Approximation Algorithms for Curvature-Constrained Shortest Paths *

Hongyan Wang† Pankaj K. Agarwal†

July 31, 1996

Abstract

Let B be a point robot in the plane, whose path is constrained to have curvature of at most 1, and let Ω be a set of polygonal obstacles with n vertices. We study the collision-free, optimal path-planning problem for B. Given a parameter ε, we present an $O((n^2/\varepsilon^4)\log n)$-time algorithm for computing a collision-free, curvature-constrained path between two given positions, whose length is at most $(1+\varepsilon)$ times the length of an optimal robust path (a path is robust if it remains collision-free even if certain positions on the path are perturbed). Our algorithm thus runs significantly faster than the previously best known algorithm by Jacobs and Canny whose running time is $O((\frac{n+L}{\varepsilon^2})^2 + n^2(\frac{n+L}{\varepsilon^2})\log n)$, where L is the total edge length of the obstacles. More importantly, the running time of our algorithm does not depend on the size of obstacles. The path returned by this algorithm is not necessarily robust. We present an $O((n^{2.5}/\varepsilon^4)\log n)$-time algorithm that returns a robust path whose length is at most $(1+\varepsilon)$ times the length of an optimal robust path. We also give a stronger characterization of curvature-constrained shortest paths, which, apart from being crucial for our algorithm, is interesting in its own right. Roughly speaking, we prove that, except in some special cases, a shortest path touches obstacles at points that have a visible vertex nearby.

*Work by the first author has been supported by NSF Grant NSF-IRI-91-00681, Rome Labs Contracts F30602-94-C-0037, ARPA/SISTO contracts N00014-91-J-1985, and N00014-92-C-0182 under subcontract KI-92-01-0182. Work by the second author has been supported by a National Science Foundation Grant CCR-93–01259, by an Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by an NYI award and matching funds from Xerox Corporation, and by a grant from the U.S.-Israeli Binational Science Foundation.

†Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129, USA. E-mail: hongyan@cs.duke.edu, pankaj@cs.duke.edu

1 Introduction

The *path-planning* problem involves planning a collision-free path for a robot moving amid obstacles. This is one of the main problems in robotics, and has been widely studied (see, e.g., the book by Latombe [30] and the survey paper by Schwartz and Sharir [46]). In the simplest form of the motion planning, given a moving robot B, a set O of obstacles, and a pair of placements I and F of B, we wish to find a continuous, collision-free path for B from I to F. This problem is PSPACE-complete [10, 42], and efficient algorithms have been developed for several special cases [46]. Most of these algorithms, however, do not take into account the dynamic constraints (for instance, velocity/acceleration bounds, curvature bounds), the so-called *nonholonomic constraints*, of a real robot imposed by its physical limitations. Although there has been considerable recent work in the robotics literature (see [3, 5, 7, 9, 22, 26, 28, 29, 31, 32, 34, 35, 38, 48, 49] and references therein) on nonholonomic motion-planning problems, relatively little theoretical work has been done on these important problems, because they are considerably harder than holonomic motion-planning problems.

In holonomic motion planning, the placement of a robot with k degrees of freedom is determined by a tuple of k (typically real) parameters, each describing one degree of freedom. The set of all placements is called the *configuration space*, and the set of placements at which the robot does not intersect any obstacles is called the *free configuration space*. There exists a path between an initial placement and a final placement if and only if the two placements lie within the same (path-) connected component of the free configuration space. This is not necessarily true if the robot has to obey nonholonomic constraints. In nonholonomic motion planning, usually a placement is not enough to describe the robot. Instead, a robot is completely described by its *state*, consisting of the k parameters and their derivatives (see [30] for a more detailed discussion), which makes the problem considerably harder.

In this paper, we study the path-planning problem for a point robot whose path is constrained to have curvature of at most 1. More formally, given a continuous differentiable path $P : I \to \mathbb{R}^2$ parameterized by arc length $s \in I$, the *average curvature* of P in the interval $[s_1, s_2] \subseteq I$ is defined by $\|P'(s_1) - P'(s_2)\|/|s_1 - s_2|$. We require that the robot's path has an average curvature at most 1 in every interval. This restriction corresponds naturally to constraints imposed by a steering mechanism on a *car-like* robot (see [30] for a formal description), because the path traced out by the middle point between the two rear wheels has an instantaneous curvature of $\frac{1}{\lambda}\tan\phi$, where ϕ is the steering angle and λ a parameter of the car. The maximum curvature of the path is therefore $\frac{1}{\lambda}\tan\phi_{\max}$, assuming $\phi_{\max}$ is the maximum steering angle.

1.1 Previous work

Dubins [20] was perhaps the first to study the curvature-constrained shortest paths. He proved that, in absence of obstacles, a curvature-constrained shortest path from any start position to any final position consists of at most 3 segments, each of which is either a straight line or an arc of a unit-radius circle. Reeds and Shepp [41] extended this obstacle-free characterization to robots that are allowed to make reversals. Using ideas from control theory, Boissonnat *et al.* [6] gave an alternative proof for both cases, and recently Sussmann [47] was able to extend the characterization for the 3-dimensional case. In presence of obstacles, Fortune and Wilfong [21] gave a $2^{poly(n,m)}$-time algorithm, where n is the total number of vertices in the polygons defining the obstacles and m the number of bits of precision with which all points are specified; their algorithm only decides whether a path is feasible, without necessarily finding one. Jacobs and Canny [11, 25] gave an $O((\frac{n+L}{\varepsilon^2})^2 + n^2(\frac{n+L}{\varepsilon^2})\log n)$-time algorithm that finds an approximate path whose length is no more than $(1+\varepsilon)$ times the length of a shortest ε-*robust* path, where L is the total edge length of the obstacles. (Informally, a path is ε-robust if perturbations of certain points along the path by $\pm\varepsilon/2$ — in distance or in angle — do not violate the feasibility of the path.) The path returned by this algorithm is not necessarily robust. They also presented an $O(n^4 \log n + (\frac{n+L}{\varepsilon^2})^2)$-time algorithm that computes an $(\varepsilon/2)$-robust path whose length is no more than $(1+\varepsilon)$-times the length of an optimal ε-robust path. For the restricted case of *moderate obstacles*, i.e., when the curvature of the boundary of obstacles is also bounded by 1, Agarwal *et al.* [2] give efficient approximation algorithms, and Boissonnat and Lazard [8] give a polynomial-time algorithm for computing the exact shortest paths for the case when the edges of obstacles are circular arcs of unit radius. Wilfong [48] studies a restricted problem in which the robot must stay on one of m line segments (thought of as "lanes"), except to turn between lanes. For a scene with n obstacle vertices, his algorithm preprocesses the scene in time $O(m^2(n^2 + \log m))$, following which queries are answered in time $O(m^2)$. There has also been work on computing curvature-constrained paths when B is allowed to make reversals [3, 28, 29, 33, 37]. Other, more general, dynamic constraints are considered in [12, 13, 19, 39, 44].

1.2 Model and results

Let B be a point robot. A *position* X for B is a pair $(\text{LOC}(X), \text{VEC}(X))$, where $\text{LOC}(X)$ is a point representing the location of the robot and $\text{VEC}(X)$ is an angle between 0 and 2π, representing its orientation, both lying in the plane. A *path* is an oriented curve; $\|\cdot\|$ denotes the length of a path. A path is called *legal* if its average curvature is at most 1 in every interval. Let Ω be a set of disjoint polygonal obstacles, with a total of n vertices. A legal path is *feasible* (with respect to Ω) if it does not intersect the interior of any obstacle of Ω. A path Π from a position X to another position Y is *optimal* if it is a feasible path with the minimum arc length, where the minimum is taken over all feasible paths from X to Y (it can be shown that the minimum always exists). Finally, following the definition in [25], a path Π is called *robust* if it is feasible, and even after a small perturbation in orientation (resp. location) at each position where Π passes through an obstacle vertex (resp. edge), Π remains feasible and passes through the same set of obstacle vertices and edges. Π is called δ-*robust* if a perturbation of up to $\pm\delta/2$ can be made in the orientation and location. Π is an *optimal* δ-*robust* path from X to Y if its length is the minimum over all δ-robust paths from X to Y.

The first main result of our paper is an efficient algorithm for computing an approximation to the optimal robust path (Section 3). More precisely, given an obstacle environment Ω with n vertices, two positions I and F, and a parameter ε, we present an $O((n^2/\varepsilon^4)\log n)$-time algorithm that computes a feasible path from I to F whose length is at most $(1+\varepsilon)$ times the length of an optimal ε-robust path from I to F. Compared with the Jacobs-Canny algorithm [25], our algorithm is not only considerably faster in terms of the complexity of Ω, but more importantly, the running time is independent of L, the total edge length of Ω. The second improvement is rather significant because one cannot assume L to be small. For unconstrained optimal path planning, one can scale down the environment arbitrarily (to reduce the value of L), compute a shortest path in the scaled environment, and then scale the optimal path back. But this scaling trick does not work for curvature-constrained shortest paths, as the scaling will change the curvature of the path as well. The path returned by our algorithm is not necessarily robust. We can, however, modify the algorithm to compute an $(\varepsilon/2)$-robust path in time $O((n^{2.5}/\varepsilon^4)\log n)$, whose length is no more than $(1+\varepsilon)$-times the length of an optimal ε-robust path. In fact, an $O(n^{2+\gamma}/\varepsilon^4)$-time algorithm can be obtained, using the recent range-searching data structures [1], but we will not discuss this improvement in this paper.

Our algorithms are based on a stronger characterization of optimal (and of optimal robust) paths, which is interesting in its own right (Section 2.1). Roughly speaking, we prove that, except in some very special cases, an optimal path touches an edge e of Ω only at points near a vertex v (v is not necessarily one of the endpoints of e), such that these points are visible from v. In other words, we can ignore the portions of the edges of Ω that are not near to any vertex of Ω, which enables us to develop an algorithm whose running time does not depend on the size of Ω.

2 Characterization of Optimal Paths

In this section we characterize optimal paths in presence of obstacles, in the plane. Let Π be a feasible path. We call a nonempty subpath of Π a *C-segment* (resp. *L-segment*) if it is a circular arc of unit radius (resp. line segment) and maximal. Suppose Π consists of a C-segment, L-segment, and a C-segment, then we will say that Π is of type CLC. This notion can be generalized to an arbitrarily long sequence. Dubins [20] proved the following result.

Theorem 2.1 (Dubins [20]) *In an obstacle-free environment, an optimal path between any two positions is of type CCC or CLC, or a substring thereof.*

We will refer to such paths as *Dubins paths*. In the presence of obstacles, Fortune and Wilfong [21] and Jacobs and Canny [25] observed that any subpath of an optimal path that does not touch any obstacle except at the endpoints is a Dubins path. In particular,

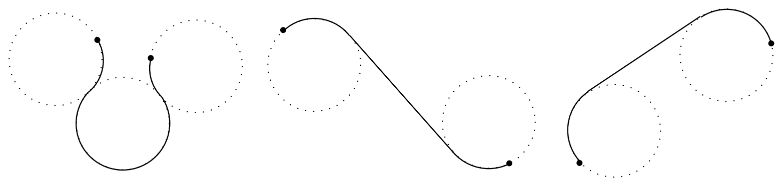

Figure 1: Examples of Dubins paths.

Theorem 2.2 (Fortune-Wilfong [21], Jacobs-Canny [25]) *Given an obstacle environment Ω, an initial position I, and a final position F, an optimal path from I to F consists of a sequence $\Pi_1||\cdots||\Pi_k$ of feasible paths, where each Π_i is a Dubins path from a position X_{i-1} to a position X_i, such that $X_0 = I$, $X_k = F$, and, for $0 < i < k$, $\text{LOC}(X_i) \in \partial\Omega$.*

Although Jacobs and Canny [25] did not prove it explicitly, the above theorem holds even for δ-robust optimal paths. (Their approximation algorithm uses this stronger claim implicitly.) This theorem implies that an optimal path is a finite sequence of C- and L-segments, so it can be represented as a finite string over the alphabet $\{C, L\}$. There are, however, infinite number of such paths — an optimal path can touch a vertex at an arbitrary orientation, or it can touch an arbitrary point of an edge. Jacobs and Canny [25] observe that if one is interested only in computing $(1+\varepsilon)$-approximate paths, one can choose a finite set of orientations at which a path can touch a vertex, and can also choose a finite set of points on each edge e at which a path can touch e. The number of points chosen on an edge is proportional to the length of the edge. That is why the running time of their algorithm depends on the total length of edges of Ω. We circumvent this problem by proving a stronger property of optimal paths.

A feasible C-segment is called *free* if it does not intersect Ω; *anchored* if it touches $\partial\Omega$ at two or more points; and *semi-free* if it touches $\partial\Omega$ at exactly one point. By Theorems 2.1 and 2.2, an optimal path cannot have two consecutive free C-segments. Moreover, there are only finite number of circles that touch $\partial\Omega$ at two or more points (assuming that there are no two edges parallel at distance 1), so there are only finite number of circles that may contain anchored C-segments. We thus need a better understanding of semi-free C-segments.

The following theorem states the main result of this section.

Theorem 2.3 *If w is a semi-free C-segment of an optimal path, then there is a vertex v of Ω within distance 15 from the intersection point ξ of w and $\partial\Omega$ that is visible from ξ (i.e., the interior of the segment ξv does not intersect Ω).*

2.1 Proof of Theorem 2.3

We will prove the theorem by a sequence of lemmata. We begin with a few notation and simple observations. For a circle C and two points $a, b \in C$, let $C[a, b]$ denote the arc of C

from a to b in the clockwise direction. For an oriented path Π and two points $a, b \in \Pi$, let $\Pi[a, b]$ denote the subpath of Π from a to b.

If ξ is a vertex of Ω, then there is nothing to prove, so assume that ξ lies in the interior of an edge $e = pq$. Without loss of generality, assume that e lies on the x-axis, that w lies below the x-axis, and that w is oriented clockwise. We can also assume that $d(p, \xi), d(q, \xi) > 15$. We will regard $\text{LOC}(I)$ and $\text{LOC}(F)$ as the vertices of Ω.

We divide the proof of Theorem 2.3 into several cases. For each case, we prove the existence of a closed, simply-connected region R satisfying the following properties:

P1. R lies on or below the x-axis, R contains ξ, and either ∂R contains a vertex of Ω or the interior of R contains a point of $\partial\Omega$.

P2. If an edge $g \in \Omega$ intersects the interior of R, it crosses ∂R in at most one point, which implies that at least one of the endpoints of g lies inside R.

P3. $d(\xi, x) \leq 15$ for all points $x \in R$.

See Figure 2 for an example. P1–P3 together imply that there is a vertex of Ω within distance 15 from ξ. In order to prove that there is also a vertex within distance 15 from ξ which is visible from ξ, we introduce R^*, the convex hull of R.

Lemma 2.4 *If a simply connected region R satisfies P1–P3, then R^* also satisfies P1–P3.*

Proof: P1 is obvious. If an obstacle edge g crosses ∂R^* at two points, then, using a continuity argument and the fact that R is simply connected, one can prove that g also crosses ∂R at two points, which contradicts property P2 of R. Hence, g crosses ∂R^* in at most one point, thereby proving P2.

For any point $x \in R^*$, the ray emanating from ξ and passing through x intersects ∂R^* at one point, say y (y may be identical to x). If $y \in R$, property P3 follows since $d(\xi, x) \leq d(\xi, y) \leq 15$. Otherwise y lies in the interior of a line segment, both of whose endpoints, say u and v, lie on ∂R. Property P3 also follows since $d(\xi, x) \leq d(\xi, y) \leq \max\{d(\xi, u), d(\xi, v)\} \leq 15$. This completes the proof of the lemma. □

For each obstacle edge e that intersects R^*, clip it within R^*. Let E denote the set of clipped segments. (If an endpoint of an edge lies on ∂R^*, but the edge does not intersect the interior of R^*, we add that endpoint as a degenerate segment to E.) By property P1, $E \neq \emptyset$. We define the following partial ordering on the segments of E. We say that $e_i \prec e_j$ ($e_i, e_j \in E$) if any ray emanating from ξ and intersecting both e_i and e_j intersects e_i before intersecting e_j (That is, viewed from ξ, e_j cannot occlude any portion of e_i). Such an ordering always exists because the segments of E are disjoint, they lie below the x-axis, and ξ lies on the x-axis; see, e.g., [17, 23]. Moreover, this partial ordering can be extended to a total ordering. By the definition of the ordering, every point on the first segment in this ordering is visible from ξ. But one of the endpoints of this segment, say, v, is a vertex of Ω, so we have found an obstacle vertex v within distance 15 from ξ that is visible from ξ.

The following simple lemma, which will be useful in many cases, follows from Lemma 2.4.

Lemma 2.5 *Let α and β be two points on an optimal path Π from I to F such that α, ξ, and β appear in that order along Π, such that $\Pi[\alpha, \beta]$ lies below or on the x-axis, and such that $d(\xi, x) \leq 15$ for all $x \in \Pi[\alpha, \beta]$. Let R be the convex hull of $\Pi[\alpha, \beta]$. If α is an obstacle vertex or if it lies in the interior of an obstacle edge whose supporting line intersects $\Pi[\alpha, \beta]$ at a point other than α, then R satisfies P1–P3. See Figure 2.*

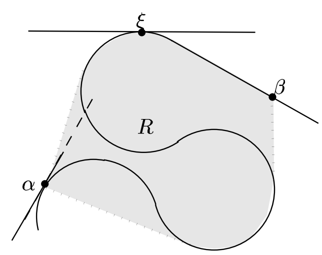

Figure 2: $\Pi[\alpha, \beta]$ and its convex hull.

Proof: Since R is the convex hull of $\Pi[\alpha, \beta]$, R is a simply connected region. We can show that R satisfies P2 and P3, following an argument similar to the one in the proof of Lemma 2.4. If α is a vertex, R satisfies P1. Otherwise the obstacle edge containing α lies in the interior of R near α, because the line supporting this edge intersects the interior of R. Hence, R satisfies P1 also. This completes the proof. □

Since we regard LOC(I) and LOC(F) as vertices of Ω and assume that ξ is not a vertex, the semi-free C-segment w has to be a middle segment of Π. Let w^- (resp. w^+) be the segment of Π immediately before (resp. after) w.

Using a perturbation argument, similar to the one used in [2] (see Figure 3), one can easily prove the following lemma, whose proof is omitted from here.

Lemma 2.6 *If w is neither the first segment nor the last segment of Π, then (i) $\|w\| > \pi$, and (ii) either w^- or w^+ is a C-segment.*

Let us assume that w^- is a C-segment. Let ξ^- (resp. ξ^+) be the point on Π that intersects Ω immediately before (resp. after) ξ. Since $\Pi[\xi^-, \xi]$ is a Dubins path and w^- is a C-segment, $\Pi[\xi^-, \xi]$ consists of two or three C-segments. In either case,

$$d(\xi, x) \leq 6, \tag{1}$$

for any $x \in \Pi[\xi^-, \xi]$. Here onwards, we will assume that

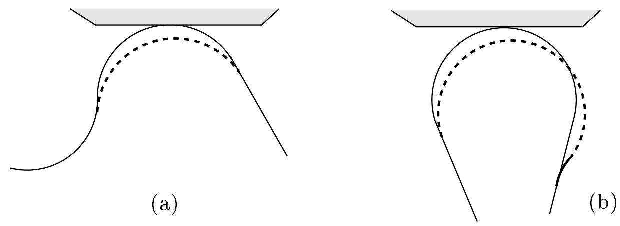

Figure 3: Length reducing perturbations: (a) $\|w\| \leq \pi$, (ii) both w^- and w^+ are L-segments.

($\star$) ξ^- lies in the interior of an edge e^- and the line, ℓ^-, supporting e^- does not intersect $\Pi[\xi^-, \xi]$.

If Π does not satisfies ($\star$), Theorem 2.3 easily follows from Lemma 2.5. Let σ be the intersection point of ℓ^- and the x-axis. Without loss of generality, assume that σ lies to the left of ξ.

We consider the following three cases and prove the existence of a region R satisfying P1–P3 for each case separately.

Case 1. The angle $\angle \xi\sigma\xi^- \geq \pi/6$; see Figure 4.

Case 2. The angle $\angle \xi\sigma\xi^- < \pi/6$.

Case 2.1. $d(\xi, \xi^+) < 8$; see Figure 5.

Case 2.2. $d(\xi, \xi^+) \geq 8$; see Figure 6.

Lemma 2.7 *There exists a region R satisfying P1–P3, for Case 1, i.e., when $\angle \xi\sigma\xi^- \geq \pi/6$.*

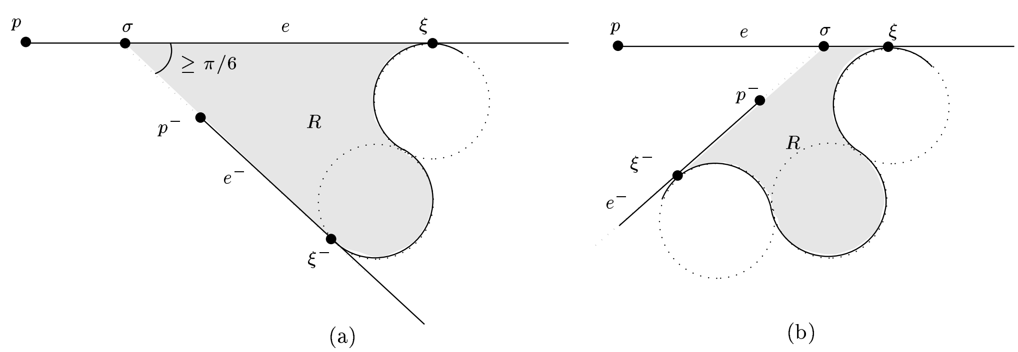

Figure 4: $\angle \xi\sigma\xi^- \geq \pi/6$: (a) $\Pi[\xi^-, \xi]$ has two C-segments; (b) $\Pi[\xi^-, \xi]$ has three C-segments.

Proof: We define R to be the closed region bounded by $\Pi[\xi^-, \xi]$ and the segments $\sigma\xi$ and $\sigma\xi^-$. See Figure 4. By assumption ($\star$), neither x-axis nor ℓ^- intersects the interior of $\Pi[\xi^-, \xi]$, therefore R is a simply connected region. Using the sine law and the fact that $d(\xi, \xi^-) \le 6$ (see (1)), we obtain

$$d(\xi, \sigma) = d(\xi, \xi^-) \frac{\sin \angle \sigma \xi^- \xi}{\sin \angle \xi \sigma \xi^-} \le \frac{d(\xi, \xi^-)}{\sin(\pi/6)} \le \frac{6}{\sin(\pi/6)} \le 12.$$

Since $d(p, \xi) > 15$, we have $\sigma \in e$. The obstacle edges are disjoint, so the left endpoint of e^-, say p^-, has to lie on the segment $\sigma\xi^-$, and thus on the boundary of R, implying that R satisfies P1. R also satisfies P2 because any obstacle edge $g \in \Omega$ can cross ∂R only at the segment σp^-. Finally, for any $x \in R$, $d(\xi, x) \le \max\{d(\xi, \sigma), 6\} \le 12$, therefore R satisfies P3 as well. □

Lemma 2.8 *There exists a region R satisfying P1–P3, for Case 2.1, i.e., when $\angle \xi \sigma \xi^- < \pi/6$ and $d(\xi, \xi^+) < 8$.*

Proof: If ξ^+ is an obstacle vertex, then the claim follows from Lemma 2.5, so assume that ξ^+ lies in the interior of an obstacle edge e^+. Since $d(\xi, \xi^+) \le 8$, ξ^+ lies in a disc D_ξ of radius 8 centered at ξ. Let u (resp. v) be the point on e at distance 8 from ξ to its left (resp. right). If the left endpoint of e^- lies to the left of u, let u' be the point on e^- with the same x-coordinate as u; otherwise, let u' be the left endpoint of e^-. Similarly, if the right endpoint of e^- lies to the right of v, let v' be the point on e^- with the same x-coordinate as v; otherwise let v' be the right endpoint of e^-. See Figure 5. We set R to be the quadrilateral $uu'v'v$. R obviously satisfies P1, because either an endpoint of e^- lies on ∂R (see Figure 5b for an example), or ξ^+ (which is a point on the edge e^+) lies in the interior of R (this is because by the construction of R, R contains the portion of the disc D_ξ in between e and e^-; see Figure 5a for an example). An obstacle edge g cannot intersect uv or $u'v'$ (as they are portions of obstacle edges), and g cannot intersect both uu' and vv' (as this would imply that g intersects $\Pi[\xi^-, \xi]$). Hence R satisfies P2.

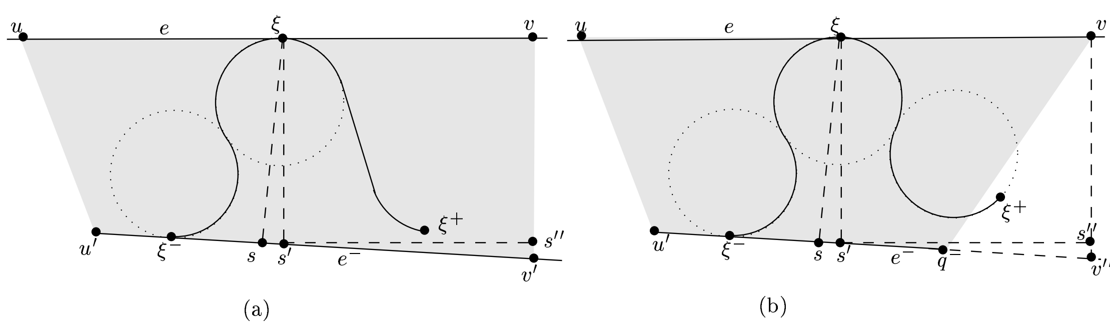

Figure 5: $\angle \xi \sigma \xi^+ < \pi/6$ and $d(\xi, \xi^+) < 8$: (a) ξ^+ lies in the interior of R; (b) q^- lies on ∂R.

Finally, we prove that R satisfies (P3). Let v'' be the point on ℓ^- that has the same x-coordinate as v ($v'' = v'$ if the right endpoint of e^- lies to the right of v). It can be shown that, for any $x \in R$, $d(\xi, x) \leq d(\xi, v'')$ (here we are assuming that ℓ and ℓ^- intersect to the left of ξ). Let s (resp. s') be a point on e^-, such that $\xi s \perp e^-$ (resp. $\xi s' \perp e$). Let s'' be a point on the segment vv'' such that $s's'' \perp vv''$. Notice that $\angle s\xi s' = \angle v''s's'' = \angle \xi\sigma\xi^- < \pi/6$, where σ is the intersection point of lines containing e and e^-. Since

$$d(v, s'') = d(\xi, s') = d(\xi, s) \sec \angle s\xi s' < d(\xi, \xi^-) \sec \frac{\pi}{6} \leq 6 \cdot \frac{2}{\sqrt{3}} = \frac{12}{\sqrt{3}},$$

and

$$d(s'', v'') = d(s', s'') \tan \angle v''s's'' < d(\xi, v) \tan \frac{\pi}{6} = 8 \cdot \frac{1}{\sqrt{3}} = \frac{8}{\sqrt{3}},$$

we have $d(v, v'') < 20/\sqrt{3}$. Therefore for any $x \in R$,

$$d(\xi, x) \leq d(\xi, v'') = \sqrt{d(\xi, v)^2 + d(v, v'')^2} \leq \sqrt{8^2 + (20/\sqrt{3})^2} \leq 15.$$

This completes the proof of the lemma. □

Lemma 2.9 *There exists a region R satisfying P1–P3, for Case 2.2, i.e., when $\angle \xi\sigma\xi^- < \pi/6$ and $d(\xi, \xi^+) \geq 8$.*

Proof: If w^+, the segment of Π following w, is a C-segment, then, by Theorem 2.1, $\Pi[\xi, \xi^+]$ consists of at most three C-segments and $d(\xi, \xi^+) \leq 6$, which contradicts the assumption that $d(\xi, \xi^+) > 8$. Hence w^+ is an L-segment, $\Pi[\xi, \xi^+]$ is of CLC type, and $\|w^+\| \geq d(\xi, \xi^+) - 4 > 4$.

Let C_1 and C_2 be the circles containing w^- and w, respectively, and let h be the line supporting w^+. There are two cases to consider:

(i) h does not intersect C_1,

(ii) h intersects C_1.

Case (i): See Figure 6a. Let C_3 be the other (unit-radius) circle tangent to both h and C_1. Let a (resp. b) be the intersection point of C_3 with C_1 (resp. h). Since $|w^+| > 4$, b lies on the segment w^+.

Let c be the common point of C_1 and C_2. Any line tangent to C_1 at a point on the arc $C_1[c, a]$ intersects the path $\Pi[c, b]$. Hence, by assumption ($\star$), ξ^- can not lie on $C_1[c, a]$, i.e., $a \in \Pi[\xi^-, \xi]$. Define R to be the closed region bounded by $\Pi[a, b]$ and the segment ab; see the shaded region in Figure 6a. If the arc $C_3[a, b]$ does not cross any obstacle edge, we can shorten Π by replacing $\Pi[a, b]$ with $C_3[a, b]$, contradicting the optimality of Π. Hence, an edge of Ω intersects $C_3[a, b]$, thus, also intersects the interior of R; thereby implying that R satisfies P1. Since an edge of Ω can cross ∂R only at ab, P2 is obvious. Finally, every point on ∂R is within distance 15 from ξ, P3 also follows.

Case (ii): See Figure 6b. Let f be the first intersection point of h with C_1. Let ab be the segment tangent to C_1 and C_2 at a and b respectively. Let C_3 be the unit-radius circle

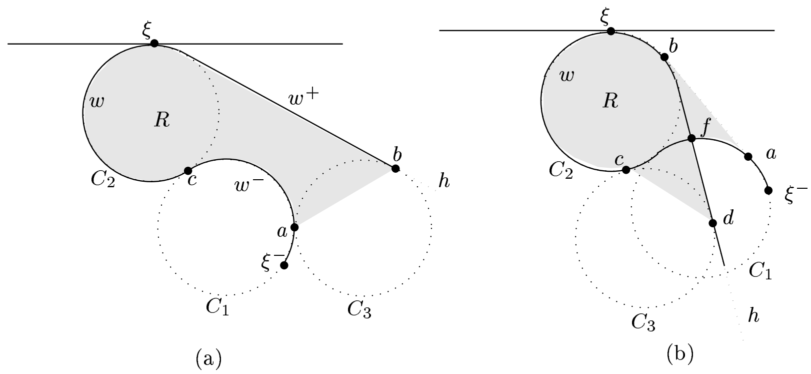

Figure 6: $\angle\xi\sigma\xi^- < \pi/6$ and $d(\xi, \xi^+) \geq 8$: (a) h does not intersect C_1; (b) h intersects C_1.

tangent to C_2 and h at c and d respectively. Since $|w^+| > 4$, both d and f lie on the segment w^+.

The same argument as in case (i) implies that $a \in \Pi[\xi^-, \xi]$. Define R to be the region bounded by the segment ab, the circular arc $C_2[c, b]$ the segments cd and df, and the circular arc $C_1[f, a]$; see the shaded region in Figure 6b. An obstacle edge crosses either the segment ab or the arc $C_3[c, d]$, thus intersecting the interior of R, because otherwise the path obtained by concatenating the segment ab and the circular arcs $C_2[c, b]$ and $C_3[c, d]$ is shorter than the path $\Pi[a, d]$, contradicting the optimality of Π. This means that R satisfies P1. An obstacle edge can cross ∂R only at segments cd and ab, and none of the edges can cross both the segments (because then it would cross Π), P2 follows. Finally, every point on ∂R is within distance 15 from ξ, so P3 also follows. □

These lemmata together complete the proof of Theorem 2.3.

A closer look at the proof reveals that Theorem 2.3 holds even if Π is a δ-robust optimal path. As mentioned in the beginning of the section, Theorem 2.2 is true even for δ-robust optimal paths, therefore it suffices to argue that Lemmas 2.6–2.9 hold for the δ-robust case. Indeed, Lemmas 2.7 and 2.8 do not perform any perturbation and rely only on the feasibility of Π, so they obviously hold for robust paths also. Lemmas 2.6 and 2.9 perturb the path in such a way that the positions at which the perturbed path touches Ω is a subset of those at which the original path touches Ω (see e.g., Figure 6). Hence, by the definition of robustness, if Π is robust, then the perturbed path is also robust. We can therefore conclude

Theorem 2.10 *If w is a semi-free C-segment of an optimal δ-robust path, then there is a vertex v of Ω within distance 15 from the intersection point ξ of w and $\partial\Omega$ that is visible from ξ.*

2.2 Anchored C-segments

Recall that a C-segment is called anchored if it touches $\partial\Omega$ at two (or more) points. We prove a property of anchored C-segments, which will be useful later. We call an anchored C-segment, touching $\partial\Omega$ at two points $p_1 \in e_1$ and $p_2 \in e_2$, *good* if we can find two obstacle vertices v_1 and v_2 (not necessarily distinct) such that v_i is visible from p_i and $d(p_i, v_i) \le 15$. Otherwise, it is called *bad*.

We first mention a simple lemma, which can be proved using a perturbation argument similar to the one in Figure 3. We omit the proof from here, and refer the reader to [2].

Lemma 2.11 *An optimal does not contain a C-segment that touches two parallel obstacle edges at their interior points, and that does not touch any other obstacle edge.*

Lemma 2.12 *If w is a feasible bad anchored C-segment, then no obstacle edge intersects the interior of the circle containing w.*

Proof: Let w be an anchored C-segment touching $\partial\Omega$ at two points $p_1 \in e_1$ and $p_2 \in e_2$. Since w is a bad anchored C-segment, one of p_i, say p_1, has to be at least 15 distance away from the endpoints of e_i. Without loss of generality, let $e = e_1$, $e^- = e_2$, $\xi = p_1$, $\xi^- = p_2$, where e, e^-, ξ, ξ^- are as defined in the last section.

It can be shown that the angle between e and e^- is $< \pi/6$. Otherwise, following the proof of Lemma 2.7, we can find a vertex v_i, such that v_i is visible to p_i and $d(v_i, p_i) \le 15$, contradicting the assumption that w is a bad anchored C-segment.

Let C be the circle containing w. We construct a region R, as in the proof of Lemma 2.8 (except that we set $d(\xi, u) = d(\xi, v) = 2$, since the distance between any point inside C and ξ is ≤ 2). R satisfies P2 (since w is feasible) and P3. If an obstacle edge intersects the interior of C, it also intersects R, making R satisfy P1. If so, we can find for each $i = 1, 2$ a vertex v_i such that v_i is visible from p_i and $d(v_i, p_i) \le 15$, contradicting that w is a bad anchored C-segment. Thus no obstacle edge intersects the interior of C. This proves the lemma. □

Lemma 2.13 *There are only $O(n)$ circles that can contain a feasible bad anchored C-segment.*

Proof: For each edge $e \in \Omega$, let D_e be the Minkowski sum of e and the unit-radius disk. D_e is a race-track bounded by two semi-circles of unit radius and two translated copies of e; see Figure 7. An intersection point p of the straight-line edges of D_e and $D_{e'}$ corresponds to the center of a unit-radius circle tangent to e and e' at their interior points. (Notice that there may be an infinite number of intersection points if e and e' are parallel and unit distance apart.) A point p lies inside D_e if and only if e intersects the interior of the unit-radius circle centered at p. Set $\mathcal{D} = \bigcup_{e \in \Omega} D_e$. A unit-radius circle is tangent to two edges and does not intersect any obstacle edge in its interior, only if its center is at a vertex of $\partial\mathcal{D}$, or

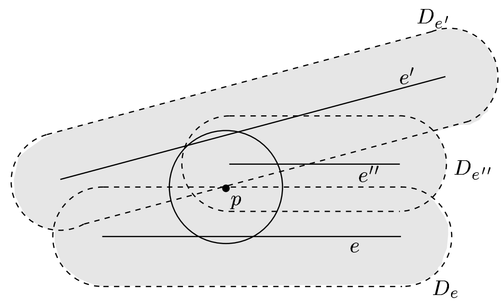

Figure 7: The straight line edges of D_e and $D_{e'}$ intersect at p. The unit radius circle centered at p is tangent to both e and e', and is crossed by e'', whose race-track $D_{e''}$ contains p in its interior.

its center lies on a line segment of $\partial\mathcal{D}$, which is the common part of the race-tracks of two edges parallel and unit distance apart. By Lemma 2.11, a feasible C-segment tangent to two parallel edges can not be bad, thus the number of circles that can contain a feasible bad anchored C-segment is at most the number of vertices of $\partial\mathcal{D}$, which, by a result of Kedem et al. [27], is $O(n)$; so the lemma follows. □

3 Computing Near Optimal Paths

In this section we present an efficient algorithm for computing a feasible path whose length is no more than $(1+\varepsilon)$ times the length of an optimal ε-robust path, for any $\varepsilon > 0$. As in [25], we construct a weighted directed graph $G = (V, E)$, where V is a set of discretized positions. These positions are obtained by discretizing the set of orientations at which a path touches a vertex and the set of points at which a path touches an edge. Jacobs and Canny [25] choose points uniformly spaced along each edge. We, on the other hand, use Theorem 2.3 and Lemma 2.13 to choose points more carefully, as described in Section 3.1. There is an edge $(X, Y) \in E$ from a position X to another position Y if there exists a feasible Dubins path from X to Y. If there is more than one such path, we choose the one with the minimum arc length. The weight of an edge is the arc-length of the chosen Dubins path. We claim that by choosing the proper parameter δ for discretization, for any optimal ε-robust path Π from X to Y, there is a graph path from X to Y in G whose length is at most $(1+\varepsilon)$ times the length of Π. The choice of δ and the proof of this claim are given in Section 4. Therefore, the problem reduces to computing a shortest path in G, which can be done in time $O(|V|^2)$, using Dijkstra's shortest-path algorithm.

3.1 Computing the node set

In this subsection we describe the node set V and an efficient algorithm for computing it. We set $V = \{I, F\} \cup V_1 \cup V_2 \cup V_3$, where each subset V_i corresponds to a specific type of positions. The first set V_1 corresponds to positions located at the vertices of Ω. More precisely, for each vertex v of Ω, V_1 contains the positions $(v, i\delta)$ for $0 \le i \le \lfloor 2\pi/\delta \rfloor$. V_1 can be constructed in $O(n/\delta)$ time in a straight-forward manner.

The second set V_2 corresponds to bad anchored C-segments. For a pair of non-parallel edges e_1, e_2, if the unit-radius circle C tangent to e_1 and e_2 does not intersect any other edge of Ω, we add four positions $(p_1, \theta_1), (p_1, \theta_1 + \pi), (p_2, \theta_2)$, and $(p_2, \theta_2 + \pi)$ to V_2, where p_i is the point at which C is tangent to e_i, and θ_i is the angle between e_i and the x-axis. Using an algorithm of Kedem et al. [27], the set of unit-radius circles tangent to two edges and not intersecting any other edge can be computed in $O(n \log^2 n)$ time, and so can be the set V_2.

The third set V_3 corresponds to semi-free C-segments and good anchored C-segments. For each edge $e \in \Omega$, we first mark a portion $\hat{e}$ of it, as described below, and then choose those points on e that are at distance $i\delta$ from its left endpoint, for any natural number i such that the semi-open interval $[i\delta, (i+1)\delta)$ intersects $\hat{e}$. Let S^e denote the set of points selected on e. Assuming that the angle between e and the x-axis is θ, for each point $p \in S^e$, we add two positions (p, θ) and $(p, \theta + \pi)$ to V_3.

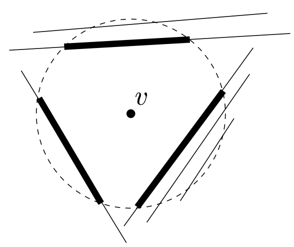

Figure 8: Segments in E_v; fat edges denote the portion of E_v visible from v.

We now describe which portion of each edge is marked. We mark the edges in two stages. First, for each edge e, we mark the portion that lies within distance 30 from any of its endpoints. Next, for each vertex $v \in \Omega$, let D_v be the disk of radius 15 centered at v. Let E_v be the set of edges that contain at least one unmarked point inside D_v, i.e., $e \in E_v$ if $e \cap D_v$ contains at least one point whose distance to both endpoints of e is more than 30. For each edge $e \in E_v$, we mark the portion of $e \cap D_v$ that is visible from v with respect to the edge set E_v (i.e., we ignore the edges in the set $E \setminus E_v$). We repeat this step for all vertices of Ω.

Lemma 3.1 *Let p be a point on an edge $e \in \Omega$ such that there is a vertex v of Ω visible from p and such that $d(p, v) \le 15$. Then there is a point $q \in S^e$ within distance δ from p.*

Proof: We only need to show that every such point p lies on $\hat{e}$ because, by construction, for every point $q \in \hat{e}$, there is a point $q' \in S^e$ such that $d(q, q') \leq \delta$. Since $d(p, v) \leq 15$ and v is visible from p, the above algorithm would have been marked p, implying that $p \in \hat{e}$. $\square$

Lemma 3.2 *Let Π be an optimal ε-robust path from I to F, and let $\langle X_1, \ldots, X_k \rangle$ be the sequence of positions on Π s.t. $\text{LOC}(X_i) \in \partial\Omega$ for every $1 \leq i \leq k$. There is a node $Y_i \in V$ such that*

(i) If $\text{LOC}(X_i)$ is a vertex v of Ω, then $\text{LOC}(Y_i) = v$ and $|\text{VEC}(Y_i) - \text{VEC}(X_i)| \leq \delta$; or

(ii) if $\text{LOC}(X_i)$ is an interior point of an obstacle edge e, then $\text{LOC}(Y_i) \in e$, $d(\text{LOC}(Y_i), \text{LOC}(X_i)) \leq \delta$, and $\text{VEC}(Y_i) = \text{VEC}(X_i)$.

Proof: If $\text{LOC}(X_i)$ is a vertex, the lemma follows from the definition of V_1. If $p = \text{LOC}(X_i)$ lies in the interior of an obstacle edge e, then p lies on a semi-free C-segment, a good anchored C-segment, or a bad anchored C-segment. In the last case, X_i is a node in V_2. In the first two cases, there is a vertex $v \in \Omega$ so that v is visible from p and that $d(p, v) \leq 15$. By Lemma 3.1 and the definition of V_3, there exists a node $Y_i \in V_3$ so that $d(\text{LOC}(Y_i), \text{LOC}(X_i)) \leq \delta$ and $\text{VEC}(Y_i) = \text{VEC}(X_i)$. This completes the proof of the lemma. $\square$

Next, we prove that the size of V is $O(n/\delta)$. Since $|V_1| = O(n/\delta)$ and $|V_2| = O(n)$, it suffices to bound the size of V_3.

Lemma 3.3 $\sum_e |S^e| = O(n/\delta)$.

Proof: For each edge $e \in \Omega$, let $\hat{e}$ denote the marked portions of e, $\#\hat{e}$ the number of connected components of $\hat{e}$, and $\|\hat{e}\|$ the total length of $\hat{e}$. For each connected component γ of $\hat{e}$, the algorithm chooses $2 + \|\gamma\|/\delta$ points. Therefore, $|S^e| \leq 2\#\hat{e} + \|\hat{e}\|/\delta$.

The total measure of points marked in the first stage is at most $60n$. Recall that for each vertex v, $e \in E_v$ if $e \cap D_v$ contains at least one point whose distance to both endpoints of e is more than 30. Thus none of the endpoints of E_v lie inside D_v, and the measure of points in $(\bigcup E_v) \cap D_v$ that are visible from v is at most 30π (the length of the perimeter of a disk of radius 15); see Figure 8. Therefore

$$\sum_{e \in \Omega} \|\hat{e}\| \leq 60n + 30\pi n = O(n).$$

Next, for each connected component $\gamma \in \hat{e}$, if γ is the first or the last connected component of $\hat{e}$, we charge γ to e itself. Otherwise, charge γ to any of the vertices that marked it in the second stage. We will prove in Lemma 3.4 below that each vertex is charged by at most 10 connected components, so

$$\sum_{e \in \Omega} \#\hat{e} = 2n + 10n = O(n).$$

This completes the proof. □

Lemma 3.4 *Each obstacle vertex is charged by at most* 10 *connected components.*

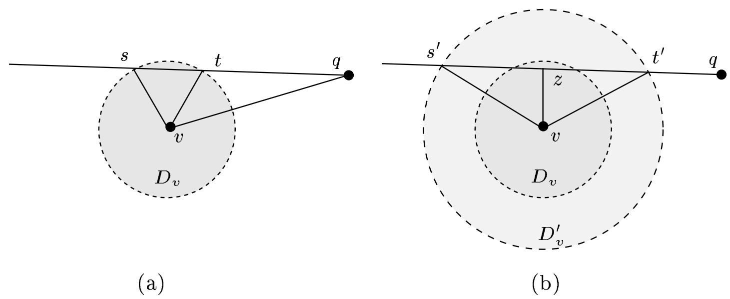

Figure 9: $d(s,t) < 15$ and $\angle s'vt' > \pi/2$.

Proof: Since the edges of E_v are disjoint and do not contain any of its endpoints inside D_v, v marks at most one connected component of each edge in E_v; see Figure 8. That is, for each edge $e \in E_v$, either v marks the entire chord $e \cap D_v$ or it does not mark any point of e. Hence, the number of connected components charged to v is bounded by the number of edges in E_v that are visible from v.

Partition the set of edges visible from v into two subsets Γ_1 and Γ_2. An edge e is in Γ_1 if the length of the chord $e \cap D_v$ is at least 15, and in Γ_2 otherwise. Each edge $e \in \Gamma_1 \cup \Gamma_2$ splits the circle ∂D_v into two circular arcs; we refer to the shorter one as the *cap* induced by e. Since the edges in $\Gamma_1 \cup \Gamma_2$ are visible from v, the caps induced by them are pairwise disjoint.

Each cap induced by an edge of Γ_1 spans an angle of $\geq \pi/3$ because the length of the chord $e \cap D_v$ is at least 15 and the radius of D_v is 15. Since the caps are disjoint, $|\Gamma_1| \leq 6$.

Next, let e be an edge of Γ_2. Let p (resp. q) denote the left (resp. right) endpoint of e, and let s (resp. t) denote the left (resp. right) endpoint of $e \cap D_v$; see Figure 9a. By definition, $d(s,t) < 15$, and therefore $\angle stv, \angle tsv > \pi/3$. Since $e \in E_v$, e contains a point u such that $d(u,p), d(u,q) > 30$. Hence, $d(s,q) = d(s,u) + d(u,q) > 30$, and

$$\begin{aligned} d(v,q) &= \sqrt{d(s,q)^2 + d(v,s)^2 - 2\,d(s,q)\,d(v,s)\cos\angle vst} \\ &> \sqrt{30^2 + 15^2 - 2 \cdot 30 \cdot 15 \cdot \cos\pi/3} \\ &= 15\sqrt{3}. \end{aligned}$$

Similarly, one can show that $d(v,p) > 15\sqrt{3}$.

Draw a disk D'_v of radius $15\sqrt{3}$ centered at v. Since $d(v,p), d(v,q) > 15\sqrt{3}$, the endpoints of every edge in Γ_2 lie outside D'_v. Moreover, each edge of Γ_2 is visible from v, the caps of $\partial D'_v$ induced by the edges of Γ_2 are also pairwise disjoint. For an edge e, let s', t' be the endpoints of $e \cap D'_v$, and z be the midpoint of the segment $s't'$; See Figure 9b. Since $d(v,z) \leq 15$ and $d(v,t') = 15\sqrt{3}$,

$$\angle s'vt' = 2\angle zvt' = 2 \cdot \cos^{-1}\frac{d(v,z)}{d(v,t')} \geq 2 \cdot \cos^{-1}\frac{1}{\sqrt{3}} > \pi/2.$$

Hence, the cap of $\partial D'_v$ induced by $e \in \Gamma_2$ spans an angle of $> \pi/2$, which implies $|\Gamma_2| \leq 4$. This completes the proof of the lemma. □

We now show that the node set V_3 can be computed in time $O(n^2 \log n + n/\delta)$. For every vertex v, the set E_v can be computed in $O(n)$ time in a straight-forward manner. Let $E'_v = \{e \cap D_v \mid e \in E_v\}$. The portions to be marked are the subset of edges of E'_v that are visible from v with respect to the edge set E'_v. (Recall that for every edge $e \in E'_v$, either every point on e is visible from v, or no point on e is visible from v.) This set can be computed in $O(n \log n)$ time by performing an angular sweep around v. Let $\rho(\theta)$ be the ray emanating from v in direction θ. Let $\theta_1, \ldots, \theta_k$ be the orientations such that $\rho(\theta)$ passes through an endpoint of an edge in E'_v. We sweep the plane with the ray $\rho(\theta)$ by varying θ from 0 to 2π. For each θ, we maintain the edges of E'_v intersecting $\rho(\theta)$, sorted in the order they intersect $\rho(\theta)$. Since the segments of E'_v are pairwise disjoint, the ordering changes only at θ_i's. Let $e_i \in E'_v$ be the first edge in this ordering in the interval $[\theta_i, \theta_{i+1})$. We mark e_i. At each θ_i, we can update the ordering in time $O(\log n)$. Repeating this process for all the vertices of Ω, V_3 can be computed in time $O(n^2 \log n + n/\delta)$. Hence, we conclude the following

Lemma 3.5 *The node set V can be computed in time $O(n^2 \log n + n/\delta)$.*

3.2 Computing the edge set

We now describe how to compute the edge set E. For each pair of positions $X, Y \in V$, we compute all $O(1)$ Dubins paths from X to Y, check which of them are feasible, and select the one with the minimum arc length. The only nontrivial step is to determine whether a given Dubins path is feasible. We will consider CCC and CLC paths separately.

Testing CCC paths. CCC paths can be further classified into two subcategories, as follows. If we label a clockwise (resp. counterclockwise) oriented C-segment as C^+ (resp. C^-), then a CCC path is either $C^+C^-C^+$ type or $C^-C^+C^-$ type. We consider only $C^+C^-C^+$ paths; $C^-C^+C^-$ paths can be handled in a similar manner.

For each position X, let C_X denote the clockwise oriented circle passing through X, and let ϕ^+_X (resp. ϕ^-_X) be the intersection point of C_X and $\partial\Omega$ immediately after (resp. before) $\text{LOC}(X)$, so the interiors of the arcs $C_X[\text{LOC}(X), \phi^+_X]$ and $C_X[\phi^-_X, \text{LOC}(X)]$ do not intersect $\partial\Omega$ (see Figure 10a); ϕ^+_X and ϕ^-_X can be computed in $O(n)$ time.

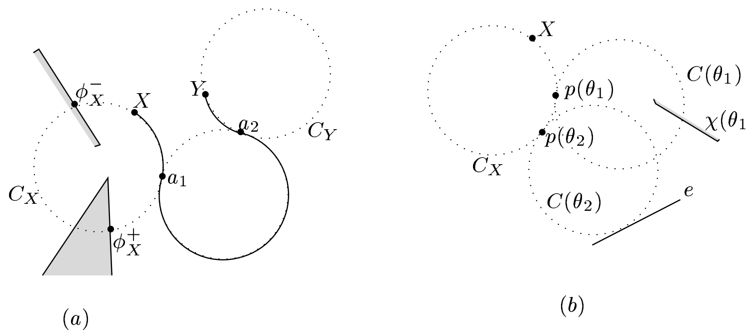

Figure 10: (a) $C^+C^-C^+$ path; (b) $p(\theta), C(\theta)$, and a critical orientation θ_2.

Let $w_1w_2w_3$ be a $C^+C^-C^+$ path from a position X to another position Y, with a_i being the common endpoint of w_i and w_{i+1}, for $i = 1, 2$. Obviously w_1 (resp. w_3) does not intersect Ω if and only if $a_1 \in C_X[\text{LOC}(X), \phi_X^+]$ (resp. $a_2 \in C_Y[\phi_Y^-, \text{LOC}(Y)]$); see Figure 10a. After computing ϕ_X^+ and ϕ_X^- for all $O(n/\delta)$ positions in time $O(n^2/\delta)$, given a $C^+C^-C^+$ path, we can check in $O(1)$ time whether its first and last C-segments intersect Ω. Next, we describe how to test whether the middle C-segment of a $C^+C^-C^+$ path intersects Ω.

Fix a position X. We construct a linear-size data structure, in $O(n \log n)$ time, which, given a position Y, can determine in $O(\log n)$ time whether the middle C-segment of the $C^+C^-C^+$ path from X to Y intersects any obstacle edge.

For an orientation θ, $0 \leq \theta < 2\pi$, let $p(\theta)$ be the point on C_X such that the arc length of $C_X[\text{LOC}(X), p(\theta)]$ is θ. Let $C(\theta)$ be the counterclockwise-directed circle tangent to C_X at $p(\theta)$. Set $\chi(\theta)$ to be the first edge of Ω intersected by $C(\theta)$, as one walks along it (in the counterclockwise direction) starting from $p(\theta)$. If there is no such edge, then $\chi(\theta)$ is undefined (see Figure 10b).

We call an orientation θ *critical* if $C(\theta)$ is either tangent to an obstacle edge or it passes through an obstacle vertex. Let $\langle \theta_1, \theta_2, \ldots, \theta_k \rangle$ be the sequence of critical angles sorted in the increasing order. Using a simple continuity argument, we can prove the following.

Lemma 3.6 *For all orientations θ in any interval $[\theta_i, \theta_{i+1})$, the set of obstacle edges that intersect $C(\theta)$, and the order in which they intersect $C(\theta)$, remains the same.*

This lemma implies that the value of $\chi(\theta)$ remains the same for all orientations within each interval $[\theta_i, \theta_{i+1})$.

Lemma 3.7 *There are $O(n)$ critical orientations.*

Proof: For an edge $e \in \Omega$, let D_e be the Minkowski sum of e and the unit-radius disk. Let C'_X be the circle concentric with C_X and of radius 2. $C(\theta)$ is tangent to e, or passes through an endpoint of e, if and only if $C(\theta)$ is centered at an intersection point of C'_X and

∂D_e. There are at most 4 intersection points between C'_X and ∂D_e. Thus an edge e can contribute at most $O(1)$ critical orientations, resulting in $O(n)$ critical orientations for all edges. □

These $O(n)$ critical orientations can be computed in $O(n)$ time. By sweeping the circle $C(\theta)$ for $\theta \in [0, 2\pi)$, we can compute the values of χ for all $O(n)$ intervals $[\theta_i, \theta_{i+1})$ in $O(n \log n)$ time, as follows. For each θ, we maintain the set of edges intersecting $C(\theta)$, sorted in the order in which they intersect $C(\theta)$. By Lemma 3.6, this ordering changes only at critical orientations. At each critical orientation, we can update the ordering in $O(\log n)$ time, thus, spending a total of $O(n \log n)$ time. We record the values of χ for each interval $[\theta_i, \theta_{i+1})$.

Now, given a $C^+C^-C^+$ Dubins path $w_1w_2w_3$, we first compute the orientation θ of a_1, the common endpoint of w_1 and w_2. Using the above data structure, we can determine $e = \chi(\theta)$ in $O(\log n)$ time by a binary search. Finally, we check in $O(1)$ time whether w_2 intersects the edge e (That is, we check whether a_2, the common endpoint of w_2 and w_3, lies before the intersection point of $C(\theta)$ and e, in which case w_2 does not intersect any edge of Ω). This completes the description of the data structure.

We can thus conclude that for all $X, Y \in V$, we can determine in time $O((n^2/\delta^2) \log n)$ whether there is a feasible CCC-path from X to Y.

Testing CLC paths. There are four types of CLC paths, namely C^+LC^+, C^+LC^-, C^-LC^+ and C^-LC^-. Consider C^+LC^+ paths. Let $w_1w_2w_3$ be a C^+LC^+ path. After $O(n^2/\delta)$ preprocessing as above, we can easily determine whether w_1 or w_3 intersects Ω. As for w_2, we construct a similar data structure. Fix a position X. For a given θ, we now define $\ell(\theta)$ to be the ray tangent to C_X and emanating from $p(\theta)$, and define $\chi(\theta)$ to be the first edge of Ω intersected by $\ell(\theta)$. An orientation θ is *critical* if $\ell(\theta)$ passes through a vertex of Ω. We can again construct a linear-size data structure in $O(n \log n)$ time that given a position Y, can determine in $O(\log n)$ time whether the line segment of C^+LC^+ path from X to Y intersects Ω.

Hence, we can compute in $O((n^2/\delta^2) \log n)$ time all pairs $X, Y \in V$ that admit a feasible C^+LC^+ path from X to Y. Repeating this procedure for other types of CLC paths, we can compute in $O((n^2/\delta^2) \log n)$ time all the pairs $X, Y \in V$ for which there is a feasible CLC path from X to Y.

After having computed the vertices and edges of G, we can compute a shortest path in G, using any standard shortest-path algorithm [18]. Putting everything together, we obtain the following result.

Theorem 3.8 *The graph G, as described above, can be constructed in time $O((n^2/\delta^2) \log n)$, and a shortest path from I to F in G can be computed in an additional $O(n^2/\delta^2)$ time.*

4 Error Analysis

In this section we prove that if δ is chosen to be at most $c\varepsilon^2$, where c is a sufficiently small constant independent of ε, then the above algorithm computes an $(1+\varepsilon)$-approximate shortest path from I to F. We first bound the change in the length of a Dubins path as we perturb its end-positions, and then we bound the length of the path computed by the above algorithm.

4.1 Error induced by a single Dubins path

To give an error bound for our approximation algorithm, we need to answer the following question: given two Dubins paths of the same type whose end-positions differ by a small amount, what is the difference in length of these two paths? Let X and X' be two positions. If $\text{LOC}(X') = \text{LOC}(X)$, we define $\Delta(X', X)$ to be $|\text{VEC}(X') - \text{VEC}(X)|$; and if $\text{VEC}(X') = \text{VEC}(X)$, then we define $\Delta(X', X)$ to be $\|\text{LOC}(X') - \text{LOC}(X)\|$. For two paths Π and Π', let $\Delta(\Pi', \Pi)$ be the difference in length of these two paths, i.e., $\Delta(\Pi', \Pi) = \Big| \|\Pi'\| - \|\Pi\| \Big|$. Two Dubins paths Π_{XY} and $\Pi_{X'Y'}$ of the same type from X to Y and from X' to Y', respectively, will be called *homotopic* if Π_{XY} can be continuously deformed to $\Pi_{X'Y'}$ in such a way that every intermediate path is also a Dubins path of the same type.

Lemma 4.1 *Let Π be a CLC path from a position I to a position F. Let I' and F' be positions such that $\Delta(I', I) = \delta_I$ and $\Delta(F', F) = \delta_F$, for any reals $\delta_I, \delta_F \geq 0$. Let Π' be the path from I' to F' of the same type as Π and homotopic to Π. Then*

$$\Delta(\Pi', \Pi) = O(\delta_I + \delta_F).$$

Proof: We will prove that if $\delta_F = 0$ (i.e., $F' = F$), then $\Delta(\Pi', \Pi) = O(\delta_I)$. By reversing the direction of Π, this also implies that $\Delta(\Pi', \Pi) = O(\delta_F)$ if $\delta_I = 0$. If both $\delta_I, \delta_F > 0$, then let Π'' be the CLC path from I' to F of the same type as Π. Then

$$\Delta(\Pi', \Pi) \leq \Delta(\Pi'', \Pi) + \Delta(\Pi', \Pi'') = O(\delta_I + \delta_F),$$

as claimed. We now assume $F' = F$, and set $\delta = \delta_I$.

We will prove the claim for the case in which I is located at an obstacle vertex u, i.e., $I = (u, \theta_I)$ and $|\text{VEC}(I') - \text{VEC}(I)| = \delta$. Let C_1 (resp. C_2) be the unit-radius circle containing the initial (resp. final) C-segment of Π, and let o_i (for $i = 1, 2$) denote the centers of C_i. Let C_1' be the circle containing the initial C-segment of Π', and let o_1' be the center of C_1'. If Π is a C^+LC^+ or C^-LC^- type path, then

$$\|\Pi\| = |\text{VEC}(I) - \text{VEC}(F)| + d(o_1, o_2),$$

in which case,

$$\Delta(\Pi, \Pi) \leq |\text{VEC}(I') - \text{VEC}(I)| + |d(o_1, o_2) - d(o_1', o_2)| \leq \delta + d(o_1, o_1').$$

By applying the cosine law to $\triangle o_1uo_1'$ (see Figure 11(b)),

$$d(o_1, o_1') = \sqrt{1+1-2\cos\delta} = 2\sin(\delta/2) \le \delta. \qquad (1)$$

Therefore, $\Delta(\Pi', \Pi) \le 2\delta$.

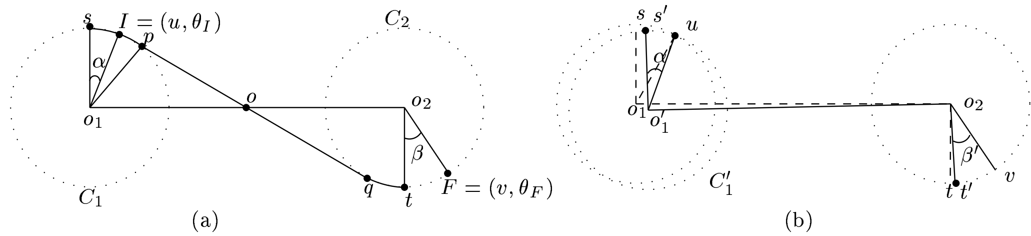

Figure 11: Bounding the difference in path length for CLC paths.

In the following we prove the lemma for the case when Π is a C^+LC^- type path; the other case, when Π is a C^-LC^+ type path, is symmetric. Let p (resp. q) be the initial (resp. final) point of the L-segment of Π, i.e., it is the point at which the common tangent of C_1^+ and C_2^- touches C_1 (resp. C_2). Let s be the point on C_1 such that $so_1 \perp o_1o_2$ and $\angle so_1p < \pi/2$, and let S be the position on C_1 corresponding to s, i.e., $\textsc{loc}(S) = s$ and $\textsc{vec}(S)$ is the orientation of the tangent to C_1 (assuming that C_1 is clockwise oriented) at s. We define a similar point t and position T on C_2; see Figure 11(a). Instead of considering Π, we consider the C^+LC^- path from S to T; and let ρ be the length of this path. If we measure the angles in counterclockwise direction and choose their values between $-\pi$ and π, then

$$\|\Pi\| = \rho + \alpha + \beta$$

(see Figure 11(a)), where $\alpha = \angle so_1u$ and $\beta = \angle to_2v$. We define ρ', α' and β' corresponding to Π'. Then $\|\Pi'\| = \rho' + \alpha' + \beta'$, and

$$\Delta(\Pi', \Pi) \le |\rho' - \rho| + |\alpha' - \alpha| + |\beta' - \beta|. \qquad (2)$$

We first bound $|\alpha' - \alpha|$. Applying the sine law to $\triangle o_1o_2o_1'$,

$$\frac{\sin\angle o_1o_2o_1'}{d(o_1, o_1')} = \frac{\sin\angle o_1'o_1o_2}{d(o_1, o_2)}.$$

Using (1), the fact that $d(o_1, o_2) \ge 2$, and the inequality $\sin^{-1} x \le 2x$ for any $0 \le x \le 1$, we obtain

$$\angle o_1o_2o_1' \le \sin^{-1}\frac{\delta}{2} \le \delta.$$

Let w be the intersection point of lines supporting the segments so_1 and $s'o_1'$. Since $so_1 \perp o_1o_2$ and $s'o_1' \perp o_1'o_2$, $\angle o_1wo_1' = \angle o_1o_2o_1' \le \delta$; see Figure 12. Moreover, $\alpha' + \angle o_1uo_1' = \alpha + \angle o_1wo_1'$, therefore

$$|\alpha' - \alpha| \le \angle o_1uo_1' + \angle o_1wo_1' = O(\delta). \qquad (3)$$

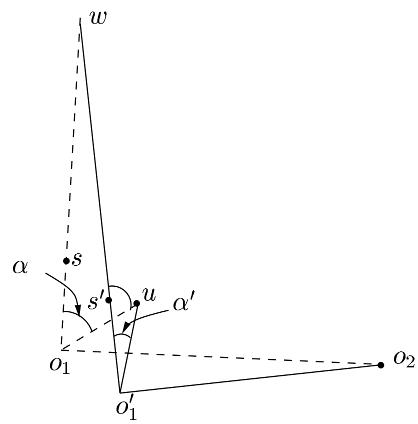

Figure 12: Bounding $|\alpha' - \alpha|$.

Similarly, we can show that $|\beta' - \beta| = O(\delta)$.

Next, we bound $|\rho' - \rho|$. Notice that

$$\rho = 2\left(d(o,p) + \|C_1[s,p]\|\right).$$

Let $\mu = d(o_1, o) = d(o_1, o_2)/2$, then $\angle so_1p = \angle poo_1 = \sin^{-1}(1/\mu)$, and we have

$$\rho = 2\left(\sin^{-1}\left(\frac{1}{\mu}\right) + \sqrt{\mu^2 - 1}\right).$$

Similarly, if we let $\nu = d(o_1', o_2)/2$, then

$$\rho' = 2\left(\sin^{-1}\left(\frac{1}{\nu}\right) + \sqrt{\nu^2 - 1}\right).$$

As in (1), $d(o_1', o_1) \leq \delta$, which implies that $\nu - \mu \leq \delta/2 \leq \delta$. Consider the function

$$f(x) = \sin^{-1}\left(\frac{1}{x}\right) + \sqrt{x^2 - 1}.$$

Then $|\rho' - \rho| = 2|f(\nu) - f(\mu)|$. Since both $f(x)$ and it derivative are monotonically increasing,

$$|\rho' - \rho| \leq 2(f(\mu + \delta) - f(\mu))$$

and $|\rho' - \rho|$ maximizes as μ tends to infinity. Using Taylor expansion, one can write

$$f(x) = x + \sum_{i=1}^{\infty} \frac{c_i}{x^{2i-1}},$$

where $|c_i|$'s are monotonically decreasing with i. Thus

$$|\rho' - \rho| \leq \lim_{\mu \to \infty} 2\left(\mu + \delta + \sum_{i=1}^{\infty} \frac{c_i}{(\mu + \delta)^{2i-1}} - \left(\mu + \sum_{i=1}^{\infty} \frac{c_i}{\mu^{2i-1}}\right)\right) = O(\delta). \quad (4)$$

Combining together (3) and (4), we obtain that

$$\Delta(\Pi', \Pi) = O(\delta).$$

This completes the proof of the lemma. □

Lemma 4.2 *Let Π be a CCC path from a position I to a position F. Let I' and F' be positions such that $\Delta(I', I) = \delta_I$ and $\Delta(F', F) = \delta_F$, for any reals $\delta_I, \delta_F \geq 0$. Let Π' be the Dubins path from I' to F' of the same type as Π and homotopic to Π. Then*

$$\Delta(\Pi', \Pi) = O(\sqrt{\delta_I} + \sqrt{\delta_F}).$$

Proof: As in the proof of Lemma 4.1, we only need to prove the lemma for the case when $\delta_I > 0$ and $\delta_F = 0$. Let $\delta = \delta_I$. We assume that I is located at an obstacle vertex, thus $\textsc{loc}(I') = \textsc{loc}(I)$ and $|\textsc{vec}(I') - \textsc{vec}(I)| = \delta$. We prove the lemma for the case when Π is a $C^-C^+C^-$ path; the other case, when Π is a $C^+C^-C^+$ path, is symmetric.

Let p (resp. q) be the initial (resp. final) location of the path Π. Let o_i be the center of the unit circle containing the ith C-segment of Π. Consider the triangle $\Delta o_1 o_2 o_3$; see Figure 13(a). Without loss of generality, assume that p lies inside $\triangle o_1 o_2 o_3$ and q does not; other cases can be handled similarly. Let b_i be the angle of the triangle adjacent to o_i.

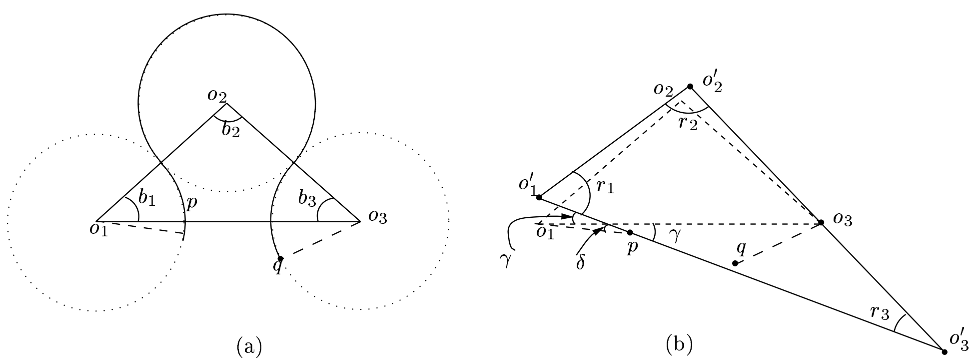

Figure 13: Bounding the difference in path length for CCC paths.

Then

$$\begin{aligned}
\|\Pi\| &= b_1 + \angle p o_1 o_3 + 2\pi - b_2 + b_3 + \angle o_1 o_3 q \\
&= (b_1 + b_3) + 2\pi - b_2 + \angle p o_1 o_3 + \angle o_1 o_3 q \\
&= \pi - b_2 + 2\pi - b_2 + \angle p o_1 o_3 + \angle o_1 o_3 q \\
&= 3\pi - 2b_2 + \angle p o_1 o_3 + \angle o_1 o_3 q.
\end{aligned}$$

Let o_1', o_2' and o_3 be the centers of circles containing the C-segments of Π'. Notice that $\angle o_1' p o_1 = \delta$; see Figure 13(b). Consider the triangle $\Delta o_1' o_2' o_3'$, where o_3' is the intersection point of the line supporting the segment $o'_1 p$ and the line supporting the segment $o'_2 o_3$; assume that o'_1 is situated so that o_3 lies between o_2 and o'_3. Let r_i be the angle of the triangle adjacent to o'_i. Then

$$\begin{aligned}
\|\Pi'\| &= r_1 + 2\pi - r_2 + \angle o'_2 o_3 o_1 + \angle o_1 o_3 q \\
&= r_1 + 2\pi - r_2 + r_3 + \gamma + \angle o_1 o_3 q \\
&= r_1 + 2\pi - r_2 + r_3 + \angle o'_1 p o_1 + \angle p o_1 o_3 + \angle o_1 o_3 q \\
&= (r_1 + r_3) + 2\pi - r_2 + \delta + \angle p o_1 o_3 + \angle o_1 o_3 q \\
&= 3\pi - 2r_2 + \delta + \angle p o_1 o_3 + \angle o_1 o_3 q.
\end{aligned}$$

Thus

$$\Delta(\Pi', \Pi) = |\delta + 2b_2 - 2r_2| \le \delta + 2|b_2 - r_2|.$$

Let $\rho = d(o_1, o_3)$ and $\rho' = d(o'_1, o_3)$. Applying the cosine law to $\Delta o_1 o_2 o_3$ and $\Delta o'_1 o'_2 o_3$, and using the fact that $d(o_1, o_2) = d(o_2, o_3) = d(o'_1, o'_2) = d(o'_2, o_3) = 2$, we obtain

$$\begin{aligned}
b_2 &= \cos^{-1}\left(\frac{2^2 + 2^2 - \rho^2}{2 \cdot 2 \cdot 2}\right) = \cos^{-1}\left(1 - \frac{\rho^2}{8}\right), \text{and} \\
r_2 &= \cos^{-1}\left(\frac{2^2 + 2^2 - \rho'^2}{2 \cdot 2 \cdot 2}\right) = \cos^{-1}\left(1 - \frac{\rho'^2}{8}\right).
\end{aligned}$$

Define a function

$$f(x) = \cos^{-1}\left(1 - \frac{x^2}{8}\right).$$

Then $|b_2 - r_2| = |f(\rho) - f(\rho')|$. Since $f(x)$ is monotonically increasing, $|b_2 - r_2| \le f(\rho) - f(\rho - \delta)$. Moreover, $\frac{df}{dx} = 8 \,/\, \sqrt{1 - \frac{x^2}{16}}$ is also monotonically increasing and $\rho \le 4$, therefore $|b_2 - r_2|$ maximizes at $\rho = 4$. Thus

$$\begin{aligned}
|b_2 - r_2| &\le \cos^{-1}\left(1 - \frac{4^2}{8}\right) - \cos^{-1}\left(1 - \frac{(4-\delta)^2}{8}\right) \\
&\le \pi - \cos^{-1}(-1 + \delta) \\
&= \pi - \left(\pi - 2\sqrt{\sin^{-1}\frac{\delta}{2}}\right) \\
&= 2\sqrt{\sin^{-1}\frac{\delta}{2}} \quad \le 2\sqrt{\delta},
\end{aligned}$$

as desired. □

Remark 4.3 Notice that $\Delta(\Pi, \Pi') = O(\sqrt{\delta})$ only if the distance between the centers o_1 and o_3 is almost 4. If $d(o_1, o_3) \le 4 - c$ for some constant $c > 0$, then $\Delta(\Pi, \Pi') \approx \delta/\sqrt{c}$.

4.2 Goodness of our approximation

Lemma 4.4 *If $\delta = c\varepsilon^2$, where c is a sufficiently small constant , then there exists a path from I to F in the graph G computed in Section 3 whose length is at most $(1+\varepsilon)$ times the length of an optimal ε-robust path from I to F.*

Proof: Let $\Pi = \Pi_1 \| \ldots \| \Pi_k$ be an optimal ε-robust path from I to F, where each Π_i is a Dubins path from a position X_{i-1} to a position X_i, such that $X_0 = I$, $X_k = F$ and $\textsc{loc}(X_i) \in \partial\Omega$, for $0 < i < k$. By Lemma 3.2, there exist graph nodes $Y_0 = I, Y_1, \ldots, Y_{k-1}, Y_k = F$, such that $\Delta(Y_i, X_i) \le \delta$, such that if $\textsc{loc}(X_i)$ is a vertex of Ω then $\textsc{loc}(Y_i) = \textsc{loc}(X_i)$, and such that if $\textsc{loc}(X_i)$ is an interior point of an edge of Ω then $\textsc{vec}(Y_i) = \textsc{vec}(X_i)$. Since each Π_i is an ε-robust path and we are going to choose $\delta = c\varepsilon^2 < \varepsilon$, there is a feasible Dubins path Π'_i of the same type as Π_i from Y_i to Y_{i+1}, for $0 \le i \le k-1$. Therefore (Y_i, Y_{i+1}) is an edge in G, and $\Pi' = \Pi'_1 \| \ldots \| \Pi'_k$ is the desired path from I to F in G. To prove the lemma, it suffices to show that $\|\Pi'_i\| \le (1+\varepsilon)\|\Pi_i\|$, for $1 \le i \le k$, provided we choose $\delta = c\varepsilon^2$ small enough.

If Π_i is a CLC path, by Lemma 4.1, $\Delta(\Pi'_i, \Pi_i) \le O(\delta)$. Since Π_i is an ε-robust path, its length is at least ε. Therefore,

$$\|\Pi'_i\| \le \|\Pi_i\| + O(\delta) \le \|\Pi_i\| + \varepsilon^2 \le (1+\varepsilon)\|\Pi_i\|,$$

provided the constant c is chosen sufficiently small. If Π_i is a CCC path, by Lemma 4.2, $\Delta(\Pi'_i, \Pi_i) \le O(\sqrt{\delta})$. But the length of a CCC path is at least π, therefore

$$\|\Pi'_i\| \le \|\Pi_i\| + O(\sqrt{\delta}) \le \|\Pi_i\| + \varepsilon \le \left(1 + \frac{\varepsilon}{\pi}\right) \|\Pi_i\| \le (1+\varepsilon)\|\Pi_i\|,$$

provided c is chosen small enough. This completes the proof of the lemma. □

Plugging $\delta = O(\varepsilon^2)$ in Theorem 3.8, we obtain the following result.

Theorem 4.5 *Given a polygonal obstacle environment Ω, an initial position I, a final position F, and a parameter ε, we can compute in time $O((n^2/\varepsilon^4)\log n)$ a feasible path from I to F whose arc length is at most $(1+\varepsilon)$ times the length of an optimal ε-robust path.*

Remark 4.6 Recall that the running time of the algorithm is $O((n^2/\varepsilon^4)\log n)$ because we choose $\delta = \varepsilon^2$, and the graph G has $O((n/\delta))$ vertices and in the worst-case every pair of vertices is connected by an edge. If the distance between the centers of initial and final circles of CCC-type paths for most pairs of vertices is not close to 4, one can show that it suffices to add edges between $O((n^2/\delta)\log(1/\delta))$ pairs of vertices, and that these pairs can be computed in time $O((n^2/\delta^2)\log(1/\delta))$. In this case the time complexity improves to $O((n^2/\varepsilon^2)\log n \log(1/\varepsilon))$.

5 Computing Near Optimal Robust Paths

The path computed by the above algorithm is not necessarily robust because some of the edges in G may not correspond to robust paths. We can compute a graph $G' = (V, E')$, where E' is the set of edges corresponding to $(\varepsilon/2)$-robust paths. An easy argument shows that if δ is chosen correctly, there is an $(\varepsilon/2)$-robust path in G whose length is at most $(1+\varepsilon)$ times the length of an optimal ε-robust path from I to F.

Next we show that E' can be computed in $O((n^{2.5}/\varepsilon^4)\log n)$ time. For each pair of positions $X, Y \in V$, we compute all $O(1)$ Dubins paths from X to Y, check which of them are $(\varepsilon/2)$-robust, and select the one with the minimum arc length. The only nontrivial step is to determine whether a given Dubins path is $(\varepsilon/2)$-robust.

Let Π_{XY} be a Dubins path from X to Y. We can assume that $\text{LOC}(X)$ and $\text{LOC}(Y)$ are obstacle vertices. The other cases when $\text{LOC}(X)$ or $\text{LOC}(Y)$ lie in the interior of obstacle edges can be dealt in a similar manner. Let γ_{XY} be the region formed by the set of Dubins paths $\Pi_{X'Y'}$ such that $\Delta(X, X'), \Delta(Y, Y') \le \varepsilon/4$ and $\Pi_{X'Y'}$ can be obtained by deforming Π_{XY} continuously. The boundary of γ_{XY} consists of $O(1)$ x-monotone algebraic arcs, each of $O(1)$ degree, and they can be computed in $O(1)$ time.[1] Π_{XY} is $(\varepsilon/2)$-robust if and only if γ_{XY} does not intersect the interior of Ω. Since $\text{LOC}(X)$ and $\text{LOC}(Y)$ lie on the boundary of obstacles, γ_{XY} intersects the interior of any obstacle if and only if any of the obstacle edges intersect the interior of γ_{XY}. We thus have the following intersection-detection problem at hand: Let $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ be a set of m regions, each of whose boundary consists of $O(1)$ algebraic arcs of constant degrees, and let S be a set of n disjoint line segments in the plane. Report all regions in Γ whose interiors do not intersect any segment of S. We present an $O((m\sqrt{n} + n)\log n)$-time algorithm to report such a subset. Since $m = O(n^2/\varepsilon^4)$ in our case, we conclude that we can computer all $(\varepsilon/2)$-robust paths in time $O((n^{2.5}/\varepsilon^4)\log n)$.

We now describe an algorithm for the intersection-detection problem just described. It suffices to describe an $O(n \log n)$-time algorithm for the case when $m = \sqrt{n}$, for otherwise we can partition Γ into $\lceil m/\sqrt{n}\rceil$ subsets, $\Gamma_1, \ldots, \Gamma_s$, each of size at most $\sqrt{n}$, and solve the intersection-detection problem for each Γ_i and S separately. The total running time is obviously $O((m\sqrt{n} + n)\log n)$.

Let E be the set of x-monotone arcs bounding the regions in Γ. A segment $e \in S$ intersects the interior of a region $\gamma \in \Gamma$ if at least one of the following two conditions is satisfied: (i) An endpoint of e lies in the interior of γ, or (ii) e intersects the boundary of γ. It is possible to check both of these conditions for all regions in Γ in $O(n \log n)$ time, using a single sweep-line algorithm. But for the sake of clarity, we explain how to check each of the two conditions separately. The first condition can be checked in $O(n \log n)$ time by a variant of the batched point-location algorithm by Preparata [40]. It basically sweeps a vertical line from left to right and maintains the subset of regions that intersect the

[1] Note that there exist X' and Y' such that no $\Pi_{X'Y'}$ can be obtained by a continuous deformation of Π_{XY}. In this case, we consider Π_{XY} non-robust even if there is a free Dubins path from X' to Y'. These cases can be detected in $O(1)$ time.

sweep-line. Whenever the sweep-line encounters an endpoint p of S, it reports and deletes all the regions of Γ whose interior contains p. These steps can be implemented efficiently using interval trees or segment trees. We omit the rather easy and standard details from here. The total running time of the algorithm is $O((m^2+n+k)\log(m+n))$, where k is the number of regions in Γ that contain an endpoint of S. Since $m=\sqrt{n}$, and each region of Γ is a semi-algebraic region of constant description complexity, $k=O(m)=O(\sqrt{n})$. Hence, the total time spent is $O(n\log n)$.

Next, we explain how to detect condition (ii). This can be done by modifying the Bentley-Ottman [4] algorithm for segment-intersection reporting, as follows. We sweep a vertical line from left to right and store the the arcs of $E\cup S$ intersecting the sweep-line in a height balanced tree T, sorted in y-direction. The algorithm maintains the invariant that none of the arcs of E intersecting the sweep-line intersects any segment of S. A region γ is deleted from Γ as soon as we detect an intersection between $\partial\gamma$ and S; all the boundary arcs of γ are deleted from E as well. The sweep-line stops at the endpoints of $E\cup S$ and the intersection points of arcs in E. At the left (resp. right) endpoint of an arc $e\in E$, we insert e into T (resp. delete e from T). We do the same at the endpoints of S. At an intersection point σ of two arcs $\rho_1,\rho_2\in E$, we swap the order of ρ_1 and ρ_2 in T. Whenever one of the adjacent element of an active arc $\rho\in E$ changes (because of insertion, deletion, or swapping of two arcs), we check whether the new adjacent element is a segment $e\in S$. If e and ρ intersect, we delete ρ from E and T. Let γ be the region bounded by ρ. We report and delete γ from Γ, and delete all the edges bounding γ. Note that deletion of these arcs from T may change the adjacent elements of other arcs stored in T, so we have to check for their intersections, but this time can be charged to the arcs deleted. Since each arc of Γ is deleted only once, and there are $m=\sqrt{n}$ arcs, the total time spent in this step is $O(\sqrt{n}\log(m+n))$. The sweep-line stops in at most $O(m^2+n)$ points, hence the overall time spent is also $O(n\log n)$. Putting all the steps together, we obtain

Theorem 5.1 *Given a polygonal obstacle environment Ω, an initial position I, a final position F, and a parameter ε, we can compute in time $O((n^{2.5}/\varepsilon^4)\log n)$ a feasible $(\varepsilon/2)$-robust path from I to F whose arc length is at most $(1+\varepsilon)$ times the length of an optimal ε-robust path.*

6 Conclusion

In this paper we presented an efficient and simple approximation algorithm for computing a curvature-constrained shortest path. The main ingredients of our algorithm are a stronger characterization of curvature-constrained shortest paths, by exploiting their geometry, and a fast and simple algorithm for constructing the graph. We conclude this paper by suggesting a few open problems.

(i) Can one improve the running time of our algorithm to almost linear? Since we are interested only in computing approximate shortest paths, it may be sufficient to construct a small subset of the edges of G. Such techniques have been used to compute approximate unconstrained shortest paths amid obstacles, e.g., [15].

(ii) How fast can one compute an approximate curvature-constrained shortest path in 3-space, especially in view of the recent result by Sussmann [47]?

(iii) Can one develop simple and efficient algorithms for more general constraints, exploiting the geometry of paths?

Acknowledgements The authors thank John Reif for several valuable discussions.

References

[1] P. K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[2] P.K. Agarwal, P. Raghavan, and H. Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *Proc. 27th ACM Symp. Theory of Computing*, pages 343–352, 1995.

[3] J. Barraquand and J.C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.

[4] J. Bentley and T. Ottmann. Algorithms for reporting and counting intersections. *IEEE Trans. Comput.*, 28:643–647, 1979.

[5] J.D. Boissonnat and X.N. Bui. Accessibility region for a car that only moves forwards along optimal paths. Tech. Rept. 2181, INRIA, Sophia Antipolis, France, 1994.

[6] J.D. Boissonnat, A. Cerezo, and J. Leblond. Shortest paths of bounded curvature in the plane. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2315–2320, 1992.

[7] J.D. Boissonnat, A. Cerezo, and J. Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Tech. Rept. 2160, INRIA, Sophia Antipolis, France, 1994.

[8] J.D. Boissonnat and S. Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles. In *Proc. 12th Symp. on Computational Geometry*, pages 242–251, 1996.

[9] X.N. Bui, J.D. Boissonnat, P. Soueres, and J.P. Laumond. Shortest path synthesis for Dubins non-holonomic robot. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2–7, 1994.

[10] J. Canny. Some algebraic and geometric configurations in PSPACE. In *Proc. 20th ACM Symp. Theory of Computing.*, pages 460–467, 1988.

[11] J. Canny. Private communication.

[12] J. Canny, B. Donald, J. Reif, and P. Xavier. On the complexity of kinodynamic planning. In *Proc. 29th Symp. Foundations of Computer Science*, pages 306–316, 1988.

[13] J. Canny, A. Rege, and J. Reif. An exact algorithm for kinodynamic planning in the plane. *Discrete Comput. Geom.*, 6:461–484, 1991.

[14] J. Canny and J. Reif. New lower bound techniques for robot motion planning. In *Proc. 28th Symp. Foundations of Computer Science*, pages 49–60, 1987.

[15] K.L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 9th Symp. Theory of Computing*, pages 56–65, 1987.

[16] E. Cockayne and G. Hall. Plane motion of a particle subject to curvature constraints. *SIAM J. Control*, 43:197–220, 1975.

[17] R. Cole and M. Sharir. Visibility problems for polyhedral terrains. *J. Symbolic Comput.*, 7:11–30, 1989.

[18] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. The MIT Press, MA, 1991.

[19] B. Donald and P. Xavier. Near-optimal kinodynamic planning for robots with coupled dynamics bounds. In *IEEE Int. Symp. Intelligent Controls*, pages 354–359, 1989.

[20] L.E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 79:497–516, 1957.

[21] S. Fortune and G. Wilfong. Planning constrained motion. *Annals of Mathematics and Artificial Intelligence*, 3:21–82, 1991.

[22] T. Fraichard. Smooth trajectory planning for a car in a structured world. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2–7, 1991.

[23] L. Guibas, M. Overmars, and M. Sharir. Ray shooting, implicit point location, and related queries in arrangements of segments. Tech. Rept. 433, Dept. Computer Science, New York University, New York, 1989.

[24] J. Hopcroft, J. Schwartz, and M. Sharir, editors. *Planning, Geometry, and Complexity of Robot Motion*. Ablex Publishing Corp., Norwood, NJ, 1984.

[25] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Nonholonomic Motion Planning* (Z. Li and J.F. Canny, editors), Kluwer Academic, Norwell, MA, pages 271–342, 1992.

[26] P. Jacobs, J.P. Laumond, and M. Taix. Efficient motion planners for nonholonomic mobile robots. In *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 1229–1235, 1991.

[27] 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.

[28] V.P. Kostov and E.V. Degtiariova-Kostova. Suboptimal paths in the problem of a planar motion with bounded derivative of the curvature. Tech. Rept., INRIA, Cedex, France, 1993.

[29] V.P. Kostov and E.V. Degtiariova-Kostova. The planar motion with bounded derivative of the curvature and its suboptimal paths. Tech. Rept., INRIA, Cedex, France, 1994.

[30] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, 1991.

[31] J.C. Latombe. A fast path-planner for a car-like indoor mobile robot. In *Proc. 9th National Conf. on Artificial Intelligence*, pages 659–665, 1991.

[32] J.P. Laumond. Finding collision free smooth trajectories for a nonholonomic mobile robot. In *Proc. 10th Int. Joint Conf. on Artificial Intelligence*, pages 1120–1123, 1987.

[33] J.P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation*, 10:577–593, 1994.

[34] J.P. Laumond and T. Simeon. Motion planning for a two degrees of freedom mobile robot with towing. Tech. Rept., LAAS/CNRS, LAAS, Toulouse France, 1989.

[35] J.P. Laumond, M. Taix, and P. Jacobs. A motion planner for car-like robots based on a global/local approach. In *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 765–773, 1990.

[36] Z. Li and J.F. Canny, editors. *Nonholonomic Motion Planning*. Kluwer Academic Publishers, Norwell, MA, 1992.

[37] B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proc. IEEE Int. Conf. on Robotics and Autonomation*, pages 2533–2540, 1992.

[38] Y. Nakamura and R. Mukherjee. Nonholonomic path planning and automation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1050–1055, 1989.

[39] C. Ó'Dúnlaing. Motion planning with inertial constraints. *Algorithmica*, 2:431–475, 1987.

[40] F. Preparata. A note on locating a set of points in a planar subdivision. *SIAM J. Comput.*, 8:542–545, 1979.

[41] J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. of Mathematics*, 145:367–393, 1990.

[42] J. H. Reif. Complexity of the generalized movers problem. In *Planning, Geometry and Complexity of Robot Motion* (J. Hopcroft, and J. Schwartz, and M. Sharir, editors), Ablex Publishing Corp., Norwood, NJ, pages 267–281, 1987.

[43] J. H. Reif and M. Sharir. Motion planning in the presence of moving obstacles. In *Proc. 26th IEEE Symp. Foundations of Computer Science*, pages 144–154, 1985.

[44] J.H. Reif and S.R. Tate. Approximate kinodynomic planning using L_2-norm dynamic bounds. *Computers Math. Applic.*, 27:29–44, 1994.

[45] J.T. Schwartz and M. Sharir. Motion planning and related geometric algorithms in robotics. In *Proc. Int. Congress of Mathematicians*, pages 1594–1611, 1986.

[46] J.T. Schwartz and M. Sharir. Algorithmic motion planning in robotics. In *Handbook of Theoretical Computer Science, Vol. A* (J. van Leeuwen, editor), Elsevier, Amsterdam, pages 391–430, 1990.

[47] H.J. Sussmann. Shortest 3-dimensional paths with a prescribed curvature bound. In *Proc. 34th Conf. on Decision and Control*, pages 3306–3312, 1995.

[48] G. Wilfong. Motion planning for an autonomous vehicle. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 529–533, 1988.

[49] G. Wilfong. Shortest paths for autonomous vehicles. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 15–20, 1989.