
ELSEVIER

Computational Geometry 5 (1995) 95–114

Computational Geometry
Theory and Applications

Separation and approximation of polyhedral objects

Joseph S.B. Mitchell [a,*,1], Subhash Suri [b]

[a] *Department of Applied Math, SUNY, Stony Brook, NY 11794-3600, USA*
[b] *Bell Communications Research, Room 2Q-358, 445 South Street, Morristown, NJ 07960, USA*

Communicated by Leonidas Guibas; submitted 1 May 1992; accepted 8 February 1994

Abstract

Given a family of disjoint polygons $P_1, P_2, \ldots, P_k$ in the plane, and an integer parameter m, it is *NP*-complete to decide if the P_i's can be *pairwise separated* by a polygonal family with at most m edges, that is, if there exist polygons $R_1, R_2, \ldots, R_k$ with pairwise-disjoint boundaries such that $P_i \subseteq R_i$ and $\sum |R_i| \leqslant m$. In three dimensions, the problem is *NP*-complete even for two *nested convex* polyhedra. Many other extensions and generalizations of the polyhedral separation problem, either to families of polyhedra or to higher dimensions, are also intractable.

In this paper, we present efficient approximation algorithms for constructing separating families of near-optimal size. Our main results are as follows. In two dimensions, we give an $O(n \log n)$ time algorithm for constructing a separating family whose size is within a constant factor of an optimal separating family; n is the number of edges in the input family of polygons. In three dimensions, we show how to separate a convex polyhedron from a nonconvex polyhedron with a polyhedral surface whose *facet-complexity* is $O(\log n)$ times the optimal, where $n = |P| + |Q|$ is the complexity of the input polyhedra. Our algorithm runs in $O(n^4)$ time, but improves to $O(n^3)$ time if the two polyhedra are nested and convex.

Our algorithm for separating a convex polyhedron from a nonconvex polyhedron extends to higher dimensions. In d dimensions, for $d \geqslant 4$, the facet-complexity of the approximation polyhedron is $O(d \log n)$ times the optimal, and the algorithm runs in $O(n^{d+1})$ time. Finally, we also obtain results on separating sets of points, a family of convex polyhedra, and separation by non-polyhedral surfaces, such as spherical patches.

* Corresponding author.

[1] Partially supported by grants from Boeing Computer Services, Hughes Research Laboratories, Air Force Office of Scientific Research contract AFOSR-91-0328, and by NSF Grants ECSE-8857642 and CCR-9204585.

SSDI 0925-7721(95)00006-2

1. Introduction

1.1. Motivation and background

A polyhedron is one of the most widely used geometric solids in computer-modeling applications. In robotic systems, for instance, polyhedra are used to model obstacles that must be circum-navigated; in computer-aided design, they might represent automobile parts or tools; and in computer graphics or geographical databases, they might model real-world objects, such as mountains or buildings. The combinatorial complexity of the model polyhedra is a major factor in the efficiency of algorithms that manipulate these polyhedral solids. Thus, ad hoc schemes of diverse nature are employed to reduce this complexity, either by storing the "scenes" hierarchically, or by replacing the models by simpler polyhedral objects that preserve the critical geometric and topological features of the original set. The problem of separating and approximating a family of polyhedral solids arises frequently in these schemes.

A large body of literature exists dealing with variants of the problem of separating a collection of geometric objects [1,5–7,9,11,17,18,25]. Much of the research interest has focused on linear separability. The problem of determining whether two sets of points in d-space are separable by a hyperplane can be formulated as a linear programming problem and, therefore, solved in polynomial time. In fact, using the linear programming algorithms of Megiddo [17] or Dyer [9], this problem can be solved in linear time, for any fixed dimension d.

If, however, the two point sets are not linearly separable, then a natural question is to find the *minimum* number of hyperplanes that together separate the two sets. It was shown by Megiddo [18] that this problem is hard unless both the dimension and the number of separating hyperplanes are fixed. In an arbitrary dimension, the problem of deciding if two point sets can be separated by *two* hyperplanes is *NP*-complete. In a fixed dimension, separability of two point sets by k hyperplanes is intractably hard, if k is not fixed. In particular, it is *NP*-complete to decide if two *planar* sets of points can be separated by k lines [18]. Of course, separability of two point sets in a fixed dimension by a fixed number of hyperplanes can always be decided in polynomial time by a brute-force search.

Results of a positive nature exist for special classes of separators. Edelsbrunner and Preparata [11] give an $O(n \log n)$ time algorithm for finding a minimum-vertex convex polygon that separates two sets of n points in the plane. Whether similar results are possible for separation by convex polyhedra in higher dimensions is an open problem. The separation complexity also has relevance to the problem of intersection detection between pairs of preprocessed simple polygons—Mount [19] presents an algorithm whose time complexity depends on the number of vertices in a minimum-complexity separator.

Separability of polyhedral solids also has received considerable attention. Two disjoint convex polyhedra can always be separated by a single hyperplane; again, the problem can be formulated as a linear programming problem. Paralleling the separability

of point sets, the only positive results known on the polyhedral separation are for special classes of separators, and only for two dimensions. Aggarwal et al. [1] give an $O(n \log n)$ algorithm for finding a minimum-vertex polygon separating two nested convex polygons. An algorithm of similar complexity is given by Wang and Chan [25] for minimum separation of two nested nonconvex polygons.

Not surprisingly, most variants of the general polyhedral separability problem are also intractable. Let $\mathscr{P} = \{P_1, P_2, \ldots, P_k\}$ be a family of pairwise-disjoint polyhedra in d dimensions. We say a family $\mathscr{R} = \{R_1, R_2, \ldots, R_k\}$ is a *separating family* for $\mathscr{P}$ if the boundaries of $\mathscr{R}$ are pairwise-disjoint and $P_i \subseteq R_i$, for $i = 1,2,\ldots,k$. A *minimum* separating family is one that has the minimum possible number of facets of any separating family. The problem of finding a minimum separating family is *NP*-complete even for a family of convex polygons in two dimensions [5]. If a family consists of only two polyhedral solids, we call its separating family a *separator*. In three dimensions, the problem of finding a minimum-facet separator for two polyhedral solids is *NP*-complete. The problem remains *NP*-complete even if only one of the solids is nonconvex, or if the two polyhedra are convex nested polytopes [6,7].

The problem of finding a minimum separator has natural applications in surface approximation. By "fattening" a surface Σ by an amount ε, one obtains a pair of new surfaces P and Q that "sandwich" Σ between them. A minimum polyhedral separator of P and Q is a surface of least combinatorial complexity that approximates Σ within a tolerance of ε. By computing a family of approximate surfaces, corresponding to various values of ε, one can construct a hierarchical representation of Σ, allowing the user the option to use a sparse representation when the exact shape of Σ is irrelevant (e.g., when flying an airplane at 36,000 feet over a terrain), or a more detailed representation when the application calls for it (e.g., when flying at 1000 feet over the terrain). In two dimensions, if the fattened region is an annulus, the methods of Aggarwal et al. [1] or Wang and Chan [25] solve this problem in $O(n \log n)$ time. For large values of ε, the fattening may create holes, in which case, one wants a minimum-vertex *simple* polygon surrounding all the holes of the fattened region. Guibas et al. [14] give an approximation algorithm for this problem. The results presented in this paper can be used to approximate convex polyhedral surfaces in three and higher dimensions.

1.2. Summary of results

We present polynomial-time approximation schemes for several *NP*-complete polyhedral separability problems. We call a separating family $\mathscr{R}$ an *$f(n)$-approximation* if the ratio between the number of facets in $\mathscr{R}$ and the number of facets in a minimum separating family is bounded by $f(n)$. Our main results are the following. In two dimensions, we give an $O(n \log n)$ time algorithm for constructing a 7-approximation of the minimum separating family for a set of disjoint polygons, where n is the number of edges in the input family of polygons. In three dimensions, we show how to separate a convex polyhedron from a nonconvex polyhedron using a polyhedral surface whose

facet-complexity is $O(\log n)$ times the optimal. Our algorithm runs in $O(n^4)$ time, but improves to $O(n^3)$ time if the two polyhedra are nested and convex.

Our algorithm for separating a convex polyhedron from a nonconvex polyhedron extends easily to higher dimensions. In d dimensions, for $d \geq 4$, our algorithm produces an $O(d \log n)$-approximation of a minimum convex separator in $O(n^{d+1})$ time. Finally, we also obtain results on separating point sets, families of convex polyhedra and on separation by non-polyhedral surfaces, such as spherical patches.

This paper is organized as follows. In Section 2, we consider the separation problem for a family of simple convex polygons in the plane. Section 3 describes our approximation scheme for separating two three-dimensional polyhedra. In Section 4, we consider the problem of separating a family of three-dimensional convex polyhedra. Extensions and generalizations to separation of two higher-dimensional polyhedra are discussed in Section 5. In Section 6, we briefly consider the polyhedral separation problem using non-polyhedral surfaces. Finally, some concluding remarks and open problems are discussed in Section 7.

2. Polygon separation in $\mathbb{R}^2$

In this section, we propose an approximation algorithm for separating a family of k pairwise-disjoint simple polygons. This problem is *NP*-complete, even if all the polygons are convex, or rectilinearly convex [5]. An approximation algorithm for a family of convex polygons is given by Edelsbrunner, Robison and Shen [12]. They obtain a separating family of size at most $6k$-9. No such result seems to be known for nonconvex polygons. In the case of *orthogonal* nonconvex polygons, Das obtained a polynomial-time approximation method that constructs a separating family of size at most $(n + OPT)/2$, where OPT is the size of an optimal separating family for $\mathscr{P}$ [5]. The approximation method of Das [5] does not guarantee a bounded ratio between the sizes of the heuristic and the optimal family.

We present a polynomial-time algorithm for constructing a separating family for a set of k arbitrary simple polygons. The algorithm runs in $O(n \log n)$ time and produces an $O(1)$-approximation of a minimum separating family; n denotes the total number of edges in the input family. The quality of our approximation family can be stated in two ways: first, the approximation family has at most 7 times the optimal number of edges, and second, the family has no more than the optimal plus $9k$ edges. In other words, the additional number of edges used by our family depends only on k, and not n.

There are several closely-related definitions of a separating family, depending upon such considerations as whether the separating family has the same topology as $\mathscr{P}$, or whether the polygons in the separating family are edge disjoint. (For example, a separating family may use *nested* polygons even if all the polygons in $\mathscr{P}$ have disjoint interiors.) Our approximation holds for all of these definitions; however, we limit our discussion to what we consider to be the most natural definition of a separating family, and only briefly mention extensions to other definitions of separators.

The definition we adopt is that of a *separating subdivision*. A separating subdivision $\mathscr{R}$ of $\mathscr{P}$ is a polygonal subdivision of the plane such that any two polygons of $\mathscr{P}$ lie in different cells of $\mathscr{R}$. The size of $\mathscr{R}$, denoted $|\mathscr{R}|$, is the total number of edges in $\mathscr{R}$. In the following, we show how to compute a 7-approximation of an optimal separating subdivision.

Without loss of generality, we assume that all the polygons in the family $\mathscr{P}$ lie inside a large rectangular box B. We call the set of points $B \setminus \mathscr{P}$ the *free space*; this is the set of points in B that lies in the exterior of all the polygons of $\mathscr{P}$. The main idea behind our method is to partition the free space into a set of $O(k)$ "moats." Our separating subdivision is derived by computing a minimum-link path for each moat.

We start by triangulating the free space. Let $\mathscr{T}$ be a triangulation of free space, and let $\mathscr{G}_{\mathscr{T}}$ be the graph-theoretic dual of $\mathscr{T}$. Observe that $\mathscr{G}_{\mathscr{T}}$ has $O(n)$ nodes and arcs, and the maximum degree of a node is three. Further, $\mathscr{G}_{\mathscr{T}}$ has $k+1$ faces, one for each of the k polygons of $\mathscr{P}$ and one for the rectangular box B.

We contract $\mathscr{G}_{\mathscr{T}}$ by repeatedly removing degree-1 nodes. More precisely, if $\mathscr{G}_{\mathscr{T}}$ contains a node of degree 1, we delete it along with its incident edge, and repeat, until there are no nodes of degree 1. Observe that at the end of this procedure, the graph still has $k+1$ faces, but now all nodes have degrees either 2 or 3. Next, we contract all degree-2 nodes: if v is a node of degree 2, we delete v and replace its two incident edges by a single edge. (Notice that this process might introduce loops, which are arcs with both endpoints incident to the same node. In this case, we adopt the convention that a loop contributes 2 to the degree of its incident vertex.)

We denote by $\mathscr{G}$ the graph that is obtained at the end of this contraction process. Fig. 1 shows an example of three polygons, along with their triangulation and graph $\mathscr{G}$. We observe that $\mathscr{G}$ is a 3-regular planar multi-graph, with $k+1$ faces. (In order to avoid

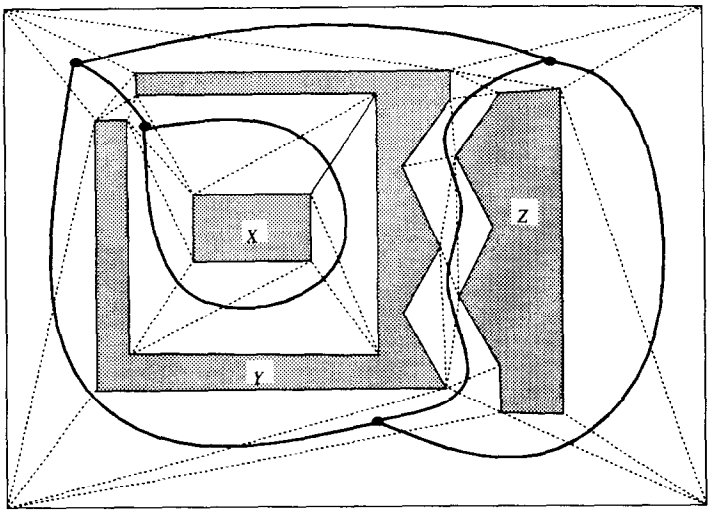

Fig. 1. Triangulation $\mathscr{T}$ and graph $\mathscr{G}$ for a family of polygons $\{X, Y, Z\}$. Triangulation edges are drawn dotted, and arcs of $\mathscr{G}$ are drawn in heavy lines. The arc surrounding X is a loop.

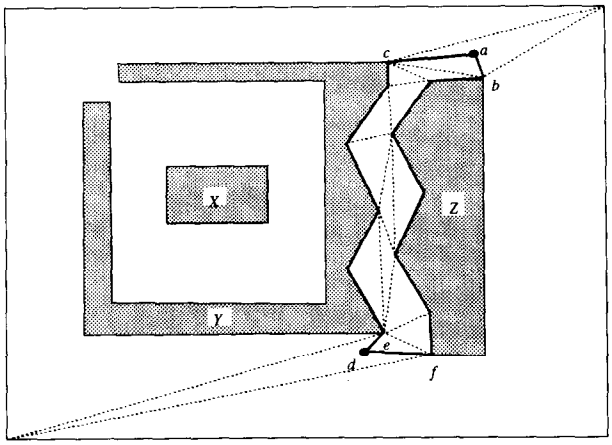

Fig. 2. A moat between the hubs a and d.

trivialities, we assume that $k \geq 2$; this guarantees that at least one node of $\mathcal{G}_{\mathcal{T}}$, and thus one node also of $\mathcal{G}$, has degree 3.) The following lemma is an easy consequence of the formula of Euler for planar graphs.

Lemma 2.1. *$\mathcal{G}$ has $k+1$ faces, $2k-2$ nodes, and $3k-3$ arcs.*

We now explain how to form "moats" using the graph $\mathcal{G}$. Consider a node v and let $t=(x, y, z)$ be the triangle in $\mathcal{T}$ that is the dual of v. We introduce a (Steiner) point a inside the triangle t, and connect it by straight line segments to the three corners of t; to be consistent, we can always choose this point to be the centroid of t (although any point interior to t suffices). The point a is called a *hub* of t and the line segments (a, x), (a, y), (a, z) are called *spokes* of t. [2] We do this for all the nodes of $\mathcal{G}$. The set of all spokes induces a partition of the free space into polygonal regions, one per arc of $\mathcal{G}$. We call these regions *moats*, and define them more formally in the following.

Consider an arc (u, v) of the graph $\mathcal{G}$. Let a and d, respectively, denote the hubs corresponding to u and v. Recall that the arc (u, v) is a contraction of a path in the original graph $\mathcal{G}_{\mathcal{T}}$, which in the primal corresponds to a sequence of triangles. (These triangles together with the triangles corresponding to u and v form a corridor.) Let (b, c) be the triangulation edges among these triangles that is closest to a, and let (e, f) be the triangulation edge closest to d. Assume that the triangles (a, b, c) and (d, e, f) are oriented clockwise. We note that the points c and e are vertices of a common polygon P_i; similarly, b and f are vertices of a polygon P_j. The *moat* corresponding to the arc (u, v) consists of the path (c, a, b), followed by the (counterclockwise) boundary chain of P_j from b to f, followed by the path (f, d, e) and then the (counterclockwise) boundary chain of P_i from e to c. Fig. 2 illustrates this construction.

[2] We associate hubs and spokes with both a triangle and its dual node.

We remark that degeneracies might arise in the above construction if the arc (u, v) is a loop. In that case, the two hub points a and d, as well as two spokes might coincide. Thus, the resulting polygon is "pinched" along the common spoke. A conceptual perturbation of d still allows us to treat the moat as a simple polygon.

In the moat M constructed above, we call a and d the *hubs* of M, and the polygonal chains between b and f, and between e and c, the *banks* of M. Either of the banks may consist of a single vertex. Let $\mathcal{M}$ denote the family of all the moats; there is one moat per arc of $\mathcal{G}$.

We find our approximate separating subdivision by computing a minimum-link path in each moat between its two hubs. If M is a moat, and a and d are its two hubs, then we compute a minimum-link path from a to d inside M; observe that in a moat corresponding to a loop, this minimum-link path resembles a nested polygon.

Let $\mathcal{S}$ denote the union of the minimum-link paths for all the moats. It is easy to see that $\mathcal{S}$ is a subdivision. The following lemma shows that in fact it is a separating subdivision of $\mathcal{P}$.

Lemma 2.2. *$\mathcal{S}$ is a separating subdivision of $\mathcal{P}$.*

Proof. The graph $\mathcal{G}_{\mathcal{T}}$ can be embedded in the plane (in the obvious way) to yield a separating subdivision, $\mathcal{S}_{\mathcal{T}}$, of $\mathcal{P}$. Since $\mathcal{G}$ is obtained from $\mathcal{G}_{\mathcal{T}}$ by removing degree-1 nodes and doing edge contractions, $\mathcal{G}$ has an embedding in the plane that is homotopically equivalent to the separating subdivision $\mathcal{S}_{\mathcal{T}}$. By its definition, $\mathcal{S}$ is homotopically equivalent to such an embedding of $\mathcal{G}$, and hence to the separating subdivision $\mathcal{S}_{\mathcal{T}}$. □

It remains to show that $|\mathcal{S}|$ is within a constant factor of the size of an optimal separating subdivision. We start by defining a canonical separating subdivision. A polygonal subdivision is called a *canonical separating subdivision* if (1) its edges are partitioned among the moats of $\mathcal{M}$, and (2) for each moat, there is a path between its two hubs, using only the edges of the subdivision, and the vertices of the subdivision include all the hubs.

Clearly, our subdivision $\mathcal{S}$ is a canonical separating subdivision. We prove that an optimal separating subdivision can be transformed into a canonical separating subdivision by the addition of $O(k)$ edges.

Consider a moat $M \in \mathcal{M}$ with hubs a and d. We say that a polygonal chain C *separates* M if a subchain of C lies within M and joins a point on the spokes incident to a to a point on the spokes incident to d. The following lemma shows that a separating subdivision must separate all the moats of $\mathcal{M}$.

Lemma 2.3. *If $\mathcal{R}$ is a separating subdivision for $\mathcal{P}$, then for every moat $M \in \mathcal{M}$, there is a chain of edges in $\mathcal{R}$ that separates M.*

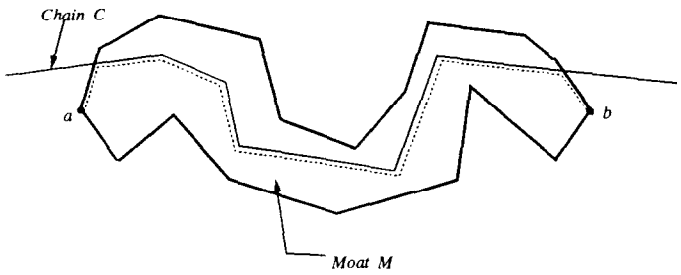

Fig. 3. Transforming a separating subdivision into a canonical separating subdivision.

Proof. Suppose that $\mathscr{R}$ does not have a chain that separates M. Then the two (different) polygons, P_i and P_j, of $\mathscr{P}$ that form the banks of M can be connected by a path without crossing an edge of $\mathscr{R}$. But this contradicts the fact that P_i and P_j must lie in different cells of $\mathscr{R}$, by the definition of separating subdivision. □

The following lemma shows how to transform a separating subdivision into a canonical separating subdivision, by adding $O(k)$ new edges.

Lemma 2.4. *Let $\mathscr{R}$ be a separating subdivision of $\mathscr{P}$. For each moat $M \in \mathscr{M}$, there exists a polygonal chain that joins the two hubs of M within the moat and all but four edges of the chain are in $\mathscr{R}$.*

Proof. By Lemma 2.3, there exists a chain C that joins a spoke incident to one hub of M to a spoke incident to its other hub. Furthermore, all edges of C, except perhaps the two extreme ones, lie completely in M. We can modify C by adding four edges, two on each end, so that the new chain joins the two hubs. Fig. 3 shows an example. In the figure, the chain C is shown by solid lines, and the modified chain is shown by dotted lines. □

It follows from Lemma 2.4 that, by adding at most four vertices per moat (two at hubs and two at spokes), we can transform $\mathscr{R}$ into a canonical separating subdivision. Among the newly added vertices, $2k-2$ are hub vertices, each of which has degree 3 in the final subdivision, and $2(3k-3)$ are spoke vertices, each of which has degree 2. The total increase in vertex-edge incidences in the subdivision is

$$3 \times (2k-2) + 2 \times (6k-6) = 18k - 18.$$

Thus, the total number of newly added edges is $(18k-18)/2 = 9k-9$. Thus, an optimal separating subdivision $\mathscr{R}$ can be converted into a canonical separating subdivision by the addition of $9(k-1)$ edges.

Finally, observe that $\mathscr{S}$, by construction, is an optimal canonical separating subdivision; we used a minimum-link path in each moat. Thus, by comparing $\mathscr{S}$ with an optimal subdivision $\mathscr{R}$, we obtain

$$|\mathscr{S}| \leqslant |\mathscr{R}| + 9(k-1) = OPT + 9(k-1).$$

In order to obtain a ratio-bound, we note that any subdivision of the plane with k faces has at least $3k/2$ edges. Thus, $OPT \geqslant 3k/2$, and our approximation bound is

$$f(n) = \frac{|\mathscr{S}|}{OPT} \leqslant 1 + \frac{9k-9}{3k/2} \leqslant 7.$$

Finally, let us analyze the time complexity of our algorithm. A triangulation $\mathscr{T}$ can be computed in time $O(n \log n)$ [21], or in time $O(n + k \log^{1+\epsilon} k)$, for an arbitrary $\epsilon > 0$, using a recent result of Bar-Yehuda and Chazelle [2]. We can compute and contract the dual graph in linear additional time. Finally, finding minimum-link paths in moats also takes linear time; this can be done using an algorithm of Suri [24]. (Notice that the moats are disjoint, and hence their combined complexity is linear). We summarize our result in the following theorem.

Theorem 2.5. *Given a family $\mathscr{P}$ of k pairwise-disjoint simple polygons, with a total of n edges, one can compute in time $O(n \log n)$ a separating subdivision of $\mathscr{P}$ with at most $9(k-1)$ plus the optimal number of edges. The ratio between the sizes of the computed separating subdivision and an optimal subdivision is bounded by 7.*

Remark. If instead of a separating subdivision, we consider a separating family of polygons, then a straightforward modification of our analysis shows that a canonical family obtained from minimum-link paths through moats gets within an additive term $18(k-1)$, or a multiplicative factor 7, of optimal. These bounds apply regardless of whether or not we allow nested polygons in the separating family. If a separating family has nested polygons, then its homotopy class will be different from that of $\mathscr{P}$, but it is a separating family nonetheless.

3. Polyhedral separation in $\mathbb{R}^3$

3.1 Preliminaries

A polyhedron P in three dimensions is a set of points whose boundary consists of a union of a finite number of (closed) planar pieces, called facets. The facets of P meet along straight line segments, called edges, and its edges meet at points, called vertices. We assume that our polyhedra have genus zero, meaning that they are simply connected. The number of vertices $V(P)$, edges $E(P)$ and facets $F(P)$ of P are related by the well-known formula of Euler:

$$V(P) - E(P) + F(P) = 2$$

We will be concerned mainly with the number of facets of a polyhedron. We use the shorter notation $|P|$ to denote the number of facets, or the *facet complexity*, of the polyhedron P. To simplify the presentation, we restrict our discussion to bounded polyhedra, although all our results hold for unbounded polyhedra as well.

A convex polyhedron P can be represented either as the convex hull of its vertex set, or the common intersection of halfspaces determined by its facets. In this paper, we will primarily use the halfspace-intersection form. Given a convex polyhedron P, we let $\mathscr{H}(P)$ denote the family of hyperplanes that bound the facets of P; thus, $|P| = |\mathscr{H}(P)|$. (Each member of $\mathscr{H}(P)$ contributes a facet to P.) A hyperplane $h \in \mathscr{H}(P)$ bounds two halfspaces, denoted h^+ and h^-, and we adopt the convention that the halfspace containing P is positive:

$$P = \bigcap_{h \in \mathscr{H}(P)} h^+.$$

The family $\mathscr{H}(P)$ will be referred to as the *intersection form* representation of P.

In the following two sections, we develop our approximation algorithm for separating a convex polyhedron from a nonconvex polyhedron. For technical reasons, we assume that the two given polyhedra, P and Q, are nested, namely, $Q \subseteq P$. This results in no loss of generality, since we can easily modify one of the polyhedra to surround the other. (Without loss of generality, assume that P is the nonconvex polyhedron. We put a suitably large box B around $P \cup Q$, and consider a vertex $p \in P$ that is also a vertex of $CH(P \cup Q)$, the convex hull of the two polyhedra. The vertex p sees at least one corner, say b, of the box B. We add a thin "tube" having a constant number of faces between p and b, thus connecting P to B. Now, $B \cap P^c$ plus the tube connecting p and b is a polyhedron P' that contains Q, where P^c is the complement of P. It is easy to see that any polyhedron separating P and Q can be modified, by the addition of at most one face, into a polyhedron nested between P' and Q.) Thus, a separator of P and Q is a polyhedron K such that $Q \subseteq K \subseteq P$. We say that K is a *minimum* separator if K has the minimum possible facet-complexity. Clearly, if M is a minimum separator, then $|M| \leqslant \min\{|P|, |Q|\}$. We say that a separator K is an *$f(n)$-approximation* of a minimum separator if $|K|/|M| \leqslant f(n)$. Our algorithm finds an $O(\log n)$-approximation of the minimum separator, where $n = |P| + |Q|$ is the input complexity.

A surface separates P from Q if and only if every path from a point on ∂P to a point on ∂Q meets the surface. This notion of separation, though clearly sufficient, is algorithmically not very attractive. In order to formulate the separation problem as a discrete problem, we partition the boundary ∂P into a polygonal cell complex, and then use the hyperplanes from the set $\mathscr{H}(Q)$ to separate portions of this complex from Q. A set of hyperplanes that together separates the entire cell complex will be used to form a separator polyhedron. We use a greedy approach to select the hyperplanes, and then show that the facet-complexity of the resulting separator is not too far from the optimal. The performance analysis of our algorithm mimics the analysis of a well-known greedy set covering heuristic.

We first consider the special case of two convex nested polyhedra.

3.2. Separating nested convex polyhedra

The problem of minimally separating two nested convex polyhedra is one of the simplest forms of polyhedral separation problem in 3-space. In this problem, we are

given two nested convex polyhedra P and Q, that is, $Q \subseteq P$, and we want to find a third polyhedron K with a minimum number of facets such that $Q \subseteq K \subseteq P$. This problem in a dual form arises in the study of *sequential stochastic automata* [23], where a minimum vertex nested polyhedron is desired. In this application, the number of vertices in a polyhedron corresponds to the number of states in a given machine, and the geometric containment of one polyhedron by another corresponds to the covering of the state space of one machine by another. As mentioned in the introduction, this problem is *NP*-complete [7], but our approximation algorithm produces a solution having $O(\log n)$ times the optimal number of vertices. (The problems of minimizing the number of vertices and the number of facets are duals of each other, under a standard geometric transformation.)

3.2.1. Canonical separators and covering of P

We start by defining the canonical form of a separator polyhedron. Let $\mathscr{H}(Q) = \{q_1, q_2, \ldots, q_m\}$, where $m \leqslant n$, be the intersection form of the inner polyhedron Q. We say that a separator K is *canonical* if $\mathscr{H}(K) \subseteq \mathscr{H}(Q)$; in other words, the hyperplanes bounding the facets of K also bound facets of Q. Our first observation is that we can find an approximately minimum separator by searching only the space of all canonical nested polyhedra. This reduces the search space from infinite to finite.

Lemma 3.1 (Canonical Form). *There exists a canonical separator of P and Q whose facet-complexity is at most three times the facet-complexity of a minimum separator.*

Proof. Let M be a minimum separator of P and Q, and let $\mathscr{H}(M) = \{m_1, m_2, \ldots, m_p\}$ be its intersection form. We can translate each m_i towards Q until becomes tangent to Q and, thus, assume without loss of generality that each facet of M is incident with a vertex of Q. Given a hyperplane $m \in \mathscr{H}(M)$, let $v \in Q$ be a vertex incident to it, and let $q_1(v), q_2(v), \ldots, q_s(v)$ be the hyperplanes of $\mathscr{H}(Q)$ passing through v. Observe that $m^+ \supseteq \bigcap_{i=1}^{s} q_i^+(v)$, which implies that $m^- \subseteq \bigcup_{i=1}^{s} q_i^-(v)$. By Caratheodory's Theorem (see [3,22]), there exist three hyperplanes $q_j(v)$, $q_k(v)$, $q_l(v)$ such that $m^- \subseteq q_j^-(v) \cup q_k^-(v) \cup q_l^-(v)$. We replace m by the triple of hyperplanes $q_j(v)$, $q_k(v)$, and $q_l(v)$. Let

$$\mathscr{H}(M') = \big(\mathscr{H}(M) - \{m\}\big) \cup \big(q_j(v) \cup q_k(v) \cup q_l(v)\big).$$

Then, it is easy to see that $Q \subseteq M' \subseteq M \subseteq P$. By applying this procedure to all the hyperplanes of $\mathscr{H}(M)$, we obtain a new family of hyperplanes, say, $\mathscr{H}(K)$, such that (i) $\mathscr{H}(K) \subseteq \mathscr{H}(Q)$, (ii) $Q \subseteq K \subseteq P$, and (iii) $|K| \leqslant 3|M|$. Now, K is a canonical separator and the proof is complete. □

We now discuss the discretization of ∂P, which is a partition of the boundary into a family of polygonal cells. Let $\mathscr{C}(\mathscr{H})$ denote the *cell complex* formed on the boundary ∂P by the family of hyperplanes $\mathscr{H} = \mathscr{H}(Q)$. This cell complex is a subdivision of the surface ∂P into convex polygonal cells, with disjoint relative interiors. The edges of $\mathscr{C}(\mathscr{H})$ are either the portions of edges of P or they are contributed by the intersection between a hyperplane of $\mathscr{H}$ and a facet of P. The vertices of $\mathscr{C}(\mathscr{H})$ are formed by an

intersection between three hyperplanes in the family $\mathscr{H}(P) \cup \mathscr{H}(Q)$, such that at least one of the hyperplanes is from $\mathscr{H}(P)$. Thus, a vertex can either be a vertex of P (all three planes from $\mathscr{H}(P)$), or it can arise from an intersection between a plane of $\mathscr{H}(Q)$ and an edge of P, or it can arise from an intersection between a plane of $\mathscr{H}(P)$ and two planes of $\mathscr{H}(Q)$. We use the notation $\mathscr{C}(\mathscr{H})$ to also denote the set of all cells in the complex.

Lemma 3.2. *The cell complex $\mathscr{C}(\mathscr{H})$ has $O(n^2)$ vertices, edges and faces.*

Proof. A plane $q_i \in \mathscr{H}$ intersects the boundary ∂P in a convex polygonal curve, $\gamma(q_i)$, which has at most n vertices and edges. Two such curves, $\gamma(q_i)$ and $\gamma(q_j)$, $i \neq j$, can intersect (cross) in at most two points. This follows since the points of crossing must lie both on ∂P and on the line $q_i \cap q_j$; but a line can cross the boundary of a convex polyhedron in at most two points. Finally, the number of edges and vertices contributed by P itself is at most n. Thus, the total number of vertices, edges and faces in the cell complex $\mathscr{C}(\mathscr{H})$ is $O(n^2)$. □

Consider a hyperplane $q_i \in \mathscr{H}(Q)$. It divides the cells of $\mathscr{C}(\mathscr{H})$ into two sets: those contained in the positive halfspace q_i^+ and those contained in the negative halfspace q_i^-. (Observe that since $q_i \in \mathscr{H}(Q)$, it does not properly intersect the interior of any cell in $\mathscr{C}(\mathscr{H})$. Let $S(q_i)$ denote the set of cells contained in the *negative* halfspace bounded by q_i:

$$S(q_i) = \{c \in \mathscr{C}(\mathscr{H}) \mid c \subset q_i^-\}.$$

We say that q_i *covers* the cells in $S(q_i)$, in the sense that q_i separates any cell in $S(q_i)$ from Q, and we call $S(q_i)$ the *cover set* of q_i. We show that a subfamily of hyperplanes that covers all the cells of $\mathscr{C}(\mathscr{H})$ forms a separator of P and Q, and that the following algorithm based on a greedy strategy produces a good approximation of a minimum separator polyhedron.

3.2.2. The algorithm and its analysis

The following procedure computes a separator polyhedron for P and Q.

Separate (P, Q)
1. $\mathscr{C} \leftarrow \mathscr{C}(\mathscr{H})$
2. $\mathscr{H}(K) \leftarrow \emptyset$
3. **while** $\mathscr{C} \neq \emptyset$ **do**
4. Select a plane $q_i \in \mathscr{H}(Q)$ that maximizes $|\mathscr{C} \cap S(q_i)|$
5. $\mathscr{H}(K) \leftarrow \mathscr{H}(K) \cup \{q_i\}$
6. $\mathscr{C} \leftarrow \mathscr{C} - S(q_i)$
7. **end while**
8. **return** $\mathscr{H}(K)$

First, we show that the polyhedron K returned by the algorithm is, indeed, a separator of P and Q; then, we provide an analysis of the running time of the algorithm; and, finally, we analyze the performance of our algorithm and show that K is a $O(\log n)$-approximation of a minimum separator.

Lemma 3.3. *The polyhedron K returned by the algorithm* Separate *satisfies* $Q \subseteq K \subseteq P$.

Proof. A cell $c \in \mathscr{C}(\mathscr{H})$ is removed from the list $\mathscr{C}$ only after we add a plane q_i to $\mathscr{H}(K)$ such that $c \in q_i^-$. Since the cells of $\mathscr{C}(\mathscr{H})$ cover the boundary of P and since $K = \bigcap_{q \in \mathscr{H}(K)} q^+$, it follows that $K \subseteq P$. Further, since $\mathscr{H}(K) \subseteq \mathscr{H}(Q)$, clearly $Q \subseteq K$. This completes the proof. □

A straightforward implementation of the algorithm Separate would require $O(n^4)$ time: n iterations of the **while** loop, each requiring $O(n^3)$ for updating the cover sets $S(q)$ of all the hyperplanes $q \in \mathscr{H}(Q)$. By exploiting the fact that the total number of "incidences" between the cells of $\mathscr{C}(\mathscr{H})$ and the cover sets is $O(n^3)$, we establish the following lemma.

Lemma 3.4. *The algorithm* Separate *can be implemented to run in time* $O(n^3)$.

Proof. We build the arrangement $\mathscr{C}(\mathscr{H})$, and compute the cover sets $S(q)$, for all $q \in \mathscr{H}(Q)$. For each cell $c \in \mathscr{C}(\mathscr{H})$, we also maintain the list $L(c)$ of all the planes $q \in \mathscr{H}$ such that $c \in S(q)$. These steps take $O(n^3)$ time. Observe that

$$\sum_{c \in \mathscr{C}(\mathscr{H})} |L(c)| = O(n^3). \tag{2}$$

The bound in Eq. (2) follows from the facts that the total number of cells is $O(n^2)$ and that the maximum size of each list is n.

During the algorithm we maintain the sets $\mathscr{C} \cap S(q_i)$, for all q_i. Initially, these sets are given by our preprocessing step. After a plane q_i is selected by the algorithm and added to the set $\mathscr{H}(K)$, we remove each of the cells $c \in \mathscr{C} \cap S(q_i)$ from the cover sets of other planes. In particular, given a cell $c \in \mathscr{C} \cap S(q_i)$, we scan the list $L(c)$ and for each $q \in L(c)$, set $S(q) = S(q) - \{c\}$. This takes time proportional to $\sum_{c \in \mathscr{C} \cap S(q_i)} |L(c)|$. A cell c is processed only once during the algorithm and, thus, the bound on the running time follows from Eq. (2). □

To analyze the performance of our algorithm, we cast the problem as a set cover problem. We have an underlying universal set $\mathscr{C}(\mathscr{H})$, and a family of subsets $\mathscr{S} = \{S(h) \mid h \in \mathscr{H}\}$. We want to choose a minimum number of sets from $\mathscr{S}$ whose elements together cover $\mathscr{C}(\mathscr{H})$. We start by establishing the connection between a set cover and a separator. The following lemma shows that the sets associated with the hyperplanes in the intersection-form representation of a separator cover the cell complex $\mathscr{C}(\mathscr{H})$.

Lemma 3.5. *Let K be a canonical separator of P and Q, and let $\mathscr{H}(K)$ be its intersection form. Then, $\mathscr{C}(\mathscr{H}) = \bigcup_{h \in \mathscr{H}(K)} S(h)$.*

Proof. Assume, for the sake of contradiction, that there exists a cell $c \in \mathscr{C}(\mathscr{H})$ that is not covered by the hyperplanes in $\mathscr{H}(K)$. In other words, $c \in h^+$, for all planes $h \in \mathscr{H}(K)$. But, since $Q \subseteq \bigcap_{h \in \mathscr{H}(K)} h^+$, we can join some point of c with any point of Q by a path without intersecting any of the planes in $\mathscr{H}(K)$, which contradicts the hypothesis that K separates P from Q. Thus, the sets $S(h)$, for $h \in \mathscr{H}(K)$, must together cover all the cells of $\mathscr{C}(\mathscr{H})$. □

Thus, our problem is to show that the set cover found by the algorithm Separate is close to an optimal set cover. Our analysis mimics the analysis of a well-known greedy set cover heuristic, see Johnson [15] and Lovász [16]. We include a proof for the sake of completeness—our presentation is borrowed from [4].

Let K be the separator returned by our greedy algorithm, and let M be an optimal separator of P and Q. We will show that the set cover corresponding to K is within factor $O(\log n)$ of an optimal set cover, say, the one corresponding to M. Let $\mathscr{H}(K) = \{h_1, h_2, \ldots, h_k\}$ be the intersection-form of K, and assume without loss of generality, that the algorithm picks the planes in the order $h_1, h_2, \ldots, h_k$. Let S_i denote the cover set $S(h_i)$ corresponding to the ith hyperplane, and let T_i denote the set of elements covered by the first i cover sets: $T_i = \bigcup_{j=1}^{i} S_j$.

When the ith hyperplane is picked, our greedy algorithm incurs a cost of one, which it distributes evenly among the elements of S_i that are covered for the first time. That is, if $c \in S_i$ is covered for the first time, then

$$cost(c) = \frac{1}{|S_i - T_{i-1}|}.$$

Thus, the size of the set cover found by the greedy algorithm is

$$\begin{aligned} |\mathscr{H}(K)| &= \sum_{c \in \mathscr{C}(\mathscr{H})} cost(c) \\ &\leqslant \sum_{h^* \in \mathscr{H}(M)} \sum_{c \in S(h^*)} cost(c). \end{aligned} \tag{3}$$

Next, we show that for any set $S(h)$, $h \in \mathscr{H}(Q)$,

$$\sum_{c \in S(h)} cost(c) \leqslant O(\log n). \tag{4}$$

To prove the above inequality, let $S = S(h)$ be an arbitrary set, and let $u_i = |S - T_i|$ be the number of elements in S that remain uncovered after the greedy algorithm has picked its first i sets; we set $u_0 = |S|$. Observe that $u_{i-1} \geqslant u_i$, and $u_{i-1} - u_i$ elements of S are covered for the first time by S_i. Thus,

$$\sum_{c \in S} cost(c) = \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{|S_i - T_{i-1}|}.$$

However,

$$|S_i - T_{i-1}| \geq |S - T_{i-1}| = u_{i-1},$$

since the greedy algorithm always picks a hyperplane that covers the most among the remaining elements. It follows that

$$\sum_{c \in S} cost(c) \leq \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{u_{i-1}},$$

from which it follows easily that

$$\sum_{c \in S} cost(c) \leq \sum_{i=1}^{|S|} 1/i = O(\log n).$$

Notice that a set $S(h)$ can have size $O(n^2)$.

Now, Equations (3) and (4) together imply that

$$|\mathscr{K}(K)| \leq O(\log n)\,|\mathscr{K}(M)|,$$

which shows that the ratio between the facet-complexities of the greedy separator and an optimal separator is $O(\log n)$. The following theorem summarizes this result.

Theorem 3.6. *Let P and Q be two convex nested polyhedra in three dimensions, with a total of n facets (resp., vertices). One can find an $O(\log n)$-approximation of a minimum-facet (resp., minimum-vertex) polyhedral separator of P and Q in $O(n^3)$ time.*

3.3. Separating a convex polyhedron from a nonconvex polyhedron

We now consider two nested polyhedra P and Q, $Q \subseteq P$, one of which is convex and the other is a general nonconvex polyhedron. Clearly, if P is convex, then we can simply replace Q with the convex hull of Q and apply the algorithm of Section 3.2. If, on the other hand, P is nonconvex (Q is convex), then we cannot simply replace P with its convex hull. Instead, we note that the algorithm of the previous section still works for this case; however, the running time deteriorates by a factor of n. The key difference lies in the size of the cell complex $\mathscr{C}(\mathscr{H})$. We now elaborate on this difference.

It is easily observed that an optimal separator of P and Q is a convex polyhedron. Thus, Lemma 3.1 applies, stating that there exists a canonical separator whose facet complexity is at most three times the optimal. In order to invoke the algorithm Separate, we again partition the surface ∂P into a cell complex $\mathscr{C}(\mathscr{H})$, which is formed by intersection of ∂P with the family of hyperplanes $\mathscr{H} = \mathscr{H}(Q)$. This cell complex is similar to the one in the previous section, and consists of polygonal cells with disjoint interiors, however, the total number of vertices, edges, and faces is now $O(n^3)$ — this follows from the fact that a cell complex produced by n planes in three dimensions has $O(n^3)$ cells, facets, edges and vertices. Except for this difference in the size of $\mathscr{C}(\mathscr{H})$, the details and the analysis of the approximation algorithm are identical to the convex case. The following theorem states our result.

Theorem 3.7. *Let P and Q be two polyhedra in three dimensions, one of which is convex, with a total of n facets. One can find an $O(\log n)$-approximation of a minimum-facet polyhedral separator of P and Q in $O(n^4)$ time.*

3.4. Separating two point sets

Our method also works if P and Q are two disjoint sets of points in three dimensions. In this case, we can find an $O(\log n)$-approximation of their minimum-facet convex separator. Without loss of generality, assume that the convex hull of Q does not contain any point of P — if neither convex hull is empty of points from the other set, then clearly P and Q cannot be separated by a convex polyhedron. In this case, we let $\mathscr{H}(Q)$ denote the intersection form of the convex hull of Q. The cover set $S(q)$, for $q \in \mathscr{H}(Q)$, is defined as the set of points of P that lies in the negative halfspace bounded by q:

$$S(q) = \{ p \in P \mid p \subset q^- \}.$$

The underlying set to be covered is P. It is easy to see that the algorithm Separate finds an $O(\log n)$-approximate convex separator of P and Q. The running time improves to $O(n^2)$ since there are n cover sets, each of size at most n.

Theorem 3.8. *Let P and Q be two sets of n points in three dimensions. If the two point sets are separable by a convex polyhedron, then one can find an $O(\log n)$-approximation of their minimum-facet convex separator in $O(n^2)$ time.*

4. Separating k disjoint convex polyhedra

Let $\mathscr{P} = \{P_1, P_2, \ldots, P_k\}$ be a family of k disjoint convex polyhedra in three dimensions, with a total of n facets. We want to find a *minimum separating family* for $\mathscr{P}$, that is, a family of disjoint convex polyhedra $\{R_1, R_2, \ldots, R_k\}$, with $P_i \subseteq R_i$, such that $\sum_{i=1}^{k} |R_i|$ is minimized. This problem is *NP*-complete, even in two dimensions [5]. The following result, however, guarantees a separating family whose size depends only on k, and not on n.

Theorem 4.1. *Given a family of k convex polyhedra in three dimensions, a separating family of size $O(k^2)$ always exists, and this bound cannot be improved in the worst case. After an $O(n)$ time preprocessing, one can find a separating family with $O(k^2)$ facets in $O(k^2 \log^2 n)$ time, where n is the total complexity of the input polyhedra.*

Proof. Let $p_i \in P_i$ and $p_j \in P_j$ denote two points that form a closest pair for P_i and P_j:

$$d(p_i, p_j) = \min\{d(x, y) \mid x \in P_i, y \in P_j\},$$

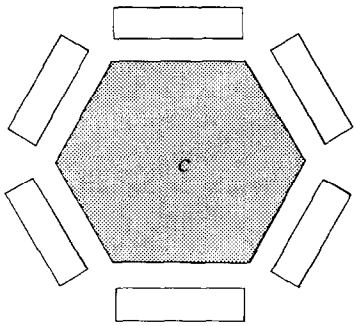

Fig. 4. The top view of our lower bound construction for separating k convex polyhedra in three dimensions.

where $d(\cdot,\cdot)$ denotes the Euclidean distance. Let B_{ij} denote the hyperplane that is normal to the line determined by p_i and p_j, and that passes through the midpoint of the segment (p_i, p_j). Let

$$R_i = \bigcap_{j \neq i} B_{ij}^+,$$

where B_{ij}^+ is the halfspace containing P_i. Observe that R_i is a convex polyhedron with at most k facets and that $P_i \subseteq R_i$. Thus, $\mathscr{R} = \{R_1, R_2, \ldots, R_k\}$ is a *separating family* for $\mathscr{P}$ and the total number of facets in $\mathscr{R}$ is $O(k^2)$.

To establish the lower bound on the size of a separating family, consider the family of convex polygons shown in Fig. 4. Let C be a unit-height cylinder whose base is shaped like the regular k-gon in the center of Fig. 4. The side length of the k-gon is assumed to be large, so that the cylinder C looks like a "pancake." We place k copies of the cylinder C at $z = 2i$, for $z = 0,1,\ldots,k-1$. We lift the outer polygons into cylinders of height $2k$, extending from $z = 0$ to $z = 2k$. Any polyhedron that separates the ith cylinder C_i from the remaining polyhedra has at least $k+2$ facets. Thus, the total number of facets in the whole separating family is at least $\Omega(k^2)$.

Finally, in order to construct the polyhedron R_i, for $i = 1,2,\ldots,k$, we only need to find the minimum distance between two convex polyhedra. This can be done in $O(\log^2 n)$ time, assuming that the input polyhedra are preprocessed in a logarithmic hierarchy [8]. Such an hierarchy can be built in linear time from any other standard description of the polyhedra, such as doubly-connected-edge-lists [20]. Thus, we can compute a separating family consisting of $O(k^2)$ facets in $O(k^2 \log^2 n)$ time. □

5. Higher dimensions

The separation algorithm of Section 3 was presented in enough abstraction to admit immediate extension to higher dimensions. Suppose that P and Q are two d-dimensional polyhedra, with a total of n facets. We assume that Q is convex and that $Q \subseteq P$. An easy extension of Lemma 3.1 shows that there always exists a canonical separator of

P and Q whose facet-complexity is at most d times the optimal. The combinatorial complexity of the cell complex $\mathscr{C}(\mathscr{H})$ on the boundary of P is $O(n^d)$; the arrangement formed by n planes in d-space has $O(n^d)$ faces of all dimensions. The cover set $S(q)$, for $q \in \mathscr{H}(Q)$, is the set of d-dimensional cells in $\mathscr{C}(\mathscr{H})$ that lie in the closed halfspace q^-. Our algorithm finds a set of hyperplanes $\mathscr{H}(K)$ that together separate every point of ∂P from Q. The same analysis as before shows that the size of $\mathscr{H}(K)$ is within $O(\log n)$ of the best canonical separator. The total running time of the algorithm is dominated by the initial construction of the arrangement $\mathscr{C}(\mathscr{H})$ and the updating of cover sets.

Theorem 5.1. *Let P and Q be two polyhedra in d dimensions, where $d \geqslant 3$ is considered fixed. Assume that Q is convex and that P and Q have a total of n facets. In $O(n^{d+1})$ time, one can find a polyhedral separator of P and Q whose facet-complexity is $O(d \log n)$ times the optimal.*

6. Curved surfaces

In many real applications, the surfaces of choice are non-polyhedral. Solid modelers, for instance, often use boundary representations that include spherical patches in addition to planes. Thus, it is interesting to investigate the separability problem using non-polyhedral surfaces.

Consider, for instance, the problem of separating two polyhedral solids using a spherical surface — we call a surface *spherical* if the enclosed solid is the intersection of a finite number of spheres. Given a spherical surface K, its intersection form $\mathscr{H}(K)$ is the set of spheres that define the "facets" of K. A separator K is called a *canonical* separator if each sphere in $\mathscr{H}(K)$ is determined by four vertices of the inner polyhedron Q.

A simple extension of Lemma 3.1 shows that an optimal spherical separator can always be transformed into a canonical separator, at the expense of increasing the facet-complexity three-fold. There are $O(n^2)$ circumspheres of Q determined by quadruples of vertices of Q — each such sphere corresponds to a Voronoi vertex in the *farthest point* Voronoi diagram of the vertices of Q and this Voronoi diagram has $O(n^2)$ vertices [10].

We form the cell complex $\mathscr{C}$ by intersecting the $O(n^2)$ canonical spheres with the boundary of P. This gives a subdivision of size $O(n^5)$ — each pair of spheres intersects ∂P in $O(n)$ points. By running our greedy heuristic on this cell complex, we can find a canonical spherical separator whose facet-complexity is $O(\log n)$ times the optimal.

Theorem 6.1. *Let P and Q be two polyhedra in three dimensions, one of which is convex, with a total of n facets. In $O(n^7)$ time, one can find a spherical separator of P and Q whose facet-complexity is $O(\log n)$ times the optimal.*

Extensions to higher dimensions and other elementary surfaces can be obtained along similar lines.

7. Conclusion

We have presented polynomial-time approximation algorithms for several polyhedral separation problems. Our main results are an $O(1)$-approximation algorithm for a family of polygons in the plane, and an $O(\log n)$-approximation algorithm for separating a convex polyhedron from a nonconvex polyhedron. The general topic of polyhedral separation and approximation is quite fundamental, and poses several challenging problems of both theoretical and practical nature. There are numerous possibilities for further research on these problems. We outline a few in the following.

(1) Our algorithms for polyhedral separation are based on the greedy set-cover heuristic. It is an interesting problem to determine if the special structure of the geometric problem can be used to improve the log n approximation ratio.

(2) Is it possible to improve the approximation ratio if the two polyhedral surfaces P and Q are scaled copies of each other, or if they arise from the fattening of a single surface?

(3) The problem of finding approximation algorithms for separation of a family of non-convex polyhedra in three (or more) dimensions remains open. While our methods here cannot be applied directly to this problem, we suspect that some transformation to a set-cover problem is possible. We are currently examining this problem.

(4) Can our methods be generalized to handle separation with more general curved surfaces?

(5) We have studied the problem of minimizing facet complexity of a separating surface. It would be interesting to study the problem of finding a minimum surface-area separator. In two dimensions, the problem of finding a minimum-length separating cycle is known as the relative convex hull problem, and is well-studied. In three dimensions, the problem of minimum-area surfaces has a long history in the physics and mathematical literature on soap films. What can be said about approximation algorithms, or about the problem of minimizing the area of a polyhedral separating surface?

References

[1] A. Aggarwal, H. Booth, J. O'Rourke, S. Suri and C.K. Yap, Finding minimal convex nested polygons, Inform. and Control 83 (1989) 98–110.

[2] R. Bar-Yehuda and B. Chazelle, Triangulating disjoint Jordan chains, Internat. J. Comput. Geom. Appl. 4 (1994) 475–481.

[3] C. Carathéodory, Über den Variabilitätsbereich der Fourierschen Konstanten von positiven harmonischen Funktionen, Rendiconto del Circolo Matematico di Palermo 32, (1911) 193–217.

[4] T. Corman, C. Leiserson and R. Rivest, Introduction to Algorithms (The MIT Press, Cambridge, MA, 1990).

[5] G. Das, Approximation schemes in computational geometry, Ph.D. Thesis, University of Wisconsin, 1990.
[6] G. Das and D. Joseph, Minimum vertex hulls for polyhedral domains, Theoret. Comput. Sci. 103 (1992) 107–135.
[7] G. Das and D. Joseph, The complexity of minimum convex nested polyhedra, Canadian Conference on Computational Geometry (1990) 296–301.
[8] D. Dobkin and D. Kirkpatrick, Determining the separation of preprocessed polyhedra — a unified approach, Proc. of ICALP '90 LNCS, Vol. 443 (Springer, Berlin, 1990) 400–413.
[9] M.E. Dyer, On a multidimensional search technique and its application to the Euclidean one-center problem, SIAM J. of Computing 15 (1986) 725–738.
[10] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Vol. 10 of EATCS Monographs on Theoretical Computer Science (Springer, Berlin, 1987).
[11] H. Edelsbrunner and F.P. Preparata, Minimum polygon separation, Inform. and Comput. 77 (1987) 218–232.
[12] H. Edelsbrunner, A.D. Robinson and X. Shen, Covering convex sets with non-overlapping polygons, Discrete Math. 81 (1990) 153–164.
[13] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, New York, 1979).
[14] L.J. Guibas, J.E. Hershberger, J.S.B. Mitchell and J.S. Snoeyink, Approximating polygons and subdivisions with minimum link paths, Internat. J. Comput. Geom. Appl. 3 (1993) 383–415.
[15] D.S. Johnson, Approximation algorithms for combinatorial problems, J. Comput. System Sci. 9 (1974) 256–278.
[16] L. Lovász, On the ratio of optimal integral and fractional covers, Discrete Math. 13 (1975) 383–390.
[17] N. Megiddo, Linear programming in linear time when the dimension is fixed, JACM 12 (1984) 114–127.
[18] N. Megiddo, On the complexity of polyhedral separability, Discrete Comput. Geom. 3 (1988) 325–337.
[19] D.M. Mount, Intersection detection and separators for simple polygons, Proc. 8th Annu. ACM Sympos. Comput. Geom. (1992) 303–311.
[20] D.E. Muller and F.P. Preparata, Finding the intersection of two convex polyhedra, Theoret. Comput. Sci. 7 (1978) 217–236.
[21] F.P. Preparata and M.I. Shamos, Computational Geometry (Springer, New York, 1985).
[22] A. Schrijver, Theory of Linear and Integer Programming (Wiley, New York, 1986).
[23] C.B. Silio, An efficient simplex coverability algorithm in E^2 with application to stochastic sequential machines, IEEE Trans. Comput. 28 (1979) 109–120.
[24] S. Suri, On some link distance problems in a simple polygon, IEEE Trans. Robotics Automation 6 (1990) 108–113.
[25] C.A. Wang and E. Chan, Finding the minimum visible vertex distance between two non-intersecting simple polygons, Proc. of 2nd Symposium on Computational Geometry (1986) 34–42.