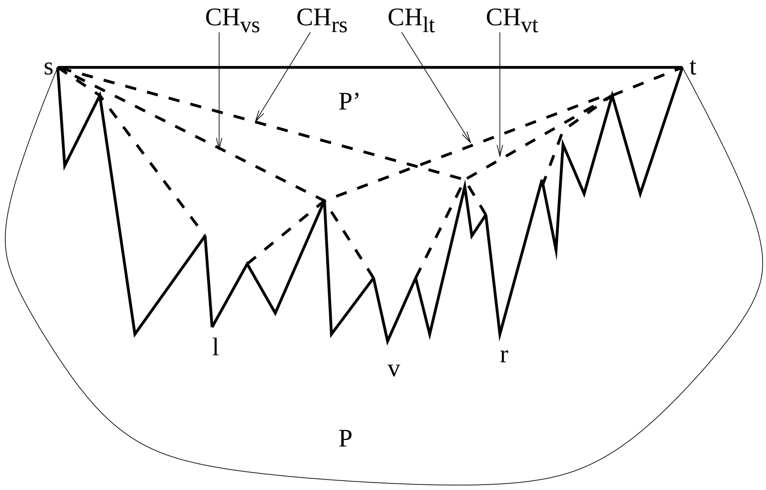

Fig. 10. Illustrating the shortest paths for the subproblems at v, l and r.

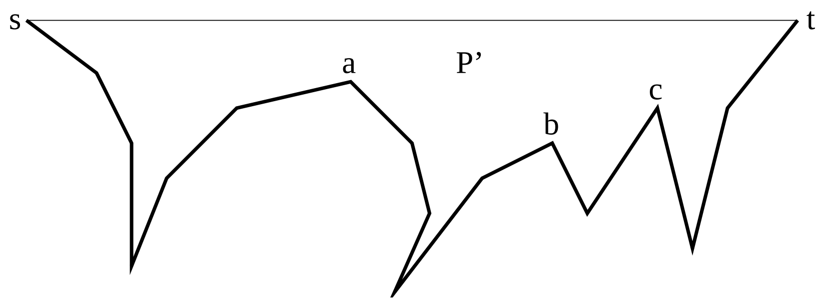

Fig. 11. Illustrating the funnel-like structure of P within a pocket P'. Here, P' has four funnels, P_{sa}, P_{ab}, P_{bc}, and P_{ct}.

cutting out the polygonal line CH_{vs} using the simplified edge cutting algorithm for the carving phase in Section 3.2 (we can assume Q is the triangle defined by sa, av and vs and P is $CH_{vs} \cup \{vs\}$). Then, every edge cut in an edge cutting sequence of the boundary of CH_{vs} that intersects with va corresponds to a ray with the property above. For example, the edge cut fb in Figure 12, along the edge cd, intersects va at b and thus the corresponding ray cut along fb originates at infinity and ends at b.

Theorem 5. *If Q is the convex hull of P, a ray cutting sequence $\mathcal{S}$ of total cost $O(|P|\log^2 n)$ can be computed in $O(n \log n)$ time. The cost of $\mathcal{S}$ is then $|\mathcal{S}| = O(|\mathcal{S}^*|\log^2 n)$, where $\mathcal{S}^*$ is an optimal ray cutting sequence.*

Proof. The second part of the theorem is obvious, since $|P| = O(|\mathcal{S}^*|)$. For the

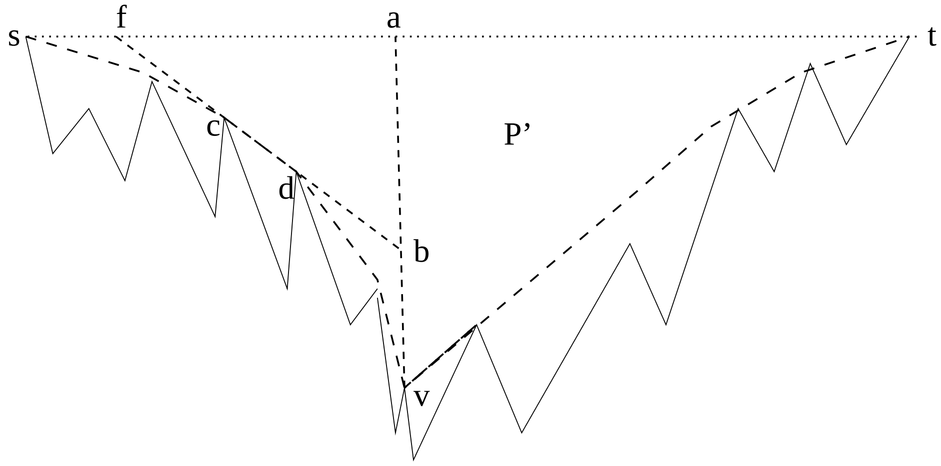

Fig. 12. The shortest paths from v to s and t, and the ray cut va.

first part, consider a pocket P' of P defined by a segment st as above and assume that st is horizontal, with P' below st. Recall that P_{st} denotes the portion of ∂P in P'. To cut out P' we first find the middle vertex v of P_{st}, compute va and the shortest paths CH_{vs} and CH_{vt} from v to s and t, and make the ray cut av. We then cut out CH_{vs} and CH_{vt} using edge cuts. Note that the polygon defined by CH_{vs}, CH_{vt} and the line segment st is cut out of P', and P' is separated into two parts: a left part, P'_l, and a right part, P'_r. Let v_l be the middle vertex of P_{sv} and let v_r be the middle vertex of P_{vt}. In the next step, we cut out $CH_{v_l s}$, $CH_{v_l t}$, $CH_{v_r s}$, and $CH_{v_r t}$ using edge cuts, and proceed recursively until P' is cut out.

Consider a subproblem for P' at some level of recursion (obtained after cutting out some CH_{us}, with u a vertex of P_{st}). Let $P_{s't'}$ be the boundary of P for this subproblem, where s' and t' are the end vertices of $P_{s't'}$. Let w be the middle vertex of $P_{s't'}$. Let wa' be the shortest length cut from w to st that has empty intersection with $P_{s't'}$ (and thus with P_{st}). Finally, let Q' be the portion of Q containing $P_{s't'}$, and let $\partial(Q' \setminus P_{s't'})$ be the boundary of Q' between s' and t' that is different from $P_{s't'}$. Then, since our cuts are along shortest paths, it follows from Observation 1 that either $\partial(Q' \setminus P_{s't'})$ is the line segment $s't'$ (see Figure 13 (a)) or it has exactly two line segments, $s'x$ and xt' (see Figure 13 (b)).

We claim that both $|wa'|$ and $|\partial(Q' \setminus P_{s't'})|$ are about the same size as $|P_{s't'}|$. Clearly, $|wa'|/|P_{s't'}| = O(1)$ if $|\partial(Q' \setminus P_{s't'})|/|P_{s't'}| = O(1)$. Then, it is enough to show that $|\partial(Q' \setminus P_{s't'})|/|P_{s't'}| = O(1)$. If $\partial(Q' \setminus P_{s't'}) = s't'$ the claim holds. If $\partial(Q' \setminus P_{s't'}) = s'x \cup xt'$, then $s'x$ and xt' are on some shortest paths to s and t. Since P is ray cuttable, all edges of $P_{s't'}$ have slopes between the slope of $s'x$ and the slope of xt'. To see this is the case, assume there is an edge of $P_{s't'}$ with slope outside this range. Its supporting line intersects $s'x$, or xt', or both. Assume it intersects $s'x$ (the

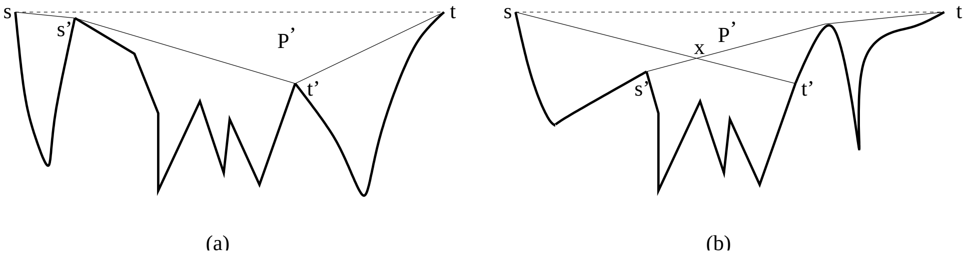

Fig. 13. (a) $\partial(Q' \setminus P_{s't'})$ is a line segment and (b) $\partial(Q' \setminus P_{s't'}) = s'x \cup xt'$.

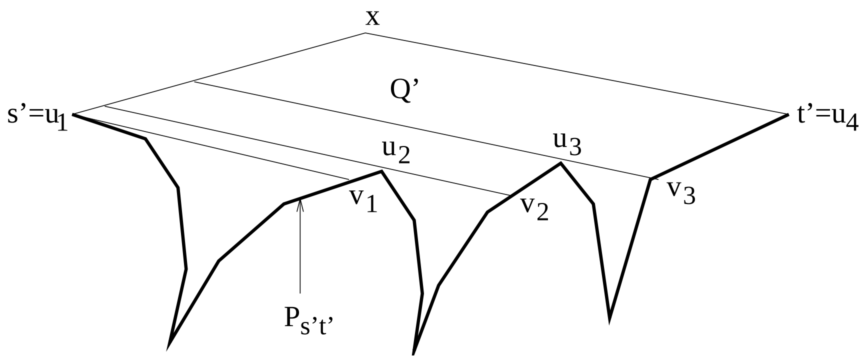

Fig. 14. Illustrating the bound on xt'.

case for xt' is similar). Then, the intersection point with $s'x$ is within the triangle $xs's$ (see Figure 13 (b)). If the line intersects the line segment sx, the slope of the edge is between the slope of $s'x$ and the slope of xt', a contradiction. If it intersects $s's$, then it must also intersect $P_{ss'}$ and thus the edge is not ray cuttable in this direction. In the other direction, either the line intersects $P_{s't'}$ or it intersects xt'. If it intersects $P_{s't'}$ then the edge is not ray cuttable, a contradiction. If it intersects xt', then it either has an intersection with $P_{t't}$, a contradiction, or it intersects the line supporting $s'x$ at some point between x and its tangency point with $P_{t't}$. The later case is not possible however, since it implies there are two intersection points between the line supporting the edge and the line supporting $s'x$.

In what follows, we prove that $|xt'| < |P_{s't'}|$. For Q', consider the visibility polygon $Vis(s'x)$ of $s'x$ along the direction of xt', obtained by projecting $s'x$ onto $P_{s't'}$ along the direction of xt'. Let v_1 be the projection point of s' (the projection point of x is t'). We obtain a set of points $\{s' = u_1, v_1, u_2, v_2, \ldots, u_{k-1}, v_{k-1}, u_k = t'\}$ with the property that u_i is a vertex of $P_{s't'}$, $v_i \in P_{s't'}$ and the interior of the line segment u_iv_i does not intersect $P_{s't'}$ (see Figure 14). Let $P'_{s't'} = \partial(Vis(s'x)) \setminus (s'x \cup xt')$. Then, $P'_{s't'}$ has a stair-case like structure. Since the slope of each edge on $P_{s't'}$ is between the slopes of $s'x$ and xt', it follows that all edges on $P_{s't'}$ between v_i

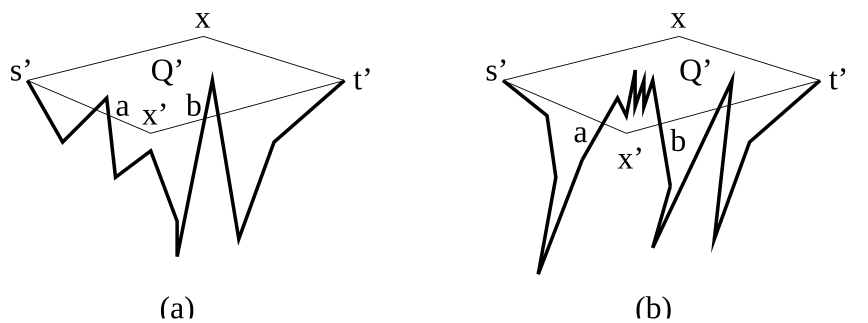

Fig. 15. The two possible cases for x'.

and u_{i+1} have slope larger than the slope of $s'x$, for $i = 1, 2, \ldots, k-1$. This implies that $|xt'| \leq |u_1v_1| + |u_2v_2| + \ldots + u_{k-1}v_{k-1}$, and thus $|xt'| \leq |P_{s't'}|$. The proof that $|s'x| < |P_{s't'}|$ is similar and thus we obtain that $|\partial(Q' \setminus P_{s't'})| = |s'x| + |xt'| < 2|P_{s't'}|$. This implies that the cost to cut out CH_{ws} and CH_{wt} with the simplified edge cutting algorithm for the carving phase in Section 3.2 is $O(|P_{s't'}| \log n_{s't'})$, where $n_{s't'}$ is the number of vertices of $P_{s't'}$.

Although it is not needed for the proof of this theorem, a more careful analysis of the approach above for bounding $|s'x| + |xt'|$, with some small changes, would reveal that in fact $|s'x| + |xt'| < |P_{s't'}|$. We prove this stronger property in a different way in Lemma 6 below.

There are $O(\log n)$ recursion levels and at each level the cost of the cut is then $O(|P'| \log n)$, if we use the edge cutting algorithm in the carving phase of Lemma 1. Thus, the total cost for cutting out the pocket P' from Q is $O(|P'| \log^2 n)$.

We can compute the shortest paths CH_{vs} and CH_{vt} for the vertices of P' in linear time [4]. The time complexity for cutting out CH_{vs} and CH_{vt} is $O(n_{st})$, where n_{st} is the number of vertices of P'. We obtain the recurrence $T(n_{st}) = 2T(\frac{n_{st}}{2}) + O(n_{st})$, which solves for $O(n_{st} \log n_{st})$. Then, over all the pockets of P the total time is $O(n \log n)$. □

Lemma 6. *For a subproblem on $P_{s't'}$ as in Theorem 5, $|\partial(Q' \setminus P_{s't'})| \leq |P_{s't'}|$.*

Proof. If $\partial(Q' \setminus P_{s't'}) = s't'$ the lemma is trivially true. Then, assume $\partial(Q' \setminus P_{s't'}) = s'x + xt'$, and consider the parallelogram $s'xtx'$, with x' on the same side of $s't'$ as $P_{s't'}$ (see Figure 15). There are two cases, depending on whether $x' \in Q'$ or not. Let a be the intersection point of $s'x'$ and $P_{s't'}$ that is closest to x'. Similarly, let b be the intersection point of $x't'$ and $P_{s't'}$ that is closest to x'. If $x' \in Q'$ (Figure 15 (a)), then $|P_{s'a}| \geq |s'a|$, $|P_{ab}| \geq |ax'| + |x'b|$ and $|P_{bt'}| \geq |bt'|$, and we obtain that $|P_{s't'}| \geq |s'x'| + |x't'| = |s'x| + |xt'|$. If $x' \notin Q'$ (Figure 15 (b)), then $|P_{s'a}| \geq |s'a|$ and $|P_{bt'}| \geq |bt'|$. To bound $|ax'| + |x'b|$ by $|P_{ab}|$ we repeat the process above, with Q', $P_{s't'}$, and x now being $P_{ab} \cup ax' \cup x'b$, P_{ab}, and x', respectively, until $x' \in Q'$. We conclude the proof by observing that the number of steps to reach $x' \in Q'$ is bounded by the number of vertices of the original problem on $P_{s't'}$. □

4.3. *Putting All Together*

By putting together the results in Section 4.1 and Section 4.2 we obtain the following results.

Lemma 7. *Given a ray cuttable polygon P with n vertices on a convex polygon Q with m vertices, in $O(n^3 + m)$ time one can compute a ray cutting sequence that is an $O(\log^2 n)$-factor approximation of an optimal ray cutting sequence.*

Proof. We first use the separation algorithm in Section 4.1.2 to obtain a triangle Q' containing P such that all edges of Q' touch P and $diam(Q') = O(diam(P))$. This takes $O(m + n)$ time. Note that from the way we upper bound C_{bmin} in Section 4.1.2, it also follows that $C_{bmin} \leq 9 \cdot |\mathcal{S}^*|$, where $|\mathcal{S}^*|$ is the optimal cost of cutting out a ray cuttable polygon P from a convex polygon Q. This is true since in the proof of Lemma 5 we only use $|\mathcal{S}^*|$ to upper bound the diameter of P and the closest distance from P to Q. Then, the cost to cut out Q' when P is a ray cuttable polygon is upper bounded by $18 \cdot |\mathcal{S}^*|$, and thus by $O(|\mathcal{S}^*|)$.

In the carving phase, we compute an optimal edge cutting sequence to cut out $CH(P)$ from Q', where $CH(P)$ is the convex hull of P. The optimal edge cutting sequence $\mathcal{S}_e^*$ has cost bounded by $O(|P| \log n)$ and can be found in $O(n^3)$ time (see Section 3.2). We then cut out P from $CH(P)$ using the algorithm for cutting out pockets in Section 4.2. Adding up, the total cutting cost is $O(|\mathcal{S}^*|) + O(|P| \log n) + O(|P| \log^2 n) = O(|\mathcal{S}^*| + |P| \log^2 n) = O(|\mathcal{S}^*| \log^2 n)$ and the running time is $O((n + m) + n^3 + n \log n) = O(n^3 + m)$. □

Theorem 6. *Given a ray cuttable polygon P with n vertices on a convex polygon Q with m vertices, in $O(m + n \log n)$ time one can compute a ray cutting sequence that is an $O(\log^2 n)$-factor approximation of an optimal ray cutting sequence.*

Proof. Same as the proof of Lemma 7, except that in the carving phase, instead of an optimal edge cutting sequence $\mathcal{S}_e^*$, we compute in $O(n)$ time an approximate edge cutting sequence $\mathcal{S}_e$ that has cost $O(|P| \log n)$, as in the carving phase for Lemma 1 (see the end of Section 3.2). Then, the total running time is $O((m+n)+n+n \log n) = O(m + n \log n)$. □

5. Conclusions

In this paper, we have discussed the problem of cutting out polygons with lines and rays. We have answered a number of open problems and have improved over previously known solutions. Specifically: (1) Our algorithm for the separation phase in approximating an optimal line cutting sequence improves over that in [3] by almost a linear factor in some cases; (2) We have proved that for the carving phase an optimal edge cutting sequence is a constant factor approximation of an optimal line cutting sequence, which implies an $O(n^3 + (n + m) \log(n + m))$ time, $O(1)$-factor approximation algorithm for cutting P out of Q, when P and Q are convex polygons with n and m vertices, respectively; (3) We have presented an $O(\log^2 n)$-factor approximation algorithm for cutting P out of a convex polygon

Q, when P is ray cuttable. The running time of the algorithm is $O(n^3 + m)$ or $O(m+n \log n)$, depending on whether we compute an optimal edge cutting sequence or an approximate edge cutting sequence.

References

1. Bhadury, J. and Chandrasekaran, R.: Stock cutting to minimize cutting length. European Journal of Operational Research. **88** (1996) 69–87.
2. Demaine, E.D., Demaine, M.L. and Kaplan,C.S.: Polygons cuttable by a circular saw. Computational Geometry: Theory and Applications. **20** (2001) 69–84.
3. Dumitrescu, A.: An approximation algorithm for cutting out convex polygons. Computational Geometry: Theory and Applications. **29(3)** (2004) 223–231.
4. Guibas, L., Hershberger, J., Leven, D., Sharir, M. and Tarjan, R.E.: Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica. **2(2)** (1987) 209–233.
5. Jaromczyk, J.W. and Kowaluk, M: Sets of lines and cutting out polyhedral objects. Computational Geometry. **25** (2003) 67–95.
6. Overmars, M.H. and Welzl, E.: The complexity of cutting paper. Procs. of the 1st Annual ACM Symposium on Computational Geometry. (1985) 316–321.
7. Pach, J. and Tardos, G.: Cutting Glass. Disc. & Comput. Geom. **24** (2000) 481–495.
8. Toussaint, G.T.: Solving geometric problems with the 'rotating calipers'. Procs. MELECON, Athens, Greece, 1983.