Theoretical Computer Science 508 (2013) 66–73

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Approximation algorithms for cutting a convex polyhedron out of a sphere[☆]

Xuehou Tan[a,*], Gangshan Wu[b]

[a] School of Information Science and Technology, Tokai University, 4-1-1 Kitakaname, Hiratsuka 259-1292, Japan
[b] State Key Lab. for Novel Software Technology, Nanjing University, Hankou Road 22, Nanjing 210093, China

ARTICLE INFO

Keywords:
Combinatorial geometry
Approximation algorithms
Polyhedron cutting
Simple cycle separators

ABSTRACT

For a given convex polyhedron P of n vertices inside a sphere Q, we study the problem of cutting P out of Q by a sequence of plane cuts. The cost of a plane cut is the area of the intersection of the plane with Q, and the objective is to find a cutting sequence that minimizes the total cost. We present three approximation solutions to this problem: an $O(n \log n)$ time $O(\log^2 n)$-factor approximation, an $O(n^{1.5} \log n)$ time $O(\log n)$-factor approximation, and an $O(1)$-factor approximation with exponential running time. Our results significantly improve upon the previous $O(n^3)$ time $O(\log^2 n)$-factor approximation solution.

1. Introduction

About two and a half decades ago, Overmars and Welzl considered the following problem: given a polygonal piece of paper Q with a polygon P of n vertices drawn on it, cut P out of Q by a sequence of "guillotine cuts" in the cheapest possible way [11]. A *guillotine cut* is a line cut that does not cut through the interior of P and separates Q into a number of disjoint pieces, and the cost of a cut is the length of the intersection of the cut with Q. The study of this type of problem is motivated by the application where a given shape needs to be cut out from a parent of material.

After the hardness of the problem (i.e., computing a cutting sequence that minimizes the total cost) was shown by Bhadury and Chandrasekaran [4], the research has recently been concentrated on finding approximation solutions. Particularly, when both P and Q are convex polygons in the plane, several $O(\log n)$ and constant factor approximation algorithms and a PTAS have been proposed [3,5,6,12].

In three dimensions, Jaromczyk and Kowaluk were the first to study the problem of cutting polyhedral shapes with a hot wire cut, and gave an $O(n^5)$ time algorithm that constructs a cutting path, if it exists [8]. Very recently, Ahmed et al. considered the following problem: given a convex polyhedron P of n vertices inside a sphere Q, find a minimum cost sequence of planes to cut Q such that after the last cut in the sequence we have $Q = P$ [1]. Here, the cost of a plane cut is the area of the intersection of the plane with the current polyhedron Q. Their proposed algorithm runs in $O(n^3)$ time and has the cutting cost $O(\log^2 n)$ times the optimal. Whether the approximation factor or the running time can be improved is left as an open problem.

In this paper, we present three new approximation algorithms for finding a minimum cost sequence of planes to cut P out of Q: an $O(n \log n)$ time $O(\log^2 n)$-factor approximation, an $O(n^{1.5} \log n)$ time $O(\log n)$-factor approximation, and an

☆ The work by Tan was partially supported by Grant-in-aid (235000024) for Scientific Research from Ministry of Education, Science and Culture, Japan, and Japanese Society for the Promotion of Science.
* Corresponding author. Tel.: +81 463 58 1211.
E-mail address: tan@wing.ncc.u-tokai.ac.jp (X. Tan).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.03.035

$O(1)$-factor approximation with exponential running time. Our results show a nice trade-off between running time and approximation quality, and give a significant improvement upon the previously known solution [1].

The key to our algorithms is the use of graph-decompositions based on separators. Lipton and Tarjan proved that given an n-node planar graph one can find in linear time a set of nodes of size $O(\sqrt{n})$, whose removal breaks the graph into two pieces each of size at most $2n/3$ [9]. A more structured separator, called the *simple cycle separator*, was later studied by Miller [10]. For a maximal planar graph (every face in a planar embedding is a triangle), one can construct in linear time a simple cycle separator such that the inside and the outside of the cycle each have at most $2n/3$ nodes. See also [2].

From the convexity of P, all the upward (downward) faces of P can be considered as a planar graph. Using Miller's cycle separators, one can cut P out of Q in a divide-and-conquer manner. To attain a good trade-off between running time and approximation quality, we show that a minimum cost *face* cutting sequence, whose cuts are all made along the faces of P, is a constant factor approximation of a minimum cost cutting sequence. It is not known whether there exists a polynomial time algorithm for finding a minimum cost face cutting sequence. But for a given cycle separator, its minimum cost face cutting sequence can be computed by dynamic programming. All these together allow us to give an $O(n^{1.5} \log n)$ time $O(\log n)$-factor approximation solution, which is the main result of this paper.

2. Preliminaries

Assume that a convex polyhedron P of n vertices is completely contained in a sphere Q. A *guillotine cut*, or simply a *plane cut* is a cut that does not intersect the interior of P and divides Q into two convex pieces, lying on both sides of the cut. Particularly, a plane cut is a *face/edge/vertex cut* if it cuts along a face/edge/vertex of P. After a cut is made, Q is updated to the piece containing P. A *cutting sequence* is a sequence of plane cuts such that after the last cut in the sequence we have $P = Q$.

The cost of a cut is the area of the intersection of the cut with Q. Our objective is to find a cutting sequence whose total cost is minimum. Denote by S a cutting sequence and $|S|$ the cost of S. An *optimal* cutting sequence S^* is a cutting sequence whose cost $|S^*|$ is minimum. A *face* cutting sequence, denoted by S_f, is a sequence of plane cuts that are all made along the faces of P. An optimal face cutting sequence S_f^* is a face cutting sequence whose cost $|S_f^*|$ is minimum among all face cutting sequences.

Let f denote a face of the polyhedron P. We will denote by $|f|$ the area of f. Also, we let $|P| = \sum |f|$, for all $f \in P$, i.e., $|P|$ denotes the surface area of P. For two points x, y, we denote by xy the line segment connecting x and y, and $|xy|$ the length of the segment xy.

2.1. Lower bounds

To estimate the cost performance of our approximation algorithms given in the next two sections, we need some lower bounds on $|S^*|$. First, $|S^*| \geq |P|$ trivially holds. The following lower bounds are similar to, but slightly different from, those of [1]. Note that our proofs are relatively simple.

Lemma 1 (*[1]*). *Suppose that the center o of Q is contained in P. Let f denote the face that is closest to o among all faces of P, and let R_1 denote the radius of the intersection (circle) of Q with the supporting plane of f. Then, $|S^*| \geq \pi R_1^2$.*[1]

Proof. First, an optimal cutting sequence contains a face cut along f; otherwise, P cannot be cut out, a contradiction. Assume that the kth cut in $S^*(=C_1, C_2, \ldots, C_k, \ldots)$ is made along f. Denote by H_1 the intersection (circle) of the given sphere Q with the supporting plane of f. Clearly, after C_k is made in S^*, H_1 is completely cut out from Q. See Fig. 1 for an example, where the portion of P having been cut off is shaded. Then, the total cost of $C_1, C_2, \ldots, C_k$ is at least πR_1^2; otherwise, the whole circle H_1 cannot be cut out from Q, a contradiction. The proof is complete. □

Lemma 2. *Suppose that the center o of Q is not contained in P. Let p denote the point of P which is closest to o, and let R_2 denote the radius of the intersection (circle) of Q with the plane, which is perpendicular to the segment op at the point p. Then, $|S^*| \geq \pi R_2^2$.*

Proof. Denote by H_2 the intersection (circle) of the sphere Q with the plane, which is perpendicular to op at p. Clearly, H_2 touches a face/edge/vertex of P, and does not intersect P. To cut out P from Q, the circle H_2 has to be cut out, too. By an argument similar to the proof of Lemma 1, we can then obtain $|S^*| \geq \pi R_2^2$. □

2.2. Outline of the application of cycle separators

Let G denote a planar graph. By assuming that the graph G is triangulated (i.e., every face of G in a planar embedding is a triangle), Miller has shown that there exists a simple cycle separator of size $O(\sqrt{n})$ such that the inside and the outside of the cycle each have at most $2n/3$ nodes. Moreover, this cycle can be found in linear time [10]. Miller's result is usually termed as the *simple cycle separator theorem*.

---

1 Lemma 4 of [1] states that $|S^*| \geq \pi R^2$ in this case, where R denotes the radius of the given sphere Q.

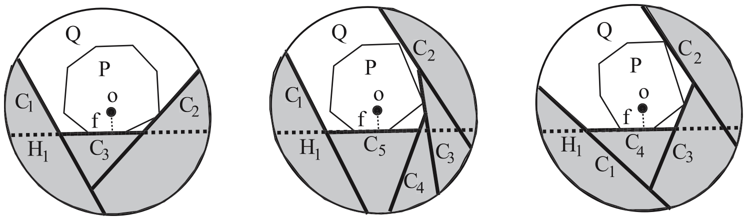

Fig. 1. A planar view of some cuts, with the last one made along the face f.

Let us see how to apply Miller's cycle separators to our problem of cutting P out of Q. Denote by P^+ (P^-) the set of the faces of P, whose outward normal has the positive (negative) z value. Since P is convex, P^+ (P^-) is a planar graph. First, we triangulate every face of P^+. The *dual* of the triangulation of P^+ clearly has $O(n)$ nodes and arcs. Recall that the dual of a planar graph assigns a node to each face and an arc for each edge between adjacent faces. Next, we select $O(\sqrt{n})$ arcs to form a cycle separator, say, $\mathcal{T}$, which partitions the dual graph into two portions with at most two third of the nodes on each side of $\mathcal{T}$. We then perform a sequence of plane cuts along the faces of P^+, which correspond to the nodes of $\mathcal{T}$. (Since each node of $\mathcal{T}$ corresponds to a triangle of P^+, several consecutive nodes of $\mathcal{T}$ may actually contribute to a single face cut.) Two portions of P^+, inside and outside of the face cycle corresponding to $\mathcal{T}$, can recursively be cut out by applying the simple cycle separator theorem to them. Thus, P^+ (as well as P^-) can be cut out from Q by divide-and-conquer.

Subsequent sections will describe how plane cuts are made along the cycle separators. Before going to the detail of our algorithms, we first treat a special case. Since the considered graph in a recursive step has to be triangulated, we assume that it also contains the faces which are newly introduced by the previous separators, but their corresponding cuts needn't be considered. (For instance, the portion of P^+ outside of $\mathcal{T}$ contains the face adjacent to the cycle separator $\mathcal{T}$, but the cut for that face needn't be considered in the step of cutting off the portion of P^+ outside of $\mathcal{T}$.) For these unnecessary faces, to be exact, their nodes have to be deleted from the considered cycle separator. So the nodes of the cycle separator, whose corresponding cuts have to make in a recursive step, actually form a simple cycle or several disjoint path(s). It is sufficient for our algorithms, since what we need is to know an order of the nodes of the separator. In subsequent sections, we assume that this special case is handled well (i.e., all unnecessary nodes have been deleted), and will not mention it any more.

3. An efficient $O(n \log n)$ time approximation algorithm

As in the previous work [1,6], our algorithm consists of two phases: *box cutting phase* and *carving phase*. In the box cutting phase, we cut a bounding box B out of Q such that P is contained in B. (Note that B is used for the worst case analysis, and only part of the box B may actually result.) Then in the carving phase, we further cut P out of B.

Instead of finding a minimum box bounding P used in [1], we will present a simple linear-time algorithm to compute a bounding box B, with $|B| \leq 6|P|$. In the carving phase, we employ the cycle separator theorem to accelerate the process of cutting P out of B.

3.1. Box cutting phase

The following two lemmas are devoted to finding the bounding box B of P and cutting B out of Q.

Lemma 3. *For a convex polyhedron P of n vertices, one can compute in $O(n)$ time a bounding box B such that P is contained in B, with $|B| \leq 6|P|$.*

Proof. First, find two vertices s, t of P such that the z-coordinates of s and t are minimum and maximum, respectively. Without loss of generality, assume that st is parallel to the z-axis; otherwise, we can simply rotate the coordinate axes such that the z-axis is parallel to st. Clearly, it takes $O(n)$ time to compute s and t.

Now, we project all vertices of P into the (x, y) plane vertically, and denote by P' the set of the projected vertices. See Fig. 2. Next, compute two points u, v of P' such that the x-coordinates of u and v are minimum and maximum, respectively. Again, we assume that uv is parallel to the x-axis. Finally, compute two vertices u', v' of P' such that their y-coordinates, denoted by $y(u')$ and $y(v')$, are minimum and maximum, respectively. (We cannot assume that $u'v'$ is parallel to the y-axis, but the value $y(v') - y(u')$ is sufficient for the performance analysis of our algorithm.)

Let B be the minimum axis-aligned box that encloses P. As discussed above, the lengths of B in three axis-directions are $|st|$, $|uv|$ and $y(v') - y(u')$. We show below that $|B| \leq 6|P|$. Denote by $CH(P')$ the convex hull of the point set P' in the (x, y) plane, and denote by $|CH(P')|$ the area of $CH(P')$. Clearly, $|P| \geq 2|CH(P')|$. Since the segment uv is parallel to the x-axis, we have $|CH(P')| \geq (y(v') - y(u'))|uv|/2$. Thus, $|P| \geq (y(v') - y(u'))|uv|$. Since st is parallel to the z-axis, we can similarly obtain $|P| \geq (y(v') - y(u'))|st|$ and $|P| \geq |uv| \cdot |st|$. In summary, we have

$$|B| = 2((y(v') - y(u'))|uv| + (y(v') - y(u'))|st| + |uv| \cdot |st|) \leq 6|P|,$$

as required. □

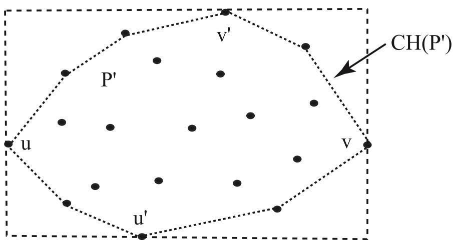

Fig. 2. The orthogonal projection of the polyhedron vertices in the (x, y) plane.

Lemma 4. *For a convex polyhedron P of n vertices inside a sphere Q, one can compute in $O(n)$ time a cutting sequence of cost $O(|S^*|)$, which cuts the box B out of Q.*

Proof. We mainly distinguish two different situations. If the center o of Q is contained in P, we simply make six cuts along all faces of the box B. From the definition of the radius R_1 (see Lemma 1), each of these cuts is of cost no more than πR_1^2. Thus, the cost of this cutting sequence is at most $6|S^*|$. If o is not contained in P, we first make a cut along the plane perpendicular to the segment op, where p is the point of P that is closest to o. Following from Lemma 2, its cutting cost πR_2^2 is no more than $|S^*|$. In the portion of the sphere which contains P but does not contain o, we further make six cuts along all faces of B. Again, from the definition of the radius R_2, each of these six cuts is of cost at most πR_2^2. Hence, the cost of this cutting sequence is at most $7|S^*|$.

Consider the time required to compute the cutting sequence described above. Since P is convex, whether o is contained in P can be determined in $O(n)$ time. In the case that P does not contain o, the point p of P that is closest to o is further computed. Since finding the bounding box B of P also takes $O(n)$ time, the lemma thus follows. □

3.2. Carving phase

Denote by P^+ (P^-) the set of the faces of P, whose outward normal has the positive (negative) z value. Denote by B_{bot} (B_{top}) the bottommost (topmost) face of B. Cutting P^+ (P^-) out of B is then independent of B_{bot} (B_{top}). Let $B^+ = B - B_{bot}$ ($B^- = B - B_{top}$). We describe below how to cut all faces of P^+ out of B^+. (Cutting P^- out of B^- can be done analogously.)

Lemma 5. *The convex polyhedron P can be cut out from the box B in $O(n \log n)$ time by a cutting sequence of cost $O(|S^*| \cdot \log^2 n)$.*

Proof. As in Section 2.2, denote by $\mathcal{T}$ the cycle separator found in the dual of P^+. We employ another divide-and-conquer procedure to compute a face cutting sequence along $\mathcal{T}$. To this end, number all nodes of $\mathcal{T}$ (in clockwise order) from 1 to m, and assume also that node 1 is identical to node m. We then define the *median* node of $\mathcal{T}$ to be the node having the middle number.

First, make a cut along the face corresponding to node 1 of $\mathcal{T}$. Clearly, its cost is at most $|B^+|$. All other nodes (numbered from 2 to $m - 1$) then form a simple path, which we denote by $\mathcal{T}'$. Next, we use $O(\log n)$ recursive steps to compute the cuts for the nodes of $\mathcal{T}'$. In the first step, we find the median node of $\mathcal{T}'$, and make a cut C_1 along its corresponding face of P^+. (Recall that all the triangles on the same plane correspond to a single face cut.) Clearly, $|C_1| \le |B^+|$, and the cut C_1 divides $\mathcal{T}'$ into two subpaths. In the next step, we take the median node from each subpath, and make their corresponding cuts, say, C_2 and C_3. Since C_2 and C_3 are separated by C_1 (and the very first cut), we have $|C_2| + |C_3| \le |B^+|$. This operation is repeatedly performed until all cuts along the nodes of $\mathcal{T}'$ are made. In each recursive step, the cutting cost is no more than $|B^+|$. Therefore, the total cost taken for the face cutting sequence along $\mathcal{T}'$ (as well as $\mathcal{T}$) is $O(|B^+| \cdot \log n) = O(|S^*| \cdot \log n)$.

After the cutting sequence along $\mathcal{T}$ is made, the problem of cutting out the faces of P^+ is partitioned into two subproblems; the inside and the outside of $\mathcal{T}$. Denote by B_1, B_2 the two portions of B^+, which are obtained after the face cutting sequence along $\mathcal{T}$ is done. So $|B_1| + |B_2| \le |B^+|$. We further apply the cycle separator theorem to B_1 (B_2) and perform the face cutting sequence along the found cycle separator. Again, the cutting cost is $O(|B_1| \cdot \log n)$ ($O(|B_2| \cdot \log n)$). Hence, the cutting cost taken in the second recursive step is also $O(|S^*| \cdot \log n)$. In this way, P^+ can be cut out in at most $O(\log n)$ recursive steps, and the cutting cost of each step is $O(|S^*| \cdot \log n)$. The total cost taken by our algorithm is thus $O(|S^*| \cdot \log^2 n)$.

Finally, since a simple cycle separator as well as its face cutting sequence can be computed in linear time, the total time required to cut P out of B is $O(n \log n)$. □

The first result of this paper immediately follows from Lemmas 3 to 5.

Theorem 1. *For a given convex polyhedron P of n vertices inside a sphere Q, an $O(\log^2 n)$-factor approximation of an optimal cutting sequence for cutting P out of Q can be computed in $O(n \log n)$ time.*

4. Constant and $O(\log n)$ factor approximation algorithms

In this section, we first prove a general property of an optimal cutting sequence S^*, i.e., any cut of S^* has to touch P, even in the case that Q is a convex polyhedron. By extending some known planar frameworks to three dimensions, we then present a constant factor and an $O(\log n)$ factor approximation algorithms for cutting P out of Q.

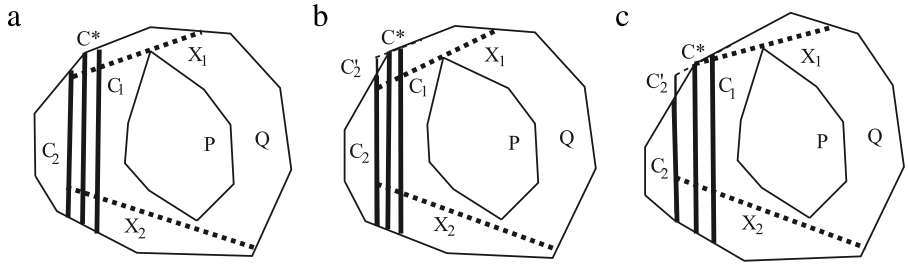

Fig. 3. A planar view of several cuts constructed from C^*, and the pseudo-cut C_2'.

4.1. A general property

Assume that both Q and P are convex polyhedra. Then, we have the following result.

Lemma 6. *Any cut of an optimal cutting sequence S^* for cutting P out of Q has to touch a vertex, an edge or a face of P.*

Proof. The proof is by contradiction [11]. Suppose that C^* is the first cut of S^*, which does not touch P. Clearly, moving or deleting C^* does not change the cost of the cuts before C^*. If no cuts after C^* end on the cut C^*, then C^* can be deleted from S^* without changing the cost of any other cuts; it contradicts the optimality of S^*. Assume below that X is the set of the cuts after C^*, which end on the cut C^*. So moving C^* only changes the cost of the cuts of X.

Assume first that C^* does not contain any vertex of the current polyhedron Q and that no cut of X contains a vertex of the cut C^* (which is a convex polygon). Denote by C_1 (C_2) the cut which is obtained by moving C^* parallel to itself, towards (away from) P, by a very small distance ϵ. (Since ϵ is arbitrarily small, C_1 does not touch P.) Then assume that all cuts of X end on C_2; otherwise, we can further move C_2 (and C_1 as well) towards C^* to make it true. See Fig. 3(a) for a planar view of C^*, C_1 and C_2, where two other cuts X_1, X_2 drawn in a dotted line belong to X. The shape of C_1 (C_2) is similar to that of C^*, if the cut C^* in S^* is replaced by C_1 (C_2). Then, $|S^*|$ cannot be strictly larger than the cost of either *new* sequence, in which the cut C^* is replaced by C_1 or C_2; otherwise, due to the similarity, the cost of the other cutting sequence is strictly smaller than $|S^*|$, a contradiction. Assume now that $|S^*|$ is equal to the cost of two new sequences. Notice that the current polyhedron Q is always convex. So if we keep to move C^* *away from P*, by the same distance ϵ every time, the change in the total cost of the current cut C^* and all the cuts of X, which still end on C^*, is a monotone *decreasing* function. This implies that either a new position of C^* yields an empty set X, or C^* is eventually moved outside Q; a contradiction occurs in either case.

Consider now the situation in which C^* contains some vertices of Q, but no cut of X contains a vertex of C^*. Again, denote by C_1 (C_2) the cut, which is obtained by moving C^* parallel to itself, towards (away from) P, by a very small distance ϵ. Let C_2' denote the intersection of the plane containing C_2, with the supporting planes of the faces of Q intersecting C^* or C_1. See Fig. 3(b). So C_2' is similar to both C^* and C_1. Clearly, some portion of C_2' is outside of Q. We call C_2' a *pseudo-cut*, and the cutting sequence in which C^* of S^* is replaced by C_2' a *pseudo-cutting sequence*. Since $C_2 \subseteq C_2'$, the cost of the pseudo-cutting sequence is at least the cost of the cutting sequence in which C^* is replaced by C_2. Then as discussed above, $|S^*|$ has to be equal to the cost of this pseudo-cutting sequence, too. The argument of keeping to move C^* away from P still works, and a contradiction eventually occurs.

Finally, if some cuts of X contain the vertices of C^*, then by extending these cuts to the plane containing C_2 (Fig. 3(c)), a similar argument also works. This completes the proof. □

4.2. Algorithms in the carving phase

Suppose that the box B has been cut out from the sphere Q, as described in Section 3.1. We focus our attention on the problem of cutting P out of B. For ease of presentation, we still use S^* to represent an optimal cutting sequence for cutting P out of B, and S_f^* an optimal face cutting sequence. The following result shows that S_f^* is a constant factor approximation of S^*.

Lemma 7. *In the carving phase of cutting P out of B, an optimal face cutting sequence S_f^* is an $O(1)$-factor approximation of S^*.*

Proof. The proof is similar to that of its planar counterpart [5], and is presented in the Appendix. □

An optimal face cutting sequence S_f^* can be computed in exponential time, because the number of all face cutting sequences is trivially bounded by $n!$. So we obtain the second result of this paper.

Theorem 2. *For a given convex polyhedron P of n vertices inside a sphere Q, an $O(1)$-factor approximation of an optimal cutting sequence for cutting P out of Q can be computed in exponential time.*

It is not known whether there exists a polynomial time algorithm for computing S_f^*. In the following, we present an $O(\log n)$ factor approximation algorithm. For a cycle separator $\mathcal{T}$, we will denote by $\mathcal{T}_f$ an optimal face cutting sequence, whose cuts are all made along the corresponding faces, or simply, the *faces* of $\mathcal{T}$.

The main idea of our algorithm is to compute $\mathcal{T}_f$ by dynamic programming. (It is essentially the same as Overmars' algorithm for computing an optimal edge cutting sequence for cutting out a convex polygon from another convex polygon in the plane [11].) Assume that at some moment faces i and j of $\mathcal{T}$ have been cut out and the faces in between i and j not. Since all faces of $\mathcal{T}$ form a simple cycle on the surface of P, the order in which we have to cut the faces in between i and j is completely independent from the order in which we cut the other faces. So it helps to precompute the order in which we make the cuts between faces i and j.

Lemma 8. *The convex polyhedron P can be cut out from the box B in $O(n^{1.5}\log n)$ time by a face cutting sequence of cost $O(|S_f^*|\cdot\log n)$.*

Proof. As in the proof of Lemma 5, we also employ a divide-and-conquer procedure to compute the cycle separators, so as to cut P^+ (P^-) out of B^+ (B^-). But, instead of a simple divide-and-conquer method, we employ a dynamic programming algorithm to compute the optimal face cutting sequence for a given cycle separator.

Let $\mathcal{T}^k$ be a cycle separator, which is found in the kth recursive step of our divide-and-conquer algorithm. Denote by P^k (B^k) the portion of P^+ (B^+), which contains $\mathcal{T}^k$ (P^k) in the kth recursive step. (We assume that $P^1 = P^+$ and $B^1 = B^+$. So B_1^2 and B_2^2 (P_1^2 and P_2^2) are the two portions of B^1 (P^1), which are obtained after the cutting sequence $\mathcal{T}_f^1$ is performed.) Denote by n_k the size of P^k, and m_k the size of $\mathcal{T}^k$. From the cycle separator theorem, we have $m_k = O(\sqrt{n_k})$. Observe that B^k is obtained after all the cuts along the faces, whose cuts form the boundary of P^k, are made. The size of B_k is bounded by $O(n_k)$, too.

We first analyze the cost performance of our solution. Since $\mathcal{T}^1$ is only a portion of P^+ (or P^1) and the face cutting sequence $\mathcal{T}_f^1$ is optimal only for $\mathcal{T}^1$, we have $|\mathcal{T}_f^1| \le |S_f^*|$. In the second recursive step, P_1^2 (P_2^2) is completely cut out from B_1^2 (B_2^2). Denote by $\mathcal{T}_1^2$ ($\mathcal{T}_2^2$) the cycle separator of P_1^2 (P_2^2), and $\mathcal{T}_{f1}^2$ ($\mathcal{T}_{f2}^2$) the optimal cutting sequence for $\mathcal{T}_1^2$ ($\mathcal{T}_2^2$). Again, $|\mathcal{T}_{f1}^2|$ ($|\mathcal{T}_{f2}^2|$) is no more than the cost of an optimal face cutting sequence that cuts P_1^2 (P_2^2) out of B_1^2 (B_2^2). Moreover, since B_1^2 and B_2^2 are disjoint in B^+, we have $|\mathcal{T}_{f1}^2| + |\mathcal{T}_{f2}^2| \le |S_f^*|$. In this way, P^+ can be cut out in at most $O(\log n)$ recursive steps, and the cutting cost of each step is $O(|S_f^*|)$. The total cost of our cutting sequence is thus $O(|S_f^*|\cdot\log n)$.

Let us now describe how to compute the cutting sequence $\mathcal{T}_f^k$. Suppose that the faces of $\mathcal{T}^k$ are numbered from 0 to $m_k - 1$ along the surface of P^+. We compute optimal face cutting sequences of all arcs of faces of $\mathcal{T}^k$, in order of length of the arcs, i.e., start with the arc of one face, next arcs of two faces, and so on. Assume that all optimal face cutting sequences for the arcs of length no more than $j-i-1$, have been computed. Consider how to cut out the faces between f_i and f_j. An optimal face cutting sequence consists of a cut along some face f_h between f_i and f_j, followed by at most two optimal cutting sequences; one for all faces between f_i and f_{h-1} and the other for all faces between f_{h+1} and f_j.[2] Trying all possible choices of h clearly gives an optimal face cutting sequence for all faces between f_i and f_j. If we know the cost of the cut along f_h in advance, such a try can be done in constant time. So an optimal face cutting sequence for the arc between (any) two faces f_i, f_j can be computed in $O(j-i)$ time. Since the total number of these arcs is bounded by $O(m_k^2)$, the time required to find $\mathcal{T}_f^k$ is $O(m_k^3)$ or $O(n_k^{1.5})$.

Consider how to precompute the costs of all the cuts along f_h, which are needed in computing $\mathcal{T}_f^k$. A cut along f_h may intersect with the two cuts previously made along f_{i-1} and f_{j+1}, and the surface of B^k. Thus, the cut along f_h can equivalently be obtained by (i) making the cut along f_h inside B^k and (ii) cutting off the portions of that cut, which are outside of the two previously made cuts along f_{i-1} and f_{j+1}.

The costs of all the cuts along f_h can then be computed as follows. First, we compute in $O(n_k)$ time the cost of the cut made along f_h inside B^k. Denote by $C(f_h)$ the found cut, which is a convex region in the supporting plane of f_h, and denote its cost by $c^k(f_h)$. Note that a previously made cut either appears as a line segment in $C(f_h)$ or does not appear at all. These segments in $C(f_h)$ have their endpoints on the boundary of $C(f_h)$ and are all disjoint. See Fig. 4. So we can find them by computing the intersections of $C(f_h)$ with the supporting planes of all other faces of $\mathcal{T}^k$, away from f_h in two opposite directions. Next, we compute all the portions of $C(f_h)$, which are cut off by the previously made cuts, and their corresponding areas as well. This can be done in $O(n_k)$ time by a simple scan of the boundary of $C(f_h)$. Denote by $a^k(f_x)$ the area of the portion of $C(f_h)$, which is cut off by the cut along a face f_x. See Fig. 4 for an example, where some previously made cuts in $C(f_h)$ are shown and the area of the shaded region gives such a value $a^k(f_x)$.

Turn back to the dynamic programming algorithm for computing $\mathcal{T}_f^k$. For any pair (i,j), assume that the face cuts along both f_{i-1} and f_{j+1} have been made. The following cut made along f_h is then of cost $c^k(f_h) - a^k(f_i) - a^k(f_j)$, $h \ne i, j$, (Note that $a^k(f_i)$ or $a^k(f_j)$ may be zero. If the cut along f_h is the very first one in the considered cutting sequence, then both $a^k(f_i)$ and $a^k(f_j)$ are zero.) It clearly takes a constant time to compute $c^k(f_h) - a^k(f_i) - a^k(f_j)$. Thus, the preprocessing time for computing

2 All expressions of face indexes are evaluated modulo m_k.

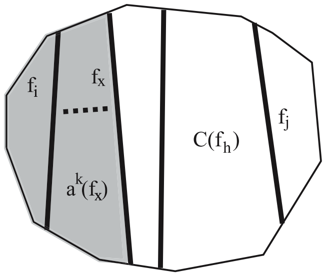

Fig. 4. Illustrating the proof of Lemma 8.

the costs of all the cuts along f_h is $O(n_k + m_k^2)$. Since $0 \leq h \leq m_k - 1$, this preprocessing step takes a total of $O(m_k^3 + m_k n_k)$ or $O(n_k^{1.5})$ time, matching the time required to compute $\mathcal{T}_f^k$.

By now, we can give the time bound of our algorithm. After the face cutting sequence $\mathcal{T}_f^1$ is performed (in $O(n^{1.5})$ time), the problem of cutting the faces of P^+ out of B^+ is further divided into two independent subproblems; either subproblem is of size at most $2n/3$. Thus, solving the recurrence $T(n) = 2T(2n/3) + O(n^{1.5})$ gives us the time bound $O(n^{1.5} \log n)$. □

The main result of this paper follows from Lemmas 4, 7 and 8.

Theorem 3. *For a given convex polyhedron P of n vertices inside a sphere Q, an $O(\log n)$-factor approximation of an optimal cutting sequence for cutting P out of Q can be computed in $O(n^{1.5} \log n)$ time.*

5. Concluding remarks

We have presented three approximation algorithms for computing a minimum cost sequence of planes to cut a convex polyhedron P of n vertices out of a sphere Q. Our algorithms with $O(n^{1.5} \log n)$ running time $O(\log n)$-factor approximation and $O(n \log n)$ running time $O(\log^2 n)$-factor approximation significantly improve upon the previous $O(n^3)$-time $O(\log^2 n)$-factor approximation solution.

We pose several open questions for further research. First, is it possible to find a polynomial-time constant-factor approximation algorithm or even a PTAS for the problem considered in this paper? It may relate to the question of giving a polynomial-time algorithm for computing an optimal face cutting sequence for cutting the convex polyhedron P out of the box B. Although its planar counterpart is true [11], we find it difficult to obtain the same result in 3D. Also, it is an interesting work to develop an approximation algorithm for cutting P out of another convex polyhedron Q. Again, whether the method employed for its planar counterpart [6] can be generalized to 3D remains open.

Acknowledgments

The authors would like to thank two anonymous referees for their valuable comments and suggestions, which help fix some errors occurred in the preliminary versions of this paper (including the one submitted to FAW-AAIM'2011).

Appendix. Proof of Lemma 7

Suppose that S^* is an optimal cutting sequence for cutting P out of B. Then, we can construct a face cutting sequence S_f such that $|S_f| \leq 10|S^*|$. Since $|S_f^*| \leq |S_f|$ holds, the claim of Lemma 7 follows.

For every optimal cut $C^* \in S^*$, in order, if C^* is a face cut, we simply add it to S_f. If C^* is tangent to an edge e of P, then we add to S_f two cuts C_1, C_2 (in this order), which are made along the two faces of P having the common edge e. A portion of C_1 lies outside of C^* as viewed from P, but the whole cut C_2 lies inside of C^*. See Fig. 5 for an example, where the cut C^* along an edge e and its corresponding face cuts C_1, C_2 are shown in Fig. 5(a) and Fig. 5(b), respectively. Since the original polyhedron containing P is the box B, the portion of C_1 lying outside C^* is of area at most $|C^*|$. See Fig. 5(c). Finally, if C^* touches a vertex v of P, we first project all the edges having the common vertex v into the plane containing C^* vertically, and find the two edges e_1, e_2 such that the smaller angle (less than π) between their projections is maximum among all of these angles. Next, we add to S_f the cuts C_{11} and C_{12} (C_{21} and C_{22}), in this order, which are made along the two faces of P having the common edge e_1 (e_2). (Two of these faces may be identical.) Again, the portion of C_{11} (C_{21}) lying outside C^* is of an area of at most $|C^*|$.

Denote by B^* the portion of the box B, which is obtained after the cut C^* in S^* is made, and B_f the portion of B obtained after the cuts corresponding to C^* in S_f are made. It follows from our construction of S_f that $B_f \subseteq B^*$. Thus, S_f is a face cutting sequence too.

Let us now give an upper bound on $|S_f|$. For a cut $C^* \in S^*$, at most four face cuts may have been added to S_f. As described above, the portions of these faces lying outside B^* are of area at most $2|C^*|$. So the total cost for all such portions (lying outside B^*) is at most $2|S^*|$. Denote by C_1' and C_2' (C_{11}', C_{12}', C_{21}' and C_{22}') the portions of the cuts C_1 and C_2 (C_{11}, C_{12}, C_{21} and

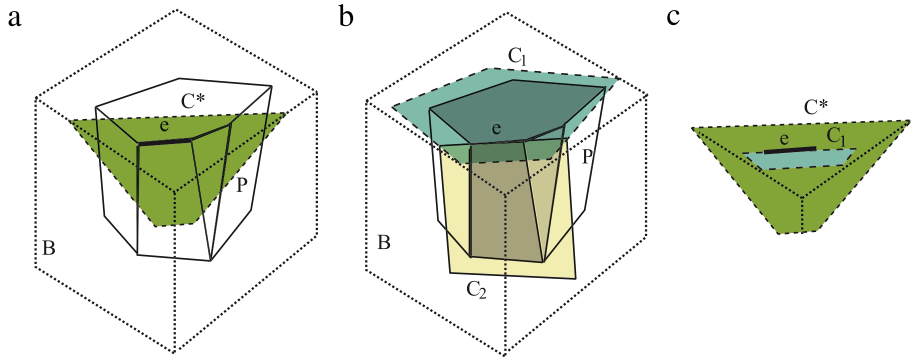

Fig. 5. Illustrating the proof of Lemma 7.

C_{22}), which are contained in B^*. (Actually, we have $C'_2 = C_2$ ($C'_{12} = C_{12}$ and $C'_{22} = C_{22}$).) Let Δ denote the part of B^*, which is exactly cut off by C_1 and C_2 (C_{11}, C_{12}, C_{21} and C_{22}). Since both B^* and P are convex, the *inner* surface of Δ, which consists of C'_1 and C'_2 (C'_{11}, C'_{12}, C'_{21} and C'_{22}), is inward-convex with respect to Δ, and the *outer* surface of Δ, which consists of all other faces (including C^*) of Δ, is outward-convex. Clearly, the area of the inner surface is no more than that of the outer surface. (It can simply be proved by an argument similar to that given for its planar counterpart [7].) So the cost $|C'_1| + |C'_2| - |C^*|$ or $|C'_{11}| + |C'_{12}| + |C'_{21}| + |C'_{22}| - |C^*|$ is no more than the total area of the outer faces f of Δ, excluding C^*. Note that these faces f belong to the surface of B or some cuts of S^*, which are made before C^*. Since Δ is cut off by C_1 and C_2 (C_{11}, C_{12}, C_{21} and C_{22}), all of the faces f considered in the cutting sequence S_f do not overlap. Hence, the sum of all the costs, in the form $|C'_1| + |C'_2| - |C^*|$ or $|C'_{11}| + |C'_{12}| + |C'_{21}| + |C'_{22}| - |C^*|$, is bounded by $|B| + |S^*|$. Putting together all results, we have $|S_f| \leq 2|S^*| + (|B| + |S^*|) + |S^*| \leq 4|S^*| + 6|P| \leq 10|S^*|$.

References

[1] S.I. Ahmed, M. Hasan, Md.A. Islam, Cutting a convex polyhedron out of a sphere, Graphs and Combinatorics 27 (2011) 307–319.
[2] N. Alon, P. Seymour, R. Thomas, Planar separators, SIAM Journal on Discrete Mathematics 7 (2) (1994) 184–193.
[3] S. Bereg, O. Daescu, M. Jiang, A PTAS for cutting out polygons with lines, Algorithmica 53 (2009) 157–171.
[4] J. Bhadury, R. Chandrasekaran, Stock cutting to minimize cutting sequence, European Journal of Operational Research 88 (1) (1996) 69–87.
[5] O. Daescu, J. Luo, Cutting out polygons with lines and rays, International Journal of Computational Geometry & Applications 16 (2–3) (2006) 227–248.
[6] A. Dumitrescu, An approximation algorithm for cutting out convex polygons, Computational Geometry: Theory and Applications 29 (3) (2004) 223–231.
[7] J. Hershberger, S. Suri, Practical methods for approximating shortest paths on a convex polytope in R^3, Computational Geometry: Theory and Applications 10 (1998) 31–46.
[8] J.W. Jaromczyk, M. Kowaluk, Sets of lines and cutting out polyhedral objects, Computational Geometry: Theory and Applications 25 (2003) 67–95.
[9] R.J. Lipton, R.E. Tarjan, A separator theorem for planar graphs, SIAM Journal on Applied Mathematics 36 (1979) 177–189.
[10] G.L. Miller, Finding small simple cycle separators for 2-connected planar graphs, Journal of Computer and System Sciences 32 (1986) 265–279.
[11] M.H. Overmars, E. Welzl, The complexity of cutting paper, in: Proc. of the 1st Annual ACM Symposium on Computational Geometry, Maryland, USA, 1985, pp. 316–321.
[12] X. Tan, Approximation algorithms for cutting out polygons with lines and rays, in: Proc. of COCOON'2005, in: LNCS, vol. 3596, Springer, 2005, pp. 534–543.