Piecewise Linear Paths Among Convex Obstacles*

M. de Berg,¹ J. Matoušek,² and O. Schwarzkopf¹

¹Vakgroep Informatica, Universiteit Utrecht,
Postbus 80.089, 3508 TB Utrecht, The Netherlands
markdb@cs.ruu.nl
otfried@cs.ruu.nl

²Katedra aplikované matematiky, Universita Karlova,
Malostranské nám. 25, 180 00 Praha 1, Czech Republic
matousek@kam.mff.cuni.cz

and
Institut für Informatik, Freie Universität Berlin,
Arnimallee 2-6, D-14195 Berlin 33, Germany

Abstract. Let $\mathcal{B}$ be a set of n arbitrary (possibly intersecting) convex obstacles in $\mathbb{R}^d$. It is shown that any two points which can be connected by a path avoiding the obstacles can also be connected by a path consisting of $O(n^{(d-1)\lfloor d/2+1 \rfloor})$ segments. The bound cannot be improved below $\Omega(n^d)$; thus, in $\mathbb{R}^3$, the answer is between n^3 and n^4 . For open disjoint convex obstacles, a $\Theta(n)$ bound is proved. By a well-known reduction, the general case result also upper bounds the complexity for a translational motion of an arbitrary convex robot among convex obstacles. Asymptotically tight bounds and efficient algorithms are given in the planar case.

1. Introduction

The results presented in this paper are motivated by the following problem. Consider n disjoint convex obstacles of an arbitrary shape in the plane or in $\mathbb{R}^3$, and a convex robot R . Suppose that a position q of R can be reached from a position p by a translational motion of R avoiding the obstacles. If such a motion is piecewise linear, we can measure its complexity by the number of segments (linear pieces) it consists of. Is it possible to bound the minimum necessary complexity of such a

* This research was supported by The Netherlands' Organization for Scientific Research (NWO) and partially by the ESPRIT III Basic Research Action 6546 (PROMotion). J. M. acknowledges support by a Humboldt Research Fellowship. Part of this research was done while he visited Utrecht University.

motion solely in terms of n , the number of obstacles? This does not seem obvious even if R is a single point. Note that the boundaries of two convex shapes in the plane may intersect an arbitrary number of times.

It turns out that the minimum necessary complexity of a motion can indeed be bounded in this way, in an arbitrary fixed dimension d .

By a well-known method we can reduce the problem of translational motion of a convex robot among convex obstacles to the case of a point robot among convex obstacles (every obstacle B is replaced by $B + (-R)$, the Minkowski sum of the obstacle and the image of the robot under central reflection). Then, however, the obstacles are not necessarily disjoint anymore, even if the original ones are. This leads us to investigate the case of a point moving among completely general (possibly intersecting) convex obstacles. Here we obtain an $\Omega(n^d)$ lower bound and an $O(n^{(d-1)(d/2+1)})$ upper bound. This gives $\Theta(n^2)$ complexity in the plane. In $\mathbb{R}^3$ we get an $\Omega(n^3)$ lower bound and an $O(n^4)$ upper bound, and the gap becomes larger in higher dimensions. We conjecture that the truth is closer to the lower bound.

The obstacles arising as Minkowski sums of disjoint convex bodies with (the central reflection of) the same robot are not completely arbitrary. For instance, in the plane they form a collection of *pseudodisks* (the boundaries of any two intersect in at most two points), and we can exploit this property and show a linear upper bound on the motion complexity. In higher dimensions we currently have no improvement over the general case, although we can only give an $\Omega(n^2)$ lower bound in $\mathbb{R}^3$.

We also consider the special case of a point robot moving among disjoint convex obstacles. In this case we obtain a $\Theta(n)$ bound for the necessary number of segments¹ to connect two points, in any dimension. This result follows easily from the planar case.

So far we have concentrated on the combinatorial bounds. The next step is to obtain efficient algorithms for computing paths with a small number of segments. We present such algorithms for the planar case. To retain generality, we suppose that the convex obstacles are given by suitable *oracles* which can answer various queries (like “find a line separating obstacles B_i, B_j ”). We obtain a nearly linear-time algorithm for the pseudodisk case and a nearly $O(n^2)$ algorithm for the general intersecting case, provided the oracles can be implemented efficiently. If the obstacles are convex polygons or they are bounded by algebraic curves of bounded degree (such as circular arcs), the oracle calls can be implemented in logarithmic time. We also have an efficient implementation for obstacles that are Minkowski sums. This leads to the following result: if the obstacles are convex polygons with a total of m vertices and the robot is a convex k -gon, a translational motion with $O(n)$ segments can be found in $O(n \log^2(k + m))$ time (assuming the vertices of each polygon are stored in a clockwise order in an array given *a priori*).

We now review the related results and approaches in the literature. As far as we know, the problem for general convex obstacles in our sense has never been considered; we regard this new point of view as one of the main contributions of our

¹ This number is usually called the *link distance* between p and q .

work. The usual approach is to consider obstacles with piecewise linear boundaries and express the bounds in terms of their total combinatorial complexity. Our approach can bring a substantial improvement, for example, when the piecewise linear surface arises as an approximation of some curved convex shape.

The motion-planning problem has received considerable attention; we mention only the works more or less directly related to our results, refer to [L], and [SS] for a wider background and literature.

A usual approach to the motion-planning problem is to compute a full combinatorial description of the *free space* (the set of all admissible placements of the robot). This approach was exploited by Kedem *et al.* [KLPS] for the translational motion of a convex robot among convex obstacles in the plane. If the robot is a convex k -gon and the environment consists of n convex polygons with a total of m edges, the complexity of this free space is $O(kn + m)$, and it can be computed in time $O(km \log^2(k + m))$.

Our approach is closer to another motion-planning method, which tries to replace the whole environment by some “simpler” subset, typically of smaller dimension. This is sometimes referred to as the *retraction method*. For instance, a generalized Voronoi diagram (a Voronoi diagram of the obstacle edges, where distances are measured with a distance that is determined by the shape of R) can be computed in the plane. This Voronoi diagram captures the connectivity information about the free space, and can be used to perform motion planning. The complexity of this diagram is in $O(kn + m)$ as well, and it can be computed in time $O(km \log(k + m))$, see [ChD]. Recently, however, Sifrony [S] has shown that the diagram can actually be stored in space $O(k + m)$ such that motion-planning queries which ask for the existence of a path can be answered in time $O(\log m)$. The running time for the computation of the compacted diagram is still $O(mk \log n)$. Furthermore, Kao and Mount [KM] have shown that a path can actually be computed in time $O(m \log m \log^2 k)$. Although they can describe it in space $O(k + m)$, the path computed by this method still has complexity $\Theta(km)$. If both k and m are large and m is significantly larger than n , our results thus provide a significant improvement.

An extension of the algorithms mentioned above to the case of nonlinear obstacle boundaries (circular arcs, say) appears nontrivial, while our approach only requires a relatively straightforward implementation of the oracle. Such nonlinear obstacles are considered important in practical situations [AY].

Another elegant method of the retraction type is due to Canny and Donald [CD]. Roughly speaking, for every pair of features of the obstacles, they define a “generalized bisector,” and they look for the path in a subset of the environment defined by these bisectors. This resembles our methods, and their work also contains topological considerations with a similar flavor to our “expansion lemma” (Lemma 2). In one sense their method is more general, since it can also handle rotational motions, etc. On the other hand, with our more restricted assumptions, we are able to get results for completely general convex obstacles, while the result of [CD] is (necessarily) expressed in terms of features of the obstacles. Also, they only give the results for motion planning in two and three dimensions (with rotation and translation and with the robot and obstacles bounded by algebraic surfaces). Canny’s general work [C] on road maps in algebraic varieties will produce paths in any dimension, but again the

shape and complexity of the path depends on the shape and complexity of the obstacles.

2. Preliminaries and Notation

By $[n]$ we denote the set $\{1, 2, \dots, n\}$, and $\binom{A}{k}$ denotes the set of all k -element subsets of a set A .

For a set $X \subseteq \mathbb{R}^d$, we let $\bar{X}$ denote the closure of X and ∂X the boundary of X .

We mainly consider the path of a point-sized robot among a family $\mathcal{B} = \{B_1, \dots, B_n\}$ of n convex obstacles. The endpoints of the path are denoted by p and q .

Throughout this paper we consider the obstacles as *open* convex bodies. This means that the path is allowed to “glide” along the boundary of the obstacles and so the environment is closed.

In various topological considerations, it is convenient to have the environment also bounded and thus compact. Then we may assume that the obstacles are bounded as well.

In order to model the case of unbounded obstacles also (in practice some obstacle might be too large to go around it) we imagine that the obstacles are enclosed in a large enough closed cube U . We can regard U as being bounded by $2d$ special open convex obstacles (“walls”) $W_1, \dots, W_{2d}$, which can also be assumed to be bounded. Our original obstacles are then allowed to intersect the walls. In this way we can assume that all the obstacles we deal with are bounded, and the relevant connected component E of their complement $\mathbb{R}^d \setminus \bigcup_{i=1}^n B_i$ is also bounded and thus compact. Note that we do not lose generality here, since in the unbounded case we can always take a large enough cube so that it contains a path with minimum complexity.

Let us now assume that $B_1, \dots, B_n$ are open convex polytopes. In such a situation, a shortest p - q path exists within E , and each shortest path γ is piecewise linear (since E can be triangulated into a finite number of simplices, and within every simplex we could replace a curved portion of γ by a segment).

Let γ be a shortest p - q path. We call a point $x \in \gamma$, where two segments of different directions meet, a *turn* of γ .

Lemma 1. *Let γ be a shortest p - q path and let x be a turn of γ . Then x belongs to the closure of a $(d - 2)$ -dimensional face of some polytope B_i .*

Proof. Let the turn x be a common endpoint of the segments a, b of the path γ . Let π be the plane containing a and b , and let α be the angular region in π defined by a, b with apex x . Since γ is a shortest path, an arbitrarily small neighborhood of x in the angular region α has to be intersected by some obstacle (otherwise we could take a shortcut on γ). Since we have finitely many obstacles, one obstacle B_i has to intersect an arbitrarily small neighborhood of x in α , so

$x \in \partial B_i$. Since B_i is an open convex polytope, x cannot be an interior point of a facet of B_i . $\square$

When a turn x of a path γ belongs to the closure of a $(d - 2)$ -face of polytope B_i , we say that B_i *supports* γ at x .

3. Expanding Convex Bodies

In this section we develop a general technique for bounding the complexity of paths among convex obstacles; it is applied to obtain specific bounds in the following sections. We enlarge the given obstacles in such a way that p and q remain connected by a path, and we ensure that the enlarged obstacles are convex polytopes with a suitably bounded combinatorial complexity. Then we find a path using the combinatorial features (edges, facets) of these polytopes.

The following lemma shows that we can safely expand the convex obstacles into larger ones as long as this does not create any new d -wise intersections between them.

Lemma 2. *Let $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ and $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$ be collections of bounded open convex bodies in $\mathbb{R}^d$, and let p, q be points not belonging to any B_i or C_i . Suppose that the following conditions are satisfied:*

- (i) $B_i \subseteq C_i$, $i = 1, \dots, n$.
- (ii) For any set $I \subseteq \left\{ \begin{smallmatrix} [n] \\ d \end{smallmatrix} \right\}$ with $\bigcap_{i \in I} B_i = \emptyset$ we have $\bigcap_{i \in I} C_i = \emptyset$.

Then p and q can be connected by a path avoiding all B_i if and only if they can be connected by a path avoiding all C_i .

Here is an intuitive explanation of why the lemma holds, at least in dimensions two and three. Imagine that we start with p and q connected by a rubber string avoiding the B_i 's. Then, as we continuously expand the B_i 's to the C_i 's, we can continuously deform the path, keeping it outside the growing bodies. In the plane, a moment when this is not possible anymore may appear only when two previously disjoint bodies touch and cut our path. In 3-space the critical situation is when three bodies form a narrow channel through which the path passes, and this channel closes as a triple intersection appears. It appears that this idea can be turned into a rigorous proof, but technically it seems easier (or shorter, at least) to use powerful ready-made tools from algebraic topology (most notably, Alexander's duality theorem) for another type of proof.

The intuition behind this proof is as follows. Suppose that there is no p - q path avoiding the C_i 's. Then there must be a "reason" for this, namely, a $(d - 1)$ -dimensional surface c within $\bigcup \mathcal{C}$ that separates p and q . In the plane it must be a topological circle; in higher dimensions it can be a topological sphere or something more complicated, like a torus. Now this separating "(hyper)surface" c can be triangulated finely enough, and then "pulled" into $\bigcup \mathcal{B}$ using the intersection

condition, while it retains its separating property. Thus p and q are also separated by the bodies of $\mathcal{B}$.

The formal proof uses notions and results of algebraic topology (homology and cohomology theory) [M]. Its understanding is not needed for the rest of this paper.

Proof. The proof proceeds in two stages. First we prove a similar lemma where, instead of open convex bodies, the B_i 's and C_i 's are compact polyhedra (which matches the assumptions of the theorems we use). Then we derive the form with open bodies by a limit argument.

Lemma 3. *Let $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ and $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$ be collections of compact convex polyhedra in $\mathbb{R}^d$, satisfying conditions (i) and (ii) of Lemma 2. Then p and q can be connected by a path avoiding all B_i if and only if they can be connected by a path avoiding all C_i .*

Proof. Set $X_{\mathcal{B}} = \bigcup \mathcal{B}$, $X_{\mathcal{C}} = \bigcup \mathcal{C}$. Suppose that points p, q belong to different components of $\mathbb{R}^d \setminus X_{\mathcal{C}}$; we want to show that they also belong to different components of $\mathbb{R}^d \setminus X_{\mathcal{B}}$. We consider the zero-dimensional reduced homology groups $\tilde{H}_0(\mathbb{R}^d \setminus X_{\mathcal{C}})$ and $\tilde{H}_0(\mathbb{R}^d \setminus X_{\mathcal{B}})$. We can choose the 0-cycle $p - q$ as one of the free generators of $\tilde{H}_0(X_{\mathcal{C}})$. We show that there is an injective homomorphism mapping $\tilde{H}_0(\mathbb{R}^d \setminus X_{\mathcal{C}})$ into $\tilde{H}_0(\mathbb{R}^d \setminus X_{\mathcal{B}})$. This homomorphism will be natural; the relevant fact needed here is that it preserves the identity of the components. Such a homomorphism cannot map the 0-cycle $p - q$ to zero, and hence p and q lie in distinct components also in $\mathbb{R}^d \setminus X_{\mathcal{B}}$. (Notice that we cannot, in general, expect the homomorphism to be onto, since some connected components might have disappeared completely when the B_i 's are expanded to the C_i 's.)

Each B_i is a closed convex polyhedron bounded by a finite number of facet hyperplanes. If we triangulate the arrangement of all the facet hyperplanes, $\mathbb{R}^d$ becomes a simplicial complex,² $X_{\mathcal{B}}$ can be viewed as its subcomplex, and each B_i is a subcomplex of $X_{\mathcal{B}}$.

We can now apply (a particular case of) Alexander's duality theorem [M], and we get that the zero-dimensional reduced homology group of $\mathbb{R}^d \setminus X_{\mathcal{B}}$ is (naturally) isomorphic to the $(d - 1)$ -dimensional reduced cohomology group $\tilde{H}^{d-1}(X_{\mathcal{B}})$.

As mentioned above, the sets of $\mathcal{B}$ form a covering of $X_{\mathcal{B}}$ by its subcomplexes. We let $\mathcal{M}(\mathcal{B})$ denote the nerve of this covering. This is an abstract simplicial complex, with vertex set $[n]$ (the set indexing $\mathcal{B}$). A subset $\sigma \subseteq [n]$ forms a simplex in $\mathcal{M}(\mathcal{B})$ iff $\bigcap_{i \in \sigma} B_i \neq \emptyset$. Since the B_i 's are convex, any set in the covering and also any intersection of these sets is convex and thus contractible. In such a situation it is known that the homology groups of $X_{\mathcal{B}}$ and the homology groups of $\mathcal{M}(\mathcal{B})$ are isomorphic (the Folkman–Leray theorem [R]). By the universal coefficient theorem, the cohomology groups are also isomorphic, so in particular $\tilde{H}_0(\mathbb{R}^d \setminus X_{\mathcal{B}}) \cong$

² This is not quite true, since the simplices should be compact, while “unbounded simplices” in the triangulation of unbounded cells of a hyperplane arrangement are also usually permitted. To remedy this, we can either add a point ∞ at infinity and work with $\mathbb{R}^d \cup \{\infty\}$ as a topological sphere, or we can triangulate the unbounded cells by countably many ordinary simplices.

$\tilde{H}^{d-1}(\mathcal{M}(\mathcal{B}))$). A similar argument applies to $\mathcal{E}$, so it remains to provide a (natural) injective homomorphism of $\tilde{H}^{d-1}(\mathcal{M}(\mathcal{E}))$ into $\tilde{H}^{d-1}(\mathcal{M}(\mathcal{B}))$.

By condition (i) of the lemma, $\mathcal{M}(\mathcal{B})$ can be regarded as a subcomplex of $\mathcal{M}(\mathcal{E})$, and by (ii) the sets of $(d-1)$ -dimensional simplices are identical in both complexes. The inclusion map $i: \mathcal{M}(\mathcal{B}) \rightarrow \mathcal{M}(\mathcal{E})$ induces a homomorphism of the $(d-1)$ -dimensional cohomology groups $\tilde{H}^{d-1}(\mathcal{M}(\mathcal{E})) \rightarrow \tilde{H}^{d-1}(\mathcal{M}(\mathcal{B}))$, and it is easy to check that the fact that $\mathcal{M}(\mathcal{E})$ has no new $(d-1)$ -dimensional simplices implies that this homomorphism is injective. This concludes the proof of Lemma 3. $\square$

It remains to derive Lemma 2 from the polyhedral version, Lemma 3. Before we do this, we prove a lemma showing how to enlarge the obstacles into polytopes of bounded complexity without creating new d -wise intersections. This lemma is also useful later.

Lemma 4. *Let $B_1, \dots, B_n$ be bounded open convex bodies in $\mathbb{R}^d$, and let p, q be points not belonging to any B_i . Then bounded open convex polyhedra $C_1, \dots, C_n$ avoiding p, q can be constructed such that $B_i \subseteq C_i$, for any $I \in \binom{[n]}{d}$, $\bigcap_{i \in I} C_i = \emptyset$ iff $\bigcap_{i \in I} B_i = \emptyset$, and each C_i has $O(n^{d-1})$ facets.*

Proof. Let $\tilde{U}$ be a large enough open cube containing all B_i . We construct $C_1, \dots, C_n$ inductively. Suppose that $C_1, \dots, C_i$ have already been defined. Let $\mathcal{F}_i$ be a collection of convex bodies defined as follows. $\mathcal{F}_i$ contains p, q and each nonempty $(d-1)$ -wise intersection of the form

$$F_{i_1, \dots, i_{d-1}} = C_{i_1} \cap C_{i_2} \cap \dots \cap C_{i_k} \cap B_{i_{k+1}} \cap B_{i_{k+2}} \cap \dots \cap B_{i_{d-1}}$$

with $i_1 < i_2 < \dots < i_k \leq i < i+2 \leq i_{k+1} < \dots < i_{d-1}$, such that $B_{i+1} \cap F_{i_1, \dots, i_{d-1}} = \emptyset$. For each $F \in \mathcal{F}_i$, we find an open half-space h_F containing B_{i+1} and disjoint from F . We then let C_{i+1} be the intersection of all half-spaces h_F , $F \in \mathcal{F}_i$, and the cube $\tilde{U}$. It is straightforward to verify that the family $\{C_1, \dots, C_n\}$ thus constructed satisfies the requirements of the lemma. $\square$

We are ready to finish the proof of Lemma 2. Let $\mathcal{B}, \mathcal{E}$ be the families as in Lemma 2. Since all the bodies are bounded, we may enclose them, together with the points p, q , in the interior of a sufficiently large cube U , and treat the walls of U as $O(1)$ additional obstacles. In this way we may suppose that p, q are contained in compact components of $\mathbb{R}^d \setminus \bigcup \mathcal{B}$.

First we deal with the “smaller” bodies of $\mathcal{B}$. For any $I \in \binom{[n]}{d}$ choose a point $x_I \in \bigcap_{j \in I} B_j$, provided that this intersection is nonempty, and define

$$B'_i = \text{conv} \left\{ x_I; i \in I \in \binom{[n]}{d}, \bigcap_{j \in I} B_j \neq \emptyset \right\}.$$

Each B'_i is a closed convex polyhedron contained in B_i , and it is easily seen that if the claim of Lemma 2 fails for some $\mathcal{B} = \{B_1, \dots, B_n\}$, then it also fails for $\mathcal{B}' = \{B'_1, \dots, B'_n\}$.

Let us now look at the C_i 's. These are first expanded into bounded complexity open polyhedra C'_i using Lemma 4. Each of these is a bounded open convex polytope whose combinatorial complexity is bounded solely in terms of n and d . They have the same d -wise intersection pattern as the C_i 's. Therefore it suffices to prove Lemma 2 for the families $\mathcal{B}'$ (compact polyhedra) and $\mathcal{C}' = \{C'_1, \dots, C'_n\}$.

We still cannot use Lemma 3 directly, since $\mathcal{C}'$ consists of *open* polyhedra: we thus express every C'_i as a limit of closed polyhedra. The bounded complexity of the C'_i is important for arguing about the limit of paths. For every $C'_i \in \mathcal{C}'$, choose a sequence, $C_i^{(1)}, C_i^{(2)}, \dots$ of closed convex polyhedra with the following properties:

- For all j , $B'_i \subseteq C_i^{(j)} \subseteq C'_i$.
- $C'_i = \bigcup_{j=1}^{\infty} C_i^{(j)}$.
- The number of facets of $C_i^{(j)}$ is no larger than the number of facets of C'_i .

It is straightforward to produce such a sequence, by shifting the facet hyperplanes of C_i inward by sufficiently small amounts converging to zero. Set $\mathcal{C}_j = \{C_1^{(j)}, \dots, C_n^{(j)}\}$. For every j , the families $\mathcal{B}'$ and $\mathcal{C}_j$ satisfy the assumptions of the “polyhedral” version—Lemma 3—so if there is a p - q path avoiding $\mathcal{B}'$, then there is also a path avoiding $\mathcal{C}_j$. It is easy to see, using the bounded combinatorial complexity of the polyhedra of $\mathcal{C}_j$, that a piecewise linear p - q path avoiding $\mathcal{C}_j$ with the number of segments bounded by some number N independent of j must also exist. Let γ_j denote some such path.

All the paths γ_j are contained in the cube U , which is a compact set. Because every γ_j is determined by at most an N -tuple of points (the endpoints of the segments) in U , a subsequence $\gamma_{j_1}, \gamma_{j_2}, \dots$, which converges to a path γ (meaning that each γ_{j_i} has the same number of defining points as γ and the defining points of γ_{j_i} converge to the appropriate defining points of γ), exists. It is straightforward to verify that this limit path γ avoids $\mathcal{C}'$. This finishes the proof of Lemma 2. $\square$

4. General Obstacles in Higher Dimensions

In this section we consider arbitrarily intersecting convex obstacles.

Theorem 5. *Let $\mathcal{B}$ be a collection of n bounded convex bodies in $\mathbb{R}^d$. Then any two points in the same connected component of $\mathbb{R}^d \setminus \bigcup \mathcal{B}$ can be connected by a path consisting of $O(n^{(d-1)(d/2+1)})$ segments.*

Proof. Using Lemma 4, we first replace $\mathcal{B}$ by a family $\mathcal{C}$ of bounded open convex polytopes, each with $O(n^{d-1})$ facets, without creating new d -wise intersections. Lemma 2 guarantees that a p - q path avoiding the obstacles of $\mathcal{C}$ exists. Hence, the theorem follows from Lemma 6 below. $\square$

Lemma 6. *Let $\mathcal{E}$ be a collection of n bounded open convex polytopes in $\mathbb{R}^d$, each defined as the intersection of at most m half-spaces. Then any two points p, q in the same connected component of $\mathbb{R}^d \setminus \bigcup \mathcal{E}$ can be connected by a path consisting of $O(n^{d-1}m^{\lfloor d/2 \rfloor})$ segments.*

Proof. Consider a shortest p - q path γ . To each turn x of the path γ , we assign a certain point $v(x)$. Let $C_0 \in \mathcal{E}$ be a supporting obstacle for γ at x .

For dimension $d = 2$, $v(x)$ will be simply equal to x , which is a vertex of C_0 . For dimension $d = 3$, x is either a vertex of C_0 , in which case we set $v(x) = x$, or it lies on an edge e of C_0 . We go along this edge from x in one (arbitrarily chosen) direction, until we either reach its end vertex u (a vertex of C_0) or we reach a point v where e enters the interior of some other obstacle $C_1 \in \mathcal{E}$. In the first case we set $v(x) = u$, in the second case $v(x) = v$. For a general dimension, the formal definition of $v(x)$ is somewhat complicated, and can be done inductively as follows.

We set $y_1 = x$, $\mathcal{D}_1 = \{C_0\}$, and $H_1 = \{h_1, h_2\}$, where h_1, h_2 are two facet hyperplanes of C_0 containing x (they exist by Lemma 1). These will be objects entering the first step of the construction. At the beginning of a general (i th) step, we will have a current point y_i , a current collection $\mathcal{D}_i \subseteq \mathcal{E}$ of at most i polytopes of $\mathcal{E}$, and a current set H_i of $i + 1$ hyperplanes, each being a facet hyperplane of some $C \in \mathcal{D}_i$. The intersection $f_i = \bigcap H_i$ of the hyperplanes is a $(d - i - 1)$ -flat. The point y_i lies in f_i and on the boundary of the intersection $\bigcap \mathcal{D}_i$ of the polytopes of $\mathcal{D}_i$.

In the i th step we choose a ray ρ_i originating at y_i and lying within the flat f_i . We consider the points of ρ_i linearly ordered along ρ_i (y_i is the first point). For every $C \in \mathcal{E}$, we define a point $v_C \in \rho_i$; we distinguish two cases:

- (i) If $C \in \mathcal{D}_i$, we let v_C be the last point of ρ_i in the boundary ∂C . It is easy to check that as ρ_i is leaving the boundary at v_C , it has to cross some facet hyperplane of C at v_C . We denote this hyperplane by h_C .
- (ii) If $C \notin \mathcal{D}_i$, we let v_C be the last point of ρ_i before ρ_i enters the (open) polytope C . If ρ_i does not intersect C , we put $v_C = \infty$. If v_C is not ∞ , then again ρ_i crosses a facet hyperplane of C at v_C ; we choose one such hyperplane and denote it by h_C .

We now let y_{i+1} be the first point among the v_C , $C \in \mathcal{E}$ (it may happen that $y_{i+1} = y_i$). We choose some C with $v_C = y_{i+1}$; if there are more possibilities, we prefer a $C \in \mathcal{D}_i$ (case (i)) over $C \notin \mathcal{D}_i$ (case (ii)). We define $H_{i+1} = H_i \cup \{h_C\}$ and $\mathcal{D}_{i+1} = \mathcal{D}_i \cup \{C\}$. After $d - 2$ steps the construction finishes, with the flat $f_{d-1} = \bigcap H_{d-1}$ being zero-dimensional and containing only the point y_{d-1} , which we define to be $v(x)$.

We need to establish that the step preserves the invariant (the above listed properties of $y_i, H_i, \mathcal{D}_i$), which is straightforward. The newly added hyperplane h_C is not parallel to the ray ρ_i , and thus not to f_i , hence $\dim f_{i+1} < \dim f_i$. The point y_i lies on the boundary of the intersection of the polytopes of $\mathcal{D}_i$, and as we go along the ray ρ_i , we can only leave this boundary by passing some point v_C for $C \in \mathcal{D}_i$. If

y_{i+1} is defined as v_C for $C \in \mathcal{D}_i$ (case (i)), we have $\mathcal{D}_{i+1} = \mathcal{D}_i$ and so y_{i+1} lies in $\partial(\cap \mathcal{D}_i)$. If the polytope C defining y_{i+1} does not belong to $\mathcal{D}_i$, there is some portion of ρ_i after y_{i+1} which still lies in the boundary of $\cap \mathcal{D}_i$, but it also lies inside the new C . This implies that y_{i+1} is on the boundary of $\cap \mathcal{D}_{i+1} = C \cap \cap \mathcal{D}_i$ as well.

The above construction guarantees that x can be connected to $v(x)$ by the piecewise linear path $x = y_1, y_2, y_3, \dots, y_{d-1} = v(x)$, which consists of at most $d - 2$ segments and avoids the obstacles. A portion of the path γ between two turns x, x' with $v(x) = v(x')$ can thus be replaced by a portion consisting of at most $2(d - 2)$ segments.

We estimate the number of vertices which may appear as $v(x)$ for some x . By construction, any such point is a vertex of the intersection of some $(d - 1)$ -tuple of the obstacles. Such an intersection is defined by $O(m)$ half-spaces, thus it has $O(m^{\lfloor d/2 \rfloor})$ vertices. There are $O(n^{d-1})$ choices for the $(d - 1)$ -tuple of obstacles, yielding at most $O(n^{d-1} m^{\lfloor d/2 \rfloor})$ vertices in total. Hence a path with $O(n^{d-1} m^{\lfloor d/2 \rfloor})$ turns exists as claimed. $\square$

Aronov and Sharir [AS2], [AS1] proved that the maximum combinatorial complexity of the union of n polygons in $\mathbb{R}^2$ with a total of M edges is $\Theta(n^2 + M\alpha(n))$, and that the complexity of the union of n polytopes in $\mathbb{R}^3$ with a total of M facets is $O(n^3 + Mn \log^2 n)$. Lemma 6 implies that the maximum path complexity in a scene consisting of n convex polygons in $\mathbb{R}^2$ with M edges in total is $O(M)$, and that the maximum path complexity in a scene consisting of n convex polytopes in $\mathbb{R}^3$ with M facets in total is $O(nM)$. Hence, our bounds resemble the bounds on the union size proved by Aronov and Sharir. It would be interesting to show a similar result in higher dimensions.

It would be tempting to conjecture that the relatively crude $O(n^{d-1})$ upper bound on the number of facets of the expanded polytopes can be improved, using some clever construction for these polytopes (rather than choosing the separating half-spaces arbitrarily as we did in Lemma 4). However, we have an example in $\mathbb{R}^3$ which rules out this direct way of improvement: $O(n)$ convex polytopes in $\mathbb{R}^3$ can be constructed such that none of them can be enlarged without creating new triple intersections, and the combinatorial complexity of the union is n^4 (even if we only count the vertices defined by an edge of one polytope and a facet of another). We omit the description of this example, as it is somewhat complicated.

As for the actual path complexity, we have a construction showing that the bound in Theorem 5 cannot be improved below $\Omega(n^d)$, for $d \geq 2$; in particular, the result is tight in the plane. The construction proceeds by induction on dimension d . Consider first the planar version (see Fig. 1 for $n = 4$). We define n^2 points A_{ij} , $i, j = 0, 1, \dots, n - 1$, on n rays starting in the origin: $A_{ij} = (1 + \varepsilon i)(\cos(2\pi j/n), \sin(2\pi j/n))$, for $i, j = 0, \dots, n - 1$ and $\varepsilon > 0$ sufficiently small. Let obstacle B_i be the convex hull of $A_{i0}, A_{n-i,1}, A_{i,2}, A_{n-i,3}, \dots$. The union of these n obstacles looks like a star. We can now fill the n concave niches by other obstacles, leaving a small pathway around the inner star. The length of this pathway is n^2 , which proves an $\Omega(n^2)$ lower bound for the planar intersecting case.

Fig. 1. A lower bound example.

Now suppose that we have a set of n obstacles, a starting position p , and a certain goal position q in $\mathbb{R}^d$, such that any path between a p and q has $\Omega(n^d)$ links. We extend each obstacle to $\mathbb{R}^{d+1}$ by taking the Cartesian product of the obstacle with, say, the interval $[0, 1]$. Let us call the extra coordinate the x_{d+1} -coordinate. We now add n new obstacles, which are the hyperplanes $h_i: x_{d+1} = i/(n+1)$, $i = 1, \dots, n$. The hyperplanes h_i partition $\mathbb{R}^{d+1}$ into $n+1$ slabs. Observe that if we restrict our attention to one of the hyperplanes h_i we have exactly the d -dimensional situation. Let p_i and q_i be the points on h_i that correspond to the starting and goal position for the d -dimensional situation. If i is odd we punch a small hole in h_i around p_i , and if i is even we punch a small hole in h_i around q_i . Now it is easy to see that any path from p_1 to q_n needs $\Omega(n^{d+1})$ links: it must first go from p_1 to q_2 using $\Omega(n^d)$ links, then it must return to p_3 using another $\Omega(n^d)$ links, and so on. Notice that the punctured hyperplanes are not convex, but they can be replaced by three (intersecting) half-hyperplanes. It is straightforward to adapt the construction so that it only uses bounded obstacles. The total number of obstacles used remains $O(n)$ for any fixed d .

5. Bounds and Algorithms for the Planar Case

We now consider the planar case of the problem, and we turn our attention to the algorithmic aspect.

Touching Planar Obstacles

Here we consider a family $\mathcal{B}$ of n closed convex obstacles with disjoint interiors. We show that the path complexity is linear in this case, and we give an efficient algorithm.

Our algorithm proceeds as follows: For every obstacle $B \in \mathcal{B}$, we identify a leftmost and a rightmost point of B , and we let s_B be the segment connecting those two points. If B is unbounded, the line segment may degenerate to a semi-infinite ray, or even to a line. We then compute the trapezoidal map defined by this set of

line segments.³ This trapezoidal map consists of at most $3n + 1$ trapezoids. Every (bounded) trapezoid has two vertical sides, and it is bounded from above and below by line segments corresponding to two of the obstacles. The interior of a trapezoid is intersected by these two obstacles only. Since they are convex we can separate their interiors by a line. Any point inside the trapezoid and outside the obstacles can be connected to this line by a vertical line segment. This means that any two points inside the trapezoid can be connected by at most three links, provided that they can be connected at all inside the trapezoid. (This need not be the case, because the two obstacles inside the trapezoid are allowed to touch.)

Consider a path γ connecting start and goal positions p and q . We can assume that γ passes through every trapezoid at most twice, since every trapezoid intersects at most two connected components of $\mathbb{R}^2 \setminus \bigcup \mathcal{B}$. Within a trapezoid, we can replace a connected portion of γ by at most three links. Hence, there is a path using a linear number of links.

We now turn this existence proof into an efficient algorithm. Consider, for every trapezoid, the two points where the line separating its two obstacles intersects its vertical sides (they are necessarily outside the obstacles). We construct a graph whose nodes are these points, and whose edges connect two nodes if the corresponding points lie on the boundary of the same trapezoid and can be connected inside the trapezoid. We connect p, q to such points in their respective trapezoids, and we find the path using a graph search algorithm.

In this algorithm, we suppose that the obstacles are described by an oracle. It is sufficient that each of the following operations can be performed by the oracle, in a worst-case time Q : finding the segment s_B for a given B , finding a line separating given obstacles B, B' , and finding the common part of their boundaries when they touch. We thus obtain.

Theorem 7. *Let $\mathcal{B}$ be a set of n convex obstacles in the plane described by an oracle as above, and let p, q be two points. Then, in $O(n(Q + \log n))$ time, we can construct a p - q path with $O(n)$ links avoiding $\mathcal{B}$ or determine that no p - q path exists.*

In particular, if the obstacles are polygons with m vertices in total, the oracle operations can be implemented in time $Q = O(\log m)$, assuming that the vertices of each polygon are stored in an array in a clockwise sorted order [DK]. Hence, a path with $O(n)$ links can be found in $O(n \log m)$ time in this situation.

The Motion-Planning Problem for a Convex Robot

Here we come back to the problem which motivated our research. Given an open convex robot R and a collection $\mathcal{B}^* = \{B_1^*, \dots, B_n^*\}$ of disjoint open convex obstacles in the plane, as well as positions p and q , find a path that translationally

³ A trapezoidal map induced by a set of line segments is the partition of the plane obtained as follows: through every endpoint, we add a vertical segment not intersecting the other segments and maximal with respect to this property. The trapezoidal map is now defined by the original segments plus the added vertical segments.

Fig. 2. Example of a trapezoidal map.

moves R from p to q . This problem can be reduced to the case of a point robot by replacing $\mathcal{B}^*$ by $\mathcal{B} = \{B_1, \dots, B_n\}$, where $B_i = B_i^* - R = \{b - x \mid b \in B_i^*, x \in R\}$ is the Minkowski sum of B_i^* and the image of the robot under central reflection. We thus obtain a new set of obstacles in the so-called configuration space. Any point x in this space corresponds to a certain position of the robot; we denote the robot at this position by $R(x)$. The complement of the union of the new obstacles, that is, the region $\mathbb{R}^2 \setminus \bigcup \mathcal{B}$, is called the *free space*; we denote it by FP .

Kedem *et al.* [KLPS] have shown that the obstacles in $\mathcal{B}$ are *pseudodisks*: their boundaries intersect in at most two points and the total number of such reflex vertices on the boundary of the union of $\mathcal{B}$ is $O(n)$. This allows us to compute a linear-size “trapezoidal” map for FP , defined as follows. Consider the points on ∂FP that are either the leftmost or the rightmost point of some B_i , or a reflex vertex of FP that is not locally extreme in the x -direction. For each such point we add a vertical segment of maximal length that contains the point and is contained in FP . Figure 2 gives an illustration. (The line through the intersection points of ∂B_i and ∂B_j in this figure is used later.) From the trapezoidal map we then compute a path using a graph search algorithm, in about the same way as in the touching case.

Next we describe the algorithm in more detail. As before, we assume that the obstacles are given by suitable oracles. More precisely, we need two oracles, one for the original bodies B_i^* and the original robot R , and one for the Minkowski sums B_i . The first oracle should be able to answer the following questions:

- Given i , find the segment $s_{B_i^*}$ (as defined in the previous subsection).
- Given i and a certain placement of R , decide whether B_i^* intersects R .
- Given a certain placement of R and a vertical line l , compute $R \cap l$.
- Given i and a vertical line l , compute $B_i^* \cap l$.

For the Minkowski sums B_i we need:

- Given i and a vertical line l , compute $B_i \cap l$.
- Given i, j , decide whether B_i, B_j are disjoint and, if yes, return a separating line, if not, return the intersections of their boundaries.

Let Q^* be the worst-case time the oracle on the original bodies and the robot needs, and let Q be the time for the oracle on the Minkowski sums.

The first phase of the algorithm—the computation of the trapezoidal map—is done using a randomized incremental algorithm that follows the abstract framework of Boissonnat *et al.* [BDS⁺]. (A randomized incremental algorithm to compute the union of a set of pseudodisks has already been presented by Miller and Sharir [MS]. Their algorithm is basically the same as ours. They assume, however, that the pseudodisks have constant complexity, so we cannot use their result directly.) We assume the reader is familiar with the algorithm and terminology of Boissonnat *et al.* In a generic step of the algorithm, when we add the next obstacle B_i , we must perform two tasks: locating the trapezoids in the current decomposition intersected by B_i , and creating new trapezoids. To create the new trapezoids we must compute the intersection points of the boundary of a trapezoid in the current map with B_i . This can be done with a constant number of calls to the oracle for the Minkowski sums. Locating the intersected trapezoids is done by searching the history. Here the basic operation is to test whether B_i intersects a given trapezoid in the history. We could use the oracle for B_i , but we can also test this using calls to the—possibly cheaper—oracle for the original bodies only.

Lemma 8. *It is possible to test in $O(Q^*)$ time whether an obstacle B_i intersects a trapezoid Δ .*

Proof. By definition, B_i intersects Δ if and only if there is a point $x \in \Delta$ such that $R(x)$ intersects B_i^* . So testing whether B_i intersects Δ amounts to testing whether B_i^* intersects the sweeping area $\text{Sweep}(\Delta) = \bigcup \{R(x) : x \in \Delta\}$.

Let B_j and $B_{j'}$ be the obstacles that bound Δ from above and from below, respectively. Let x_l and x'_l be the upper and lower endpoint of the left boundary segment of Δ . Define x_r and x'_r analogously for the right boundary segment of Δ . It is possible that $x_l = x'_l$ or $x_r = x'_r$ (but not both), but this does not change the argument. Figure 3 gives an example for a disk-shaped robot; $\partial\Delta$ is shown thick and dashed, whereas $\partial\text{Sweep}(\Delta)$ is thick and solid.

The boundary of $\text{Sweep}(\Delta)$ consists of parts of the boundaries of B_j^* , $B_{j'}^*$, $R(x_l)$, $R(x'_l)$, $R(x_r)$, and $R(x'_r)$, together with the vertical line segment s_l that connects the

Fig. 3. The sweeping area of a trapezoid.

leftmost point of $R(x_l)$ to the leftmost point of $R(x'_l)$, and the vertical line segment s_r that connects the rightmost point of $R(x_r)$ to the rightmost point of $R(x'_r)$. We first test whether B_i^* intersects $R(x_l)$, $R(x'_l)$, $R(x_r)$, or $R(x'_r)$, using our oracle. If an intersection is found we are done. Otherwise we cut $\partial\text{Sweep}(\Delta)$ into an upper part β and a lower part β' by removing the segments s_l and s_r . Because the original obstacles are disjoint and we have already tested B_i^* for intersections with $R(x_l)$, $R(x'_l)$, $R(x_r)$, and $R(x'_r)$, we know that B_i^* does not intersect β or β' . Since $\text{Sweep}(\Delta)$ is monotone (any vertical line intersects it in a single interval), B_i^* intersects $\text{Sweep}(\Delta)$ if and only if it hits β when it is translated upward, but it does not hit β' . Next we describe how to test this.

We only describe how to test whether B_i^* will hit $\beta \cap \partial B_j^*$; the other tests are performed similarly. We first try to find a vertical line that intersects both B_i^* and $\beta \cap \partial B_j^*$: this can be done in constant time after computing the segment $s_{B_j^*}$, since we know the extreme points of $\beta \cap \partial B_j^*$. If there is no such line, then B_i^* will not hit $\beta \cap \partial B_j^*$. Otherwise, we compute the intersection points of the line with B_i^* and B_j^* in time Q^* , to check whether B_i^* lies above $\beta \cap \partial B_j^*$ or below. $\square$

To analyze the algorithm we use the results from [BDS⁺]. Because the number of trapezoids in the decomposition is linear in the number of obstacles, the expected number of created trapezoids is $O(n)$ and the expected number of trapezoid-obstacle tests when searching the history is $O(n \log n)$. Hence, the trapezoidal map can be constructed in $O(n(Q^* \log n + Q))$ randomized time.

Having computed the trapezoidal map, we proceed as in the touching case: we compute, for every trapezoid in the decomposition, a separating line, we connect the start and goal positions to the lines in their respective trapezoids by a vertical segment, and we find a path using a graph search algorithm. The only nontrivial operation in this procedure is the computation of a separating line. Consider a trapezoid Δ , and let $B_j, B_{j'}$ be the obstacles that bound Δ from above and below, respectively. If B_j and $B_{j'}$ are disjoint, then life is easy: the oracle will provide us with a separating line. However, it may very well happen that B_j and $B_{j'}$ intersect as in Figure 2. Fortunately, the oracle can also be used in this case: the line through the two intersections points of ∂B_j and $\partial B_{j'}$ must separate B_j from $B_{j'}$ inside Δ , that is, it intersects $\partial \Delta$ only in the two vertical sides of Δ . See Figure 2.

Hence, the second phase of the algorithm takes $O(nQ)$ time, and we obtain the following result.

Theorem 9. *Let $\mathcal{B}^*$ be a set of n disjoint open convex obstacles, let R be an open convex robot, and let p, q be two positions of R . Assume that oracles as described above for the original bodies B_i^* and R , and for the Minkowski sums B_i are available. Then in randomized time $O(n(Q^* \log n + Q))$, one can either compute a translational motion of R between p and q consisting of $O(n)$ links, or determine that no p - q translational motion is possible.*

We now consider the case where the obstacles are convex polygons with m vertices in total and R is a convex k -gon. We assume that we are given arrays that store the vertices of the polygons in order. Dobkin and Kirkpatrick [DK] have shown

that the oracle for the original bodies B_i^* and R can be implemented to work in $O(\log(m + k))$ time. We now turn our attention to the oracle on the Minkowski sums.

Lemma 10. *The oracle calls for the Minkowski sums can be implemented in $O(\log^2(k + m))$ time.*

Proof. We only describe the second, more difficult, operation that the oracle must be able to perform. First consider the case where B_i and B_j do not intersect. As noted above, Dobkin and Kirkpatrick [DK] have given an algorithm for computing a separating line for two convex polygons in logarithmic time. However, we cannot afford to compute each B_i explicitly. Fortunately, it is enough to produce, for an implicitly given chain on ∂B_i , an edge that splits the chain into parts of roughly equal size.

This can be done as follows. Consider a chain β on the boundary of B_i ; the ordered sequence of edges of β corresponds to a merge of the ordered sequence of edges of certain chains β' and β'' on B_i^* and R , respectively. Take the median edge e on the longer of the two chains, say β' , and search with (the slope of) this edge in the other chain β'' . In this way we obtain the edges in β that are adjacent to the edge of β corresponding to e and, hence, the endpoints of this edge. The number of edges on either side of e in β is no more than $|\beta'|/2 + |\beta''| \leq 3|\beta|/4$. So the time needed for a basic operation in the algorithm of Dobkin and Kirkpatrick is $O(\log(k + m))$, and we can find a separating line for B_i and B_j in $O(\log^2(k + m))$ time in total.

In the foregoing we have assumed that B_i and B_j are disjoint. If we run the algorithm of Dobkin and Kirkpatrick on two intersecting polygons, it returns a point x in $B_i \cap B_j$. Our oracle must provide some more information, namely, the intersection of the boundaries of B_i and B_j . (Because B_i and B_j are Minkowski sums, $\partial B_i \cap \partial B_j$ consists of two points or one line segment. To simplify the discussion, we assume that the intersection consists of exactly two points: the adaptation to the general case is straightforward.) First, we compute in $O(\log^2(m + k))$ time by binary search the intersections of the horizontal line through x with ∂B_i and ∂B_j . Two of the four points that we find must be points that lie on the boundary of one of the polygons and are contained in the other. Let q be such a point and assume that it lies on ∂B_i and inside B_j . Next, we find a point $q' \in \partial B_i$ that lies outside B_j . Point q' corresponds to a placement of R where it touches B_i^* and misses B_j^* . Hence, we can find q' by computing a line that separates B_i^* and B_j^* , and placing R such that it touches B_i and does not intersect this line. Such a placement can be found in logarithmic time.

So we have two points available on ∂B_i , one inside B_j and one outside B_j . We can now use binary search to find the two intersection points of ∂B_i and ∂B_j . The basic operation during the binary search is this: is a given point on ∂B_i inside B_j or not? This amounts to checking whether a given placement of R intersects B_j^* , so it can be answered in $O(\log(m + k))$ time. Hence, the binary search takes $O(\log^2(m + k))$ time in total. $\square$

The lemma also holds if the obstacles and the robot are convex figures bounded by Jordan arcs, instead of straight line segments, because the technique of Dobkin and Kirkpatrick also works (with a few trivial adaptations) in this situation. Thus the translational motion planning can be performed in $O(n \log^2(k + m))$ time, yielding a path with $O(n)$ links. Note that the complexity of the union of the polygons in $\mathcal{B}$ can be as high as $\Theta(kn + m)$.

General Obstacles

For the general case of arbitrary open convex obstacles, we proceed as in Section 4. We replace the family $\mathcal{B}$ by another family $\mathcal{C}$. For every pair $B_i, B_j \in \mathcal{B}$ with B_i and B_j disjoint, we find a separating line ℓ_{ij} . We also compute lines ℓ_i^p and ℓ_i^q separating B_i from p and q . We then replace B_i by C_i , which is defined as the intersection of the half-spaces bounded by the ℓ_{ij} , ℓ_i^p , and ℓ_i^q . By Lemma 2, p and q can be connected among the B_i exactly if they can be connected among the C_i . Every C_i is an open convex polygon with at most $n + 1$ edges. A shortest path between p and q only bends at convex vertices of the union of the C_i . Such a vertex is a vertex of some polygon in $\mathcal{C}$, so there are at most $O(n^2)$ bends.

To compute a path, we consider the arrangement defined by all $O(n^2)$ edges of the polygons in $\mathcal{C}$. The points p and q can be connected iff they lie in the same cell of the arrangement. This cell has complexity $O(n^2\alpha(n))$ and can be computed in expected time $O(n^2\alpha(n) \log n)$, where $\alpha(n)$ is the functional inverse of the Ackermann-function [CEG⁺]. Once the cell is computed, a path with $O(n^2)$ links can be computed in time linear in the complexity of the cell. The oracle must be able to return a separating line for two obstacles (or say they intersect) and also to separate p, q from the obstacles. If Q denotes the running time for an oracle call, we get the following result.

Theorem 11. *Let $\mathcal{B}$ be a set of n convex obstacles in the plane described by an oracle as above, and let p and q be two points. In expected time $O(n^2Q + n^2\alpha(n) \log n)$, one can compute a p - q path with $O(n^2)$ links or determine that no p - q path exists.*

For polygons with m vertices in total, the oracle is implemented in $O(\log m)$ time, yielding total expected time $O(n^2\alpha(n) \log n + n^2 \log m)$.

6. Disjoint Obstacles in Higher Dimensions

Using the results on the planar case from the previous section, it is not difficult to prove the following:

Theorem 12. *Let $\mathcal{B} = \{B_1, \dots, B_n\}$ be a collection of pairwise disjoint, possibly unbounded, open convex bodies in $\mathbb{R}^d$, $d \geq 2$. Then any two points in the same connected component of $\mathbb{R}^d \setminus \bigcup \mathcal{B}$ can be connected by a path consisting of $O(n)$ segments.*

Proof. If all obstacles are bounded, the proof is very easy. Indeed, in this case the intersections of the obstacles with any plane π are also disjoint and unbounded. We can therefore fix any plane π passing through the given points p, q , and connect these points by a path lying within π . Hence we may use the planar result from Section 5.

The situation becomes slightly more complicated if we also allow unbounded obstacles, since then the complement may be disconnected. It can also happen that the two points p and q can be connected in $\mathbb{R}^d$, but not within an arbitrarily chosen hyperplane containing p and q .

The following easy lemma characterizes the situation when an open convex body disconnects $\mathbb{R}^d$:

Lemma 13. *For an open convex body $B \subseteq \mathbb{R}^d$, the following statements are equivalent:*

- (i) $\mathbb{R}^d \setminus B$ is not path connected.
- (ii) ∂B is not path connected.
- (iii) B is an open slab, that is, the set of points lying between two parallel hyperplanes.

Proof. The equivalence of (i) and (ii) is easy, as ∂B is a deformation retract of $\mathbb{R}^d \setminus B$. To see that (i) and (ii) imply (iii), consider two points x, y lying in distinct path components of ∂B , and let h_x, h_y be supporting hyperplanes of B at these points, respectively. Clearly, $h_x \neq h_y$ (otherwise the segment xy is contained in ∂B). If h_x, h_y are not parallel, x, y can be connected by a path going within h_x and then within h_y , contradicting (i). So let h_x and h_y be parallel. If z were any point between them not contained in B , we could separate z from B by a hyperplane h_z . This hyperplane cannot be parallel to h_x, h_y , and thus x and y can be connected by a path going within h_x, h_z , and h_y . Thus B must be the whole slab between h_x and h_y . $\square$

If we have a hyperplane h through p, q such that p, q cannot be connected within h , there must be one obstacle B_i “responsible” for it. Indeed, if none of the B_i would separate h , then we could take the segment pq and replace its portion within each $h \cap B_i$ by a path within $\partial B \cap h$, obtaining a p – q path within h .

We show that one obstacle can only be responsible for one hyperplane. Suppose not, let B_i separate p from q in two hyperplanes h, h' . Then both $B_i \cap h, B_i \cap h'$ are slabs (in h, h' , respectively), and it is easy to see that the convex hull of such two $(d - 1)$ -dimensional slabs contains a hyperplane separating p and q in $\mathbb{R}^d$.

Thus, there are at most n bad hyperplanes and a path exists within any hyperplane through p, q different from these. Then, by induction on the dimension, we may assume that such a hyperplane contains a path of $O(n)$ links, and we are done. $\square$

It is easy to construct n unbounded convex sets and two points p, q , such that any path between p and q needs a linear number of links, thus Theorem 12 is optimal. For completeness, we also give an example of n disjoint *bounded* convex obstacles that enforce a linear number of links. Consider a set of $n/2d$ axis-parallel hyper-

Fig. 4. Lower bound for the disjoint case.

cubes $C_1, \dots, C_{n/2d}$, each centered around the origin, such that $C_1 \subset C_2 \subset \dots \subset C_{n/2d}$. We make C_{i+1} relatively large with respect to C_i . The set of obstacles in our example is the set of facets of the hypercubes. However, the facets of a hypercube touch each other, so we need to modify them slightly. This is done as follows. Call the facets of a hypercube which are orthogonal to the x_d -axis its x_d -facets. The x_i -facets of the cubes are shrunk in the x_{i+1} - up to x_d -direction by a small amount ε , and they are extended by 2ε in the x_1 - up to x_{i-1} -direction. If we do this for each of the hypercubes C_i we end up with a set of n disjoint convex obstacles. See Fig. 4 for a planar example of the construction. It is easy to see that it takes $\Omega(n)$ links to go from the origin to a point outside the largest cube $C_{n/2d}$.

7. Conclusion and Extensions

We regard the current work as an initial study of the problems considered, and many questions remain open. It would be nice to tighten the combinatorial bounds for dimensions three and more; most notably, there is a large gap between the bounds for the translational motion of a convex robot among convex obstacles in $\mathbb{R}^3$, and a new idea seems to be needed for improving the upper bound.

There are also some other intermediate cases to be considered. For instance, the motion-planning problem for a point robot moving among disjoint convex obstacles is not very realistic, since by our definitions the robot can move even through infinitely thin passages. It would be more natural to allow for disjoint but touching obstacles, where the infinitely thin passages are sealed up. We have some preliminary results for this case in higher dimensions (the planar case being subsumed by the results of this paper).

The algorithmic aspects in $\mathbb{R}^3$ are almost untouched. Also it remains to be seen to what extent our approach can help in handling practical motion-planning problems.

One might ask about the possibility of a further generalization of our results to, say, translational and rotational movement of a convex robot among convex obstacles. It turns out that no upper bound on the number of “elementary motions” can be given for arbitrary convex obstacles (where an “elementary motion” would be defined by bounded degree algebraic curves). This can be seen by considering a rod (a segment in the plane) among obstacles arranged in such a way that the rod has to move along a unique curve (in the free space) enforced by a triple contact with three

Fig. 5. A rod moving among convex obstacles.

suitably shaped convex obstacles. See Fig. 5. Such a curve can only be described in terms of the obstacles, and thus such problems need a different framework.

Acknowledgments

We would like to thank E. Vogt for his great help in the technical realization of the proof of Lemma 2. We also thank Herbert Edelsbrunner and Petr Savický for useful discussions and comments.

References

- [AY] H. Alt and C. K. Yap. Motion planning in *cl*-environments: a case for a realistic model. *Proc. 1st Workshop on Algorithms and Data Structures (WADS)*, pp. 373–380. Lecture Notes in Computer Science, vol. 382. Springer-Verlag, Berlin, 1989.
- [AS1] B. Aronov and M. Sharir. The union of convex polyhedra in three dimensions. *Proc. 34th Annual IEEE Symp. on Foundations of Computer Science*, pp. 519–527, 1993.
- [AS2] B. Aronov and M. Sharir. The common exterior of convex polygons in the plane. Manuscript, 1994.
- [BDS⁺] J.-D. Boissonnat, O. Devillers, R. Schott, M. Teillaud, and M. Yvinec. Applications of random sampling to on-line algorithms in computational geometry. *Discrete Comput. Geom.*, 8:51–71, 1992.
- [C] J. Canny. *The Complexity of Robot Motion*. ACM–MIT Press Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1987.
- [CD] J. Canny and B. R. Donald. Simplified Voronoi diagrams. *Discrete Comput. Geom.*, 3:219–236, 1988.
- [CEG⁺] B. Chazelle, H. Edelsbrunner, L. Guibas, M. Sharir, and J. Snoeyink. Computing a face in an arrangement of line segments. *Proc. 2nd ACM–SLAM Symp. on Discrete Algorithms*, pp. 441–448, 1991.
- [CLD] L. P. Chew and R. L. Drysdale, III. Voronoi diagrams based on convex distance functions. *Proc. 1st Annual ACM Symp. on Computational Geometry*, pp. 235–244, 1985.
- [DK] D. P. Dobkin and D. G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.

- [KM] T. C. Kao and D. M. Mount. An algorithm for computing compacted Voronoi diagrams defined by convex distance functions. *Proc. Third Canadian Conf. on Computational Geometry*, pp. 104–109, 1991.
- [KLPS] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.
- [L] J.-C. Latombe. *Robot Motion Planning*. Kluwer, Dordrecht, 1991.
- [MS] N. Miller and M. Sharir. Efficient randomized algorithm for constructing the union of fat triangles and of pseudodiscs. Unpublished manuscript.
- [M] J. R. Munkers. *Elements of Algebraic Topology*. Addison-Wesley, Reading, MA, 1984.
- [R] J. J. Rothman. *An Introduction to Algebraic Topology*. Graduate Texts in Mathematics, vol. 119. Springer-Verlag, New York, 1988.
- [S] S. Sifrony. A real nearly linear algorithm for translating a convex polygon. Technical Report 476, New York University, 1989.
- [SS] J. T. Schwartz and M. Sharir. Algorithmic motion planning in robotics. In J. van Leeuwen, editor, *Algorithms and Complexity*, pp. 391–430. Elsevier, Amsterdam, 1990.

Received May 25, 1993, and in revised form August 30, 1994.