Curvature-Constrained Shortest Paths in a Convex Polygon

(Extended Abstract)

Pankaj K. Agarwal* Therese Biedl† Sylvain Lazard‡
Steve Robbins§ Subhash Suri¶ Sue Whitesides‖

Abstract

Let B be a point robot moving in the plane, whose path is constrained to have curvature at most 1, and let $\mathcal{P}$ be a convex polygon with n vertices. We study the collision-free, optimal path-planning problem for B moving between two *configurations* inside $\mathcal{P}$ (a configuration specifies both a location and a direction of travel). We present an $O(n^2 \log n)$ time algorithm for determining whether a collision-free path exists for B between two given configurations. If such a path exists, the algorithm returns a shortest one. We provide a detailed classification of curvature-constrained shortest paths inside a convex polygon and prove several properties of them, which are interesting in their own right. Some of the properties are quite general and shed some light on curvature-constrained shortest paths amid obstacles.

*Center for Geometric Computing, Computer Science Department, Duke University, Box 90129, Durham, NC 27708–0129, USA; pankaj@cs.duke.edu; http://www.cs.duke.edu/~pankaj/. Supported in part by National Science Foundation research grant CCR–93–01259, by Army Research Office MURI grant DAAH04–96–1–0013, by a Sloan fellowship, by a National Science Foundation NYI award and matching funds from Xerox Corporation, and by a grant from the U.S.-Israeli Binational Science Foundation.

†School of Computer Science, McGill University, 3480 University Street, Montreal, Qc, H3A 2A7, Canada; therese@cgm.cs.mcgill.ca.

‡School of Computer Science, McGill University; lazard@cgm.cs.mcgill.ca. Supported in part by an INRIA postdoctoral award.

§School of Computer Science, McGill University; stever@cs.mcgill.ca. Supported by an FCAR scholarship.

¶Department of Computer Science, Washington University Campus Box 1045, One Brookings Drive, St. Louis, MO 63130-4899, USA; suri@cs.wustl.edu;http://www.cs.wustl.edu/~suri/. Research partially supported by NSF Grant CCR-9501494.

‖School of Computer Science, McGill University; sue@cs.mcgill.ca. Supported by NSERC and FCAR research grants.

1 Introduction

The *path-planning* problem, a central problem in robotics, involves planning a collision-free path for a robot moving amid obstacles, and has been widely studied (see, e.g., the book by Latombe [17] and the survey papers by Schwartz and Sharir [25] and Halperin, Kavraki and Latombe [12]). In the simplest form, given a moving robot B, a set O of obstacles, and a pair of configurations I and F of B, we wish to find a continuous, collision-free path for B from I to F. This formulation, however, does not take into account the dynamic constraints (for instance, bounds on velocity, acceleration or curvature), the so-called *nonholonomic constraints*, imposed on a robot by its physical limitations. Although there has been considerable recent work in the robotics literature on nonholonomic motion-planning problems, (see [3, 4, 14, 16, 18, 20, 26, 31, 32] and references therein) relatively little theoretical work has been done in this important area.

The placement of a robot with k degrees of freedom is determined by a tuple of k parameters, each describing one degree of freedom; the set of all placements is called *configuration space*. In holonomic path planning, the points in configuration space reachable by B are determined solely by the the initial placement of B and possibly some obstacles. By contrast, in nonholonomic path planning, the placement of B is not sufficient to determine the placements reachable by B, e.g., another k parameters that are derivatives of the placement may be needed (see [17] for a more detailed discussion). This makes the nonholonomic path-planning problem considerably harder.

In this paper, we study the path-planning problem for a point robot whose path is constrained to have curvature at most 1. This curvature constraint arises naturally when the point robot models a real-world robot with a minimum turning radius; see for example [17]. Recently Reif and Wang [24] confirmed that the problem of deciding whether there exists a collision-free curvature-constrained path for B between two given configurations amid obstacles is NP-hard. This motivates interest in studying various special cases. In this paper we propose an efficient algorithm for computing a

curvature-constrained shortest path inside a convex polygon.

We establish several new properties of shortest paths inside a convex polygon and use these properties to characterize shortest paths. Using these properties of shortest paths and some results in computational geometry [2, 8], we present an efficient algorithm that, given initial and desired final configurations I and F in the polygon, determines whether a curvature-constrained path from I to F exists, and if so, computes a shortest one.

1.1 Previous results

Dubins [10] was perhaps the first to study curvature-constrained shortest paths. He proved that, in the absence of obstacles, a curvature-constrained shortest path from any start configuration to any final configuration consists of at most three segments, each of which is either a straight line or an arc of a circle of unit radius, assuming that the maximum curvature of the path is bounded by 1. Reeds and Shepp [23] extended this obstacle-free characterization to robots that are allowed to make reversals, that is, to back up. Using ideas from control theory, Boissonnat, Cérézo and Leblond [4] gave an alternative proof for both cases, and recently Sussmann [29] was able to extend the characterization to the 3-dimensional case. In the presence of obstacles, Fortune and Wilfong [11] gave a $2^{poly(n,m)}$-time algorithm, where n is the total number of vertices in the polygons defining the obstacles and m is the number of bits of precision with which all points are specified; their algorithm only decides whether a path exists, without necessarily finding one. Jacobs and Canny [13], Wang and Agarwal [30], and Sellen [27, 28] gave approximation algorithms for computing an ε-*robust* path. (Informally, a path is ε-robust if ε-perturbations of certain points along the path do not violate the feasibility of the path.) For the restricted case of pairwise disjoint *moderate obstacles*, i.e., convex obstacles whose boundaries have curvature bounded by 1, Agarwal, Raghavan and Tamaki [1] gave efficient approximation algorithms. Boissonnat and Lazard [5] gave an $O(n^2 \log n)$-time algorithm for computing the exact shortest paths for the case when the edges of the pairwise disjoint moderate obstacles are circular arcs of unit radius or line segments. Their algorithm works even if the obstacles intersect each other, and thus inside a convex polygon, but the running time is then $O(n^7)$. Wilfong [31] studied 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, 19, 21]. Other, more general, dynamic constraints have been considered in [6, 7, 9, 22].

1.2 Our model and results

Let B be a point robot and $\mathcal{P}$ a closed convex polygon with n vertices. For simplicity we assume that the edges of $\mathcal{P}$ are in general position: no two edges are parallel and no unit-radius circle is tangent to three edges of $\mathcal{P}$. A *configuration* X for B is a pair $(\text{LOC}(X), \psi(X))$, where $\text{LOC}(X)$ is a point in the plane representing the location of the robot and $\psi(X)$ is an angle between 0 and 2π representing its orientation. When the meaning is clear, we often write X instead of $\text{LOC}(X)$.

The image of a differentiable function $\Pi : [0, l] \to \mathbb{R}^2$ is called a *path*. We denote both the function and the path it defines by Π. We regard a path Π as oriented from $\Pi(0)$ to $\Pi(l)$. We assume a path Π is parameterized by its arc length, and $\|\Pi\|$ denotes its length. We say that Π is a path from a configuration X to another configuration Y if $\Pi(0) = \text{LOC}(X)$, $\Pi(l) = \text{LOC}(Y)$, and the oriented angles (with respect to the positive x-axis) of $\Pi'(0)$ and $\Pi'(l)$ are $\psi(X)$ and $\psi(Y)$, respectively. A path is called *moderate* if its average curvature is at most 1 in every positive-length interval.[1] This implies that the curvature is at most 1 whenever it is defined.

Any curve that lies entirely within the closed polygon $\mathcal{P}$ is called *free*. A path is *feasible* if it is moderate and free. A feasible path Π from a configuration X to another configuration Y is *optimal* if its length is minimum among all feasible paths from X to Y (it can be shown that whenever a feasible path from X to Y exists, then an optimal such path also exists [13]).

Main Results. Let $\mathcal{P}$ be a convex polygon in the plane with n vertices, and let I and F be two configurations inside $\mathcal{P}$.

(i) We prove that an optimal path from I to F consists of at most eight maximal segments, each of which is either a line segment or a circular arc of unit radius.

(ii) We give an $O(n^2 \log n)$-time algorithm to determine whether a feasible path from I to F exists. If such a path exists, then the algorithm returns an optimal path from I to F. If there are only k edges of $\mathcal{P}$ within distance 6 from I and F, then the running time of our algorithm can be improved to $O((n + k^2) \log n)$,

Our algorithm is significantly faster than the algorithm by Boissonnat and Lazard [5], whose running time was $O(n^7)$. Our paper is organized as follows. In Section 2, we present basic definitions, notation, and useful known results. In Section 3, we give a classification of the optimal path. In Sections 4 and 5, we describe our algorithms. Section 6 concludes.

2 Geometric Preliminaries

Given a configuration X, the oriented line passing through $\text{LOC}(X)$ with orientation $\psi(X)$ is denoted L_X. A configuration X *belongs* to an oriented path (or curve) Π if $\text{LOC}(X) \in$

[1] The *average curvature* of a path Π in the interval $[s_1, s_2]$ is defined by $\|\Pi'(s_1) - \Pi'(s_2)\|/|s_1 - s_2|$.

Π and L_X is the oriented tangent line to Π at $\text{LOC}(X)$. Note that a configuration X belongs to two oriented unit-radius circles. We will use $\mathcal{C}_X^+$ (resp. $\mathcal{C}_X^-$) to denote the two circles of unit radius, oriented counterclockwise (resp. clockwise) to which the configuration X belongs.

If X and Y are two points on a simple closed curve γ, then $\gamma^+[X,Y]$ (resp. $\gamma^-[X,Y]$) denotes the portion of γ from X to Y in the counterclockwise (resp. clockwise) direction, including X and Y; we will use $\gamma^+(X,Y), \gamma^-(X,Y)$ to denote portions excluding X, Y. Similarly, for a path Π and two configurations $X, Y \in \Pi$, we will use $\Pi[X,Y]$ to denote the portion of Π from X to Y.

Segments and Dubins paths. Let Π be a feasible path. We call a nonempty subpath of Π a *C-segment* (resp. *S-segment*) if it is a circular arc of unit radius (resp. line segment) and maximal. A *segment* is either a C-segment or an S-segment. While referring to a C-segments on a path Π, we will call it a C^+-segment (resp. C^--segment) if it is counterclockwise (resp. clockwise) oriented along Π. Suppose Π consists of a C-segment, an S-segment, and a C-segment; then we will say that Π is of type CSC, or C_1SC_2 if we want to distinguish between the two C-segments; superscripts $+$ and $-$ will be used to specify the orientations of C-segments of Π. Abusing the notation slightly, we will also use C_1, C_2 to denote the C-segments and S to denote the S-segment of Π. The above notation can be generalized to an arbitrarily long sequence. Dubins [10] proved the following result.

Lemma 2.1 (Dubins [10]) *In an obstacle-free environment, an optimal path between any two configurations is of type CCC or CSC, or a substring thereof.*

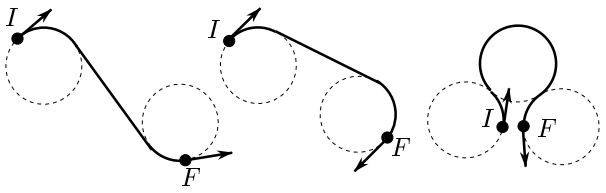

Figure 1. Different types of Dubins paths.

We will refer to paths of type CCC or CSC or substrings thereof as *Dubins paths*. In the presence of obstacles, Jacobs and Canny [13] observed that any subpath of an optimal path that does not touch any obstacle except at the endpoints is a Dubins path. In particular, they proved the following.

Lemma 2.2 (Jacobs and Canny [13]) *Let Ω be a closed polygonal environment, I an initial configuration, and F a final configuration. Then an optimal path from I to F in Ω consists of a sequence $\Pi_1 \cdots \Pi_k$ of feasible paths, where each Π_i is a Dubins path from a configuration X_{i-1} to a configuration X_i, such that $X_0 = I$, $X_k = F$, and, for $0 < i < k$, $\text{LOC}(X_i) \in \partial\Omega$.*

The above lemma implies that an optimal path in a closed polygonal environment consists of C- and S-segments. In the following, we will consider only those paths that are formed by S- and C-segments. We will refer to circles and circular arcs of unit radius simply as circles and circular arcs. Notationally, we differentiate between a C-segment and its supporting circle by using calligraphic font for the latter.

Terminal and nonterminal segments. A segment of a feasible path Π is called *terminal* if it is the first or the last segment of Π; otherwise it is called *nonterminal*. We apply the adjectives terminal and nonterminal to subpaths as well. If the first or last segment in Π is a C-segment, we will refer to it as a C_I-segment or a C_F-segment, respectively. $\mathcal{C}_I^+$, $\mathcal{C}_I^-$, $\mathcal{C}_F^+$, and $\mathcal{C}_F^-$ are called *terminal circles* (see Figure 2).

The following lemmas state some basic known properties of optimal paths; see [1, 10, 13].

Lemma 2.3 *In an optimal path inside $\mathcal{P}$,*

(i) any nonterminal C-segment has length greater than π,
(ii) any nonterminal C-segment is tangent to $\partial\mathcal{P}$ or to a terminal circle in at least one point, and
(iii) no nonterminal subpath has type CCC.

Lemma 2.4 *Let Π be an optimal path of type C_1C_2S inside $\mathcal{P}$. Let X be the common endpoint of the C_1- and C_2-segments, and let Y be the last tangent point of the C_2-segment with $\partial\mathcal{P}$ along Π. Then the length of the C_2-segment between X and Y is greater than π, i.e., $\|\Pi[X,Y]\| > \pi$.*

Anchored segments. A C-segment or circle is called *anchored* if it is tangent to $\partial\mathcal{P}$ or to terminal circles at two points. The terminal circles are not considered anchored. An anchored C-segment is denoted by $\bar{\bar{C}}$. By our general-position assumption on $\mathcal{P}$, there are a finite number of anchored circles. A C-segment tangent to $\partial\mathcal{P}$ in at least one point is denoted by $\bar{C}$.

An anchored C-segment or circle is *$\mathcal{PP}$-anchored* if it is tangent to $\partial\mathcal{P}$ at two points and *$\mathcal{PC}$-anchored* if it is tangent to $\partial\mathcal{P}$ at one point and tangent to a terminal circle at another point; see Figure 2.

A circular arc is called *long* if its length is greater than π; else it is called *short*. A $\mathcal{PP}$-anchored C-segment is called *strongly $\mathcal{PP}$-anchored* if it contains the long arc defined by the tangent points of its supporting circle with $\partial\mathcal{P}$ (see Figure 3(b)). Similarly, a $\mathcal{PC}$-anchored C-segment is called *strongly $\mathcal{PC}$-anchored* if it contains the long arc defined by two tangent points of its supporting circle with $\partial\mathcal{P}$ and with a terminal circle (see Figure 4(a)).

Pockets. Let $\mathcal{C}$ be a circle intersecting $\partial\mathcal{P}$ at two or more points, and let X, Y be two consecutive intersection points of $\partial\mathcal{P}$ with $\mathcal{C}$ so that the short arc of $\mathcal{C}$ joining X and Y lies inside $\mathcal{P}$. If $\mathcal{C}^+[X,Y]$ is the short arc and the turning angle[2]

[2]The *turning angle* of a convex polygonal chain is $\sum_i(\pi - \theta_i)$, where θ_i is the interior angle at vertex i.

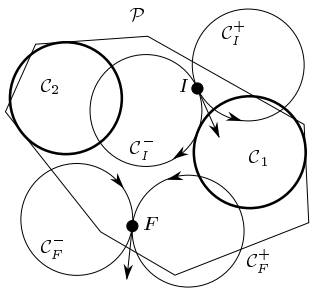

Figure 2. $\mathcal{PC}$-anchored ($\mathcal{C}_1$) and $\mathcal{PP}$-anchored ($\mathcal{C}_2$) circles.

of $\partial\mathcal{P}^+(X,Y)$ is less than π, then the closed region bounded by $\partial\mathcal{P}^+[X,Y]$ and $\mathcal{C}^+[X,Y]$ is called a *pocket* if (see Figure 3), and is denoted by $\Lambda_{\mathcal{C}}[X,Y]$. We define similarly the pocket $\Lambda_{\mathcal{C}}[X,Y]$ when $\mathcal{C}^-[X,Y]$ is the shorter arc. We will mostly be interested in pockets for which $\mathcal{C}$ is tangent to $\partial\mathcal{P}$ at X.

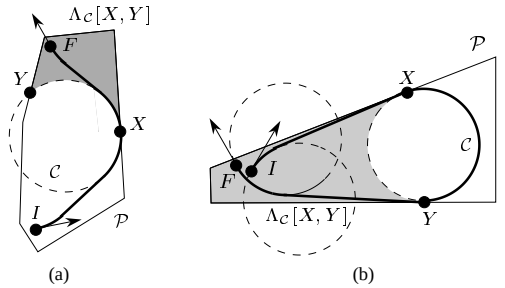

Figure 3. Pockets.

It can be verified that the condition on the turning angle implies that a pocket does not have enough room to contain a unit circle. Using this simple observation, we can prove the following lemma, which will be crucial for characterizing the optimal paths containing a strongly anchored C-segment. In particular, the lemma implies that if a feasible path enters the interior of a pocket, then it cannot escape the pocket (see Figure 3).

Lemma 2.5 *Let $\mathcal{C}$ be a circle tangent to $\partial\mathcal{P}$ at X that defines a pocket $\Lambda_{\mathcal{C}}[X,Y]$. If a feasible path Π from I to F enters (resp. escapes) $\Lambda_{\mathcal{C}}[X,Y]$ at X, then either Π contains the small arc of $\mathcal{C}$ joining X and Y, or $\Pi[X,F] \subset \Lambda_{\mathcal{C}}[X,Y]$ (resp. $\Pi[I,X] \subset \Lambda_{\mathcal{C}}[X,Y]$).*

3 Classification of Optimal Paths

The goal of this section is to prove the first of our main results, namely a detailed characterization of optimal paths in convex polygons. We show that any optimal path is of type $C_ICSCCSCC_F$ or a subsequence of this form. However, not every subsequence of the above sequence can form an optimal path. The following theorem gives a more refined description of optimal path types. Recall that a segment has non-zero length by definition. In the following, we use $\cdot$ to denote a subpath of zero length.

Theorem 3.1 *An optimal path Π inside $\mathcal{P}$ is a Dubins path or has one of the types listed below. Except in case (B.i), all the C-segments labeled $\bar{\bar{C}}$ are strongly anchored.*

(A) If Π has no nonterminal CC subpath, then Π has one of the following types:
- *(A.i) $\Pi_I S\bar{\bar{C}}S\Pi_F$ where $\Pi_I \in \{C_I, \cdot\}$, and $\Pi_F \in \{C_F, \cdot\}$ (see Figure 3(b))*
- *(A.ii) $\Pi_I S\Pi_F$ where $\Pi_I \in \{C_I\bar{\bar{C}}, C_I, \cdot\}$, and $\Pi_F \in \{\bar{\bar{C}}C_F, C_F, \cdot\}$ (see Figure 4(a))*

(B) If Π has a nonterminal CC subpath, then Π has one of the following types:
- *(B.i) $C_IC\bar{\bar{C}}C_F$ or $C_I\bar{\bar{C}}CC_F$*
- *(B.ii) $\Pi_IS\bar{\bar{C}}CC_F$ or $C_IC\bar{\bar{C}}S\Pi_F$ where $\Pi_I \in \{C_I, \cdot\}$, and $\Pi_F \in \{C_F, \cdot\}$*
- *(B.iii) $\Pi_I\bar{C}\bar{C}\Pi_F$ where $\Pi_I \in \{C_I\bar{\bar{C}}S, C_IS, C_I, S\}$, and $\Pi_F \in \{S\bar{\bar{C}}C_F, SC_F, C_F, S\}$ (see Figures 4(b), (c))*

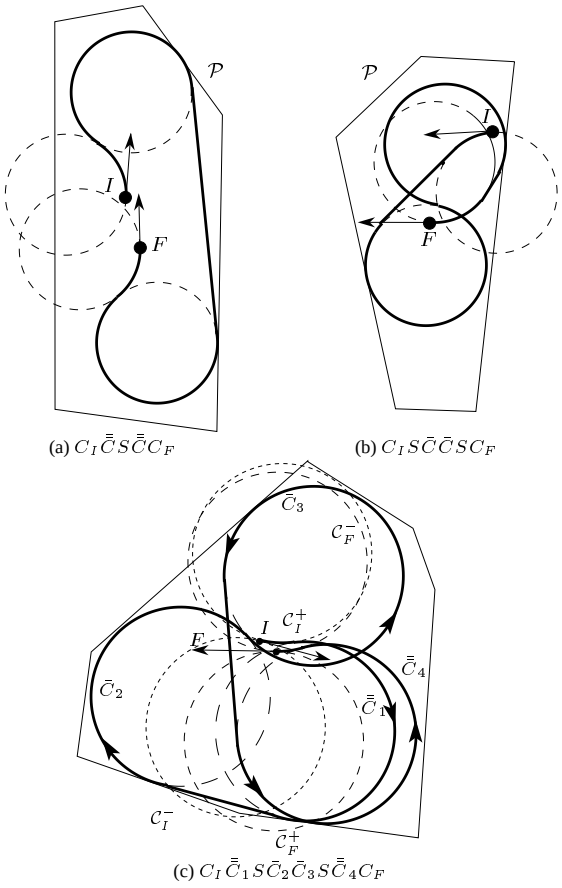

Figure 4. Examples of shortest paths.

Proposition 3.2 *The type $C_ICS\bar{C}\bar{C}SCC_F$, consisting of eight segments, does occur as an optimal path type.*

Proof (Sketch): Figure 4(c) shows an instance of $\mathcal{P}$ and initial and final configurations in which a feasible path has eight segments. We can argue that no paths of the other types described in Theorem 3.1 are feasible, which implies that the optimal path is of the given type. □

The proof of Theorem 3.1 is based on the following lemmas.

Lemma 3.3 (Agarwal, Raghavan and Tamaki [1]) *An optimal path has at most one nonterminal CC subpath. Moreover, any nonterminal C-segment that precedes (resp. follows) a C_1C_2 subpath is oriented the same way as C_1 (resp. C_2).*

Next, we state a lemma, which can be proved using geometric perturbations similar to the ones used in [1, 5].

Lemma 3.4 *(i) If an optimal path has a subpath of type SCS, then the C-segment in that subpath is strongly $\mathcal{PP}$-anchored.*
(ii) If an optimal path has a subpath of type $C_1C_2C_3S$ (or $SC_3C_2C_1$) so that the C-segment C_2 does not touch $\partial\mathcal{P}$, then C_3 is strongly $\mathcal{PP}$-anchored.

We next characterize the optimal paths that contain a strongly $\mathcal{PP}$-anchored C-segment.

Lemma 3.5 *If an optimal path Π contains a strongly $\mathcal{PP}$-anchored C-segment $\bar{\bar{C}}$, then Π is of type $C_IS\bar{\bar{C}}SC_F$, $C_IC\bar{\bar{C}}SC_F$, $C_IS\bar{\bar{C}}CC_F$, or a substring thereof (containing $\bar{\bar{C}}$).*

Proof (Sketch): By assumption, $\Pi = \Pi_I\bar{\bar{C}}\Pi_F$. $\bar{\bar{C}}$ is strongly $\mathcal{PP}$-anchored; hence its supporting circle, $\bar{\bar{\mathcal{C}}}$, has two or more intersections with $\partial\mathcal{P}$. Let X denote the first tangent point of $\bar{\bar{C}}$ with $\partial\mathcal{P}$ along Π. Let Y be the first point from X on $\bar{\bar{\mathcal{C}}}$ — moving in the opposite sense of $\bar{\bar{C}}$'s orientation — which intersects $\partial\mathcal{P}$ (see Figure 5). It is easy to prove that such a Y exists, and that $\Lambda_{\bar{\bar{\mathcal{C}}}}[X, Y]$ defines a pocket. Lemma 2.5 implies that the path up to X, i.e. Π_I and perhaps part of $\bar{\bar{C}}$, is contained in the pocket. We can also prove that Π_I consists of at most two segments, so Π_I is either C_IC, C_IS, or a substring thereof. Likewise, Π_F is CC_F, SC_F, or a substring thereof. The result follows by noting that paths of type $C_IC\bar{\bar{C}}CC_F$ are ruled out by Lemma 2.3(iii). □

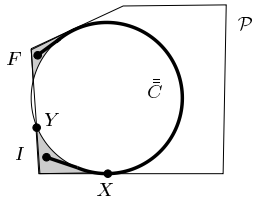

Figure 5. For the proof of Lemma 3.5. An optimal path containing a strongly $\mathcal{PP}$-anchored C-segment must start and end in a pocket.

We state now another lemma which will be useful for the algorithm. The proof is similar to that of Lemma 3.5.

Lemma 3.6 *If an optimal path Π contains a strongly $\mathcal{PC}$-anchored C-segment $\bar{\bar{C}}$ whose supporting circle is not free, then Π is of type $C_I\bar{\bar{C}}SC_F$, $C_I\bar{\bar{C}}CC_F$, $C_IS\bar{\bar{C}}C_F$, $C_IC\bar{\bar{C}}C_F$, or a substring thereof (containing $\bar{\bar{C}}$).*

We now prove Theorem 3.1.

Proof of Theorem 3.1: The proof proceeds by considering how a nonterminal C-segment may appear in Π. If there is no nonterminal C-segment in Π, then Π is of type C_ISC_F or a substring thereof, i.e., Π is a Dubins path.

Assume now that there is a nonterminal C-segment in Π. Then such a segment belongs to a subpath of Π of type either SCS or CC. Suppose Π contains a subpath of type SCS. By Lemma 3.4, the C-segment in SCS must be strongly $\mathcal{PP}$-anchored. Thus, by Lemma 3.5, Π is of type $C_IS\bar{\bar{C}}SC_F$, or substrings (containing $S\bar{\bar{C}}S$) thereof. In other words, Π is of type (A.i).

If Π contains a nonterminal C-segment but not a subpath of type SCS, we know it must contain a subpath of type CC. There are two cases to consider, depending on whether the CC subpath is terminal.

Case 1: Π does not contain any nonterminal subpath of type CC. Thus, one of the C-segments in any CC subpath must be a terminal segment. Either Π is of type C_IC_F, C_ICC_F (i.e., a Dubins path), or any nonterminal C-segment is also adjacent to an S-segment. Π must then be of type C_ICSCC_F, or any substring thereof containing S and a terminal CC. By Lemma 2.4, the nonterminal C-segments are strongly anchored. All these types of paths are covered by type (A.ii).

Case 2: Π contains a nonterminal subpath of type CC. By Lemma 3.3, it is the *only* nonterminal CC subpath in Π. Thus Π has the form $\Pi_ICC\Pi_F$. A nonterminal C-segment in Π_I must be followed by an S-segment, otherwise there will be a nonterminal CCC subpath in Π (Lemma 2.3(iii)). Furthermore, since we have no SCS subpath in Π, a nonterminal C segment must be preceded by a terminal C-segment. This means $\Pi_I = C_ICS$ or a subsequence of it. The subsequence cannot not be empty, for otherwise the middle CC subpath would be terminal; nor can it be simply CC, as noted above. Thus, $\Pi_I \in \{C_ICS, C_IS, C_I, S\}$. Similarly, $\Pi_F \in \{SCC_F, SC_F, C_F, S\}$.

If $\Pi_I = C_ICS$ or $\Pi_F = SCC_F$, then the nonterminal C-segment in Π_I or Π_F is strongly anchored by Lemma 2.4.

If both Π_I and Π_F contain an S-segment, then the nonterminal CC subpath in Π is preceded and followed by an S-segment. Thus, both C-segments of the nonterminal CC subpath in Π touch $\partial\mathcal{P}$. Indeed, otherwise Π contains a subpath of type SCC or CCS that does not touch $\partial\mathcal{P}$, which contradicts Lemma 2.2. Hence, if both Π_I and Π_F contain an S, Π is of type (B.iii).

Suppose that neither Π_I nor Π_F contains an S-segment. Then, the path is of type C_ICCC_F. One of the nonterminal C-segments must touch $\partial\mathcal{P}$ by Lemma 2.2. This C-segment is also tangent to a terminal circle and is therefore

$\mathcal{PC}$-anchored. Thus the path is of type (B.i). Note that if both nonterminal C-segments touch $\partial\mathcal{P}$, then the path is of type $C_I\bar{\bar{C}}\bar{\bar{C}}C_F$ which can be considered as type (B.i) or (B.iii).

The last case to consider is when exactly one of Π_I or Π_F contains an S-segment. Say $\Pi_I = C_I$ and $\Pi_F \neq C_F$. The path has form $C_IC_1C_2\Pi_F$ where Π_F starts with an S-segment. We know that C_2 must touch $\partial\mathcal{P}$ by Lemma 2.3(ii). If C_1 also touches $\partial\mathcal{P}$, then the path Π is of type (B.iii). Otherwise, if C_1 does not touch $\partial\mathcal{P}$, then by Lemma 3.4(ii), C_2 must be strongly $\mathcal{PP}$-anchored. Lemma 3.5 then restricts the path Π to be of type (B.ii). Similarly, if $\Pi_I \neq C_I$ and $\Pi_F = C_F$, the path Π is of type (B.ii). □

4 A Simple Algorithm

Theorem 3.1 can be used to obtain the following simple algorithm for computing an optimal path inside $\mathcal{P}$. We enumerate all possible paths of types described in Theorem 3.1 (however paths of type (B.iii) are handled specially). For each such path, we check whether it is feasible, compute the length if so, and then we return the shortest feasible path.

In order to determine whether a path is feasible, we rely on the circle-shooting data structure by Agarwal and Sharir [2] that preprocesses $\mathcal{P}$ in $O(n \log n)$ time into a data structure that makes it possible to determine in $O(\log^2 n)$ time whether a given circular arc intersects $\partial\mathcal{P}$. If the radius of all query circles is the same, then using fractional cascading [8], the data structure may be modified without affecting the preprocessing time, so that a query is answered in $O(\log n)$ time. This immediately implies the following lemma.

Lemma 4.1 *$\mathcal{P}$ can be preprocessed in $O(n \log n)$ time into a data structure that enables us to determine in $O(m \log n)$ time whether a given path consisting of m C- and S-segments is feasible.*

To bound the running time of this simple algorithm, we must count the number of candidate paths to check. We note that once a path type is given, and the supporting circles for C-segments are known, there are $O(1)$ candidate paths. These are determined by the choices of the orientations for the C-segments. Hence we are interested in the number of possible supporting circles for each path type. Note that there may be $\Omega(n^2)$ $\mathcal{PP}$-anchored circles and $\Omega(n)$ $\mathcal{PC}$-anchored circles.

There are $O(1)$ Dubins path candidates.

For paths of type (A.i) and (B.ii), once the $\mathcal{PP}$-anchored circle is chosen, there are $O(1)$ choices for other supporting circles, and hence $O(1)$ candidate paths. Since there are $O(n^2)$ $\mathcal{PP}$-anchored circles, there are $O(n^2)$ candidate paths for these two path types.

For type (A.ii), the path may have up to two $\mathcal{PC}$-anchored segments. Once their supporting circles are chosen, there are $O(1)$ path candidates. There are $O(n)$ potential $\mathcal{PC}$-anchored circles. If both anchored segments are present, we have $O(n^2)$ paths to check; otherwise, we have only $O(n)$.

Paths of type (B.i) are also determined by a $\mathcal{PC}$-anchored circle; hence there are $O(n)$ of them as well.

Paths of type (B.iii), i.e. of type $C_I\bar{\bar{C}}_1S\bar{C}_i\bar{C}_jS\bar{\bar{C}}_2SC_F$, present a special problem. If we know the supporting circles of the $\bar{C}\bar{C}$ subpath, the rest of the path is determined by a pair of $\mathcal{PC}$-anchored circles $\mathcal{C}_1, \mathcal{C}_2$, for which there are $O(n^2)$ possibilities. Unfortunately, there is an infinite family of supporting circles for the $\bar{C}\bar{C}$ subpath. The following result by Boissonnat and Lazard [5] allows us to consider only a finite set of $\bar{C}\bar{C}$ subpaths.

Lemma 4.2 (Boissonnat and Lazard [5]) *Given two configurations X and Y, and two edges e_i, e_j of $\mathcal{P}$, we can compute[3] in $O(1)$ time a finite set of paths from X to Y of type $C_1S\bar{C}_i\bar{C}_jSC_2$, where $\bar{C}_i$ and $\bar{C}_j$ are tangent to edges e_i and e_j, respectively. This set contains all optimal paths from X to Y of type $C_1S\bar{C}_i\bar{C}_jSC_2$.*

Given a pair of edges e_i, e_j and a pair of $\mathcal{PC}$-anchored circles $\mathcal{C}_1, \mathcal{C}_2$, tangent to C_I and C_F, respectively, we choose X to be the configuration determined by the intersection of C_I and $\mathcal{C}_1$ and Y to be the configuration determined by C_F and $\mathcal{C}_2$. Now by the above lemma, we can compute in $O(1)$ time a constant number of candidate paths for this pair of edges and anchored circles. Doing this for all possible pairs of edges (e_i, e_j), and pairs of $(\mathcal{C}_1, \mathcal{C}_2)$, we determine $O(n^4)$ path candidates of type (B.iii) in $O(n^4)$ time.

In summary, the simple algorithm examines $O(n^4)$ candidate paths, and for each, spends $O(\log n)$ time checking feasibility, by Lemma 4.1 with $m \leq 8$. Therefore, the overall running time is $O(n^4 \log n)$.

5 An Efficient Algorithm

In this section we prove additional properties of optimal paths that drastically reduce the number of candidates to examine. We have already shown that we need to consider only $O(1)$ Dubins paths and $O(n)$ candidates for paths of type (B.i). We will show that it suffices to consider only $O(1)$ candidate paths of type (A.i) and (B.ii), $O(n)$ candidate paths of type (A.ii), and $O(n^2)$ candidate paths of type (B.iii).

Computing paths of type (A.i) and (B.ii). The paths of types (A.i) and (B.ii) contain a strongly $\mathcal{PP}$-anchored C-segment $\bar{\bar{C}}$. The circle $\bar{\bar{\mathcal{C}}}$ supporting $\bar{\bar{C}}$ defines one or two pockets that contain a point of tangency of $\bar{\bar{\mathcal{C}}}$ with $\partial\mathcal{P}$ (see Figures 3(b) and 5). By Lemma 2.5, we know that I and F must belong to these pockets. The following lemma states that there exists at most one circle with these properties.

Lemma 5.1 *For a fixed pair of configurations I, F, there exists at most one $\mathcal{PP}$-anchored circle $\bar{\bar{\mathcal{C}}}$ so that the long arc defined by the tangent points of $\bar{\bar{\mathcal{C}}}$ with $\partial\mathcal{P}$ is free and so that I and F belong to the pocket(s) defined by $\bar{\bar{\mathcal{C}}}$ and its tangent points with $\partial\mathcal{P}$. This circle can be computed in $O(n)$ time.*

[3]The computation is performed by solving four algebraic systems of three equations in three indeterminates.

By the lemma, we can compute, in $O(n)$ time, a set of $O(1)$ candidate paths of types (A.i) and (B.ii). The candidate paths may be checked for feasibility in $O(\log n)$ time. Therefore, an optimal path of type (A.i) or (B.ii) can be computed in $O(n)$ time.

A monotonicity property of $CCSC$ paths. Subpaths of type $CCSC$ occur in both (A.ii) and (B.iii) path types. In this subsection, we ignore the polygon $\mathcal{P}$, and study paths from X to Y of type $C_1C_2SC_3$, with specified orientations on the C-segments. Then the circles $\mathcal{C}_1$ and $\mathcal{C}_3$ supporting C_1 and C_3, respectively, are fixed. Circle $\mathcal{C}_2$ is determined by M, its tangent point with $\mathcal{C}_1$. For each $M \in \mathcal{C}_1$, there is at most one path $\Pi(M)$ of type $C_1C_2SC_3$ with the specified orientations on C-segments. For certain positions of M, one of the segments may vanish. These positions of M are called *singular points*. The following lemma is proved by calculus.

Lemma 5.2 *As M moves along the oriented circle $\mathcal{C}_1$, $\|\Pi(M)\|$ increases monotonically, except at singular points.*

At singular points where a C-segment vanishes, the path length changes by $\pm 2\pi$. The S-segment vanishes when $\mathcal{C}_2$ and $\mathcal{C}_3$ have opposite orientation and are tangent.[4] Thus, there may be two singular points where the S-segment vanishes. If there are two, they split the circle $\mathcal{C}_1$ into two arcs. Along one of the arcs, circles $\mathcal{C}_2$ and $\mathcal{C}_3$ properly intersect, and so $\Pi(M)$ is not defined there. Thus, the singular points corresponding to a vanishing S-segment are the endpoints of the arc of $\mathcal{C}_1$ on which the path is defined. There may be up to six singular points. See Figure 6 for an illustration of six singular points in a path of type $C^+C^-SC^+$. All the singular points can be computed in $O(1)$ time.

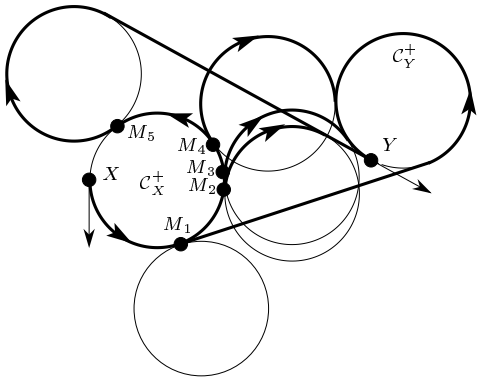

Figure 6. Paths of type $C^+C^-SC^+$ from X to Y and the six singular points X, M_1, M_2, M_3, M_4 and M_5 on $\mathcal{C}_X^+$.

Computing type (A.ii) paths. As mentioned in Section 4, we can compute in $O(n \log n)$ time the feasible candidates of type (A.ii) paths with at most one $\mathcal{PC}$-anchored segment. If the path is of type $C_I\bar{\bar{C}}S\bar{\bar{C}}C_F$, a simple analysis gives $O(n^2)$ candidates to check; we now use Lemma 5.2 to reduce the number of candidates and to compute them in $O(n \log n)$ time.

Fix the orientations of the terminal C-segments, and let $\mathcal{C}_I$ and $\mathcal{C}_F$ denote the circles supporting C_I and C_F, respectively. Let $\mathcal{K}_I$ be the sequence of $\mathcal{PC}$-anchored circles that touch $\mathcal{C}_I$ and that are free, sorted by their tangent points with $\mathcal{C}_I$. The set $\mathcal{K}_F$ is defined analogously, for $\mathcal{PC}$-anchored circles tangent to $\mathcal{C}_F$. Note that $\mathcal{K}_I$ and $\mathcal{K}_F$ can be computed in $O(n \log n)$ time, and they have $O(n)$ elements.

By Lemma 3.6, circles supporting the $\bar{\bar{C}}$-segments in an optimal path Π of type $C_I\bar{\bar{C}}_1S\bar{\bar{C}}_2C_F$ are free. Therefore, the $\bar{\bar{C}}_2$-segment of Π lies on a circle of $\mathcal{K}_F$. Suppose $\mathcal{C}_2 \in \mathcal{K}_F$ supports the $\bar{\bar{C}}_2$-segment of Π. This fixes the terminal configuration of the subpath $C_I\bar{\bar{C}}_1S\bar{\bar{C}}_2$. The above subsection on monotonicity implies we have up to up to six singular points on $\mathcal{C}_I$.

Let $\delta \subseteq \mathcal{C}_I$ be an arc joining two singular points and let $\mathcal{K}_I(\delta) \subseteq \mathcal{K}_I$ be the subsequence of circles that touch $\mathcal{C}_I$ at a point in δ. By Lemma 5.2, only the first circle of $\mathcal{K}_I(\delta)$ is a candidate for $\mathcal{C}_1$. Hence, at most six circles in $\mathcal{K}_I$ are candidates for $\mathcal{C}_1$, and they can be computed in $O(\log n)$ time by performing a binary search. By examining each $\mathcal{C}_2 \in \mathcal{K}_F$ in turn, we compute $O(n)$ candidate paths in $O(n \log n)$ time. We can therefore conclude that an optimal path of type (A.ii) can be computed in $O(n \log n)$ time.

Computing type (B.iii) paths. Let Π be an optimal path of the form $\Pi_I\bar{C}_i\bar{C}_j\Pi_F$, i.e. of type (B.iii). Suppose we know the edges e_i, e_j that are tangent to $\bar{C}_i$ and $\bar{C}_j$, respectively.

If Π does not contain any $\bar{\bar{C}}$-segment in Π_I or Π_F, then we can compute Π in $O(\log n)$ time using Lemmas 4.2 and 4.1.

Consider now the case in which Π_I and Π_F each contains a $\bar{\bar{C}}$-segment, i.e. Π is of type $C_I\bar{\bar{C}}S\bar{C}_i\bar{C}_jS\bar{\bar{C}}C_F$. We show that, given e_i and e_j, we can compute, in $O(\log n)$ time, a set of $O(1)$ candidate circles that contains the $\bar{\bar{C}}$-segments of Π. Given this set, we can compute the shortest feasible path of the above type in $O(\log n)$ time, by Lemmas 4.1 and 4.2. Thus, by considering all $O(n^2)$ pairs of edges of $\mathcal{P}$, we can compute in $O(n^2 \log n)$ time a set of $O(n^2)$ candidate paths for this case. However, we will see later that in some cases we need not consider all $O(n^2)$ pairs of edges of $\mathcal{P}$.

We first establish some simple properties of an optimal path Π of type $C_I\bar{\bar{C}}_1S\bar{C}_i\bar{C}_jS\bar{\bar{C}}_2C_F$. Assume without loss of generality that $\bar{C}_i, \bar{C}_j$ are oriented clockwise and counterclockwise, respectively. By Lemma 3.3, the $\bar{\bar{C}}_1$-segment is oriented clockwise, and the $\bar{\bar{C}}_2$-segment is oriented counterclockwise, i.e., Π is of type $C_I^+\bar{\bar{C}}_1^-S\bar{C}_i^-\bar{C}_j^+S\bar{\bar{C}}_2^+C_F^-$. Let $\bar{\bar{\mathcal{C}}}_1, \bar{\mathcal{C}}_i, \bar{\mathcal{C}}_j$, and $\bar{\bar{\mathcal{C}}}_2$ denote the circles supporting the C-segments $\bar{\bar{C}}_1$, $\bar{C}_i$, $\bar{C}_j$, and $\bar{\bar{C}}_2$, respectively.

Lemma 5.3 *If an optimal path is of type $C_I\bar{\bar{C}}_1S\bar{C}_i\bar{C}_jS\bar{\bar{C}}_2C_F$, then the circles $\bar{\bar{\mathcal{C}}}_1, \bar{\mathcal{C}}_i, \bar{\mathcal{C}}_j$, and $\bar{\bar{\mathcal{C}}}_2$ are free.*

[4] The S-segment may vanish even if $\mathcal{C}_2$ and $\mathcal{C}_3$ have the same orientation and $\mathcal{C}_2 = \mathcal{C}_3$, but in this case C_2-segment also vanishes.

Proof: Lemma 3.6 directly yields that $\bar{\bar{\mathcal{C}}}_1$ and $\bar{\bar{\mathcal{C}}}_2$ are free. Suppose now for a contradiction that $\bar{\mathcal{C}}_j$ is not free. As before, we assume that the orientations are such that $\Pi = C_I^+ \bar{\bar{C}}_1^- S \bar{C}_i^- \bar{C}_j^+ S \bar{\bar{C}}_2^+ C_F^-$. Let T be the tangent point between $\bar{C}_i$ and $\bar{C}_j$. Moving along $\bar{C}_j^+$, let X be the last tangent point between $\bar{C}_j$ and $\partial\mathcal{P}$. Starting at X and moving along $\bar{\mathcal{C}}_j^+$, let Y be the first proper intersection point between $\bar{\mathcal{C}}_j$ and $\partial\mathcal{P}$ (see Figure **??**).

By Lemma 2.4, the length of $\bar{C}_j$ between T and X is greater than π, i.e. $\|\bar{C}_j^+[T,X]\| > \pi$. It follows that $\bar{\mathcal{C}}_j$, X and Y define a pocket $\Lambda_{\bar{\mathcal{C}}_j}[X,Y]$ (see Figure **??**). By Lemma 2.5, this pocket contains $\Pi[X,F]$ and therefore contains $\bar{\bar{\mathcal{C}}}_2$. We know the free circle $\bar{\bar{\mathcal{C}}}_2$ cannot be entirely inside a pocket. The path $\bar{C}_j S \bar{\bar{\mathcal{C}}}_2$ enters the pocket at X, and since $\bar{\bar{\mathcal{C}}}_2$ is free, it is possible to escape the pocket by extending segment $\bar{\bar{\mathcal{C}}}_2$. This contradicts Lemma 2.5, establishing that $\bar{\mathcal{C}}_j$ is free. A symmetric argument shows that circle $\bar{\mathcal{C}}_i$ is free. □

We now introduce the following simple definition. Given a circle $\mathcal{C}$ and a point $X \in \mathcal{C}$, a point $M \in \mathcal{C}$ is called the *first free point after X along $\mathcal{C}^+$* if and only if the circle tangent to $\mathcal{C}$ at M is free and for any $M' \in \mathcal{C}^+[X,M)$, the circle tangent to $\mathcal{C}$ at M' is not free (in Figure 7, M' is the first free point after I along $\mathcal{C}_I^+$). Note that M could be X. The circle tangent to $\mathcal{C}$ at the first free point after X is called the *first free circle after X along $\mathcal{C}^+$*.

We show that, given I, F, e_i and e_j, we can compute in $O(\log n)$ time a set of $O(1)$ candidate circles that contain the $\bar{\bar{C}}$-segments of any optimal path from I to F of type $C_I^+ \bar{\bar{C}}_1^- S \bar{C}_i^- \bar{C}_j^+ S \bar{\bar{C}}_2^+ C_F^-$. We show how to compute candidate circles for $\bar{\bar{\mathcal{C}}}_1$; computing candidate circles for $\bar{\bar{\mathcal{C}}}_2$ is similar.

We identify two circles $\mathcal{C}'$ and $\mathcal{C}''$ that are the candidate circles for $\bar{\bar{\mathcal{C}}}_1$. Let $\mathcal{C}'$ be the first free circle after I along $\mathcal{C}_I^+$. If there is no free circle after I along $\mathcal{C}_I^+$, then $\mathcal{C}'$ and $\mathcal{C}''$ are not defined. Assume, after a possible rotation, that the line L through e_i is horizontal and $\mathcal{P}$ is above L. If the distance between L and the center of $\mathcal{C}_I^+$ is greater than 2, then $\mathcal{C}''$ is not defined. Otherwise, there exist two circles that are above L and tangent to both $\mathcal{C}_I^+$ and L. Let $\mathcal{C}_L$ be the leftmost of these two circles, and let M_L be its tangent point with $\mathcal{C}_I^+$. Let $\mathcal{C}''$ be the first free circle after M_L along $\mathcal{C}_I^+$. Note that $\mathcal{C}'$ and $\mathcal{C}''$ only depend on I, $\mathcal{C}_I^+$, and on the line L through e_i.

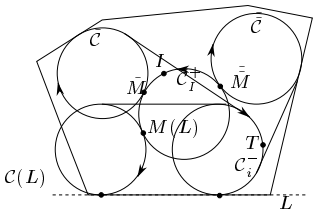

Figure 7. Definition of $\mathcal{C}'$ and $\mathcal{C}''$.

Lemma 5.4 *Let Π be an optimal path of type $C_I^+ \bar{\bar{C}}_1^- S\, \bar{C}_i^- \bar{C}_j^+ S \bar{\bar{C}}_2^+ C_F^-$, and let L be the line through the edge tangent to $\bar{C}_i$. Then $\bar{\bar{C}}_1$ is supported by $\mathcal{C}'$ or $\mathcal{C}''$.*

Proof (Sketch): We prove the lemma only in the case where $\mathcal{C}_I^+$ and $\mathcal{C}_i$ properly intersect. Let $T \in \Pi$ be the configuration at the tangent point between $\bar{C}_i$ and $\bar{C}_j$.

The circle $\mathcal{C}_1$ supporting the $\bar{\bar{C}}_1$-segment is tangent to $\mathcal{C}_I^+$. As before, any choice of a point $M \in \mathcal{C}_I^+$ defines at most one path $\Pi(M)$ of the form $C_I^+ C_1^- S C_i^-$, which begins at I and ends at T, and where $\mathcal{C}_I^+$ and $\mathcal{C}_1^-$ are tangent at M. Let M^* be the intersection point of the C_I^+- and $\bar{\bar{C}}_1^-$ segments of the optimal path Π. Then $\Pi(M^*)$ is a subpath of Π and so it is an optimal path from I to T. By the monotonicity property (Lemma 5.2), and since $\mathcal{C}_1$ and $\mathcal{C}_i$ are free (Lemma 5.3), M^* must be the first free point along $\mathcal{C}_I^+$ after a singular point of $\Pi(M)$. Since $\mathcal{C}_I^+$ and $\mathcal{C}_i$ properly intersect, there are only two singular points I and M_1 of $\Pi(M)$, where M_1 corresponds to the vanishing of $\bar{\bar{\mathcal{C}}}_i$.

If M^* is the first free point after I along $\mathcal{C}_I^+$, then C_1 is supported by $\mathcal{C}'$, the first free circle after I. If M^* is the first free point after M_1, then we show that C_1 is supported by $\mathcal{C}''$, the first free circle after M_L.

By Lemma 2.4, the arc length of $\bar{C}_i$ from its tangent point with L to T must be at least π. See Figure 4(c). In other words, T must be in the right half of $\bar{\mathcal{C}}_i$ (as L is horizontal and $\mathcal{P}$ is above L). Therefore by definition of M_L, the arc length of C_i in $\Pi(M_L)$ is less than π.

It follows that for any point $M \in \mathcal{C}_I^+[M_1, M_L]$, the arc length of C_i in $\Pi(M)$ is less than π, so by Lemma 2.3, $\Pi(M)$ cannot be part of the optimal path. Thus, M^* does not belong to $\mathcal{C}_I^+[M_1, M_L]$. So if M^* is the first free point after M_1, then it is the first free point after M_L. In other words, C_1 is supported by $\mathcal{C}''$. □

Lemma 5.5 *$\mathcal{C}'$ and $\mathcal{C}''$ can be computed in $O(\log n)$ time.*

Proof: Let Γ be the circle of radius 2 concentric with $\mathcal{C}_I^+$. Let I_Γ (resp. M_Γ) be the intersection point between Γ and the ray emanating from the center of $\mathcal{C}_I^+$ and going through I (resp. M_L) (see Figure 8). Let R be the *retracted polygon* of $\mathcal{P}$ with respect to a unit circle, i.e., R is the set of points p such that the unit circle centered at p lies inside $\mathcal{P}$; R is a convex polygonal region with at most n edges, and it can be computed in linear time. Let O' be the first intersection point between Γ and R starting at I_Γ and moving along Γ^+. The center of $\mathcal{C}'$ is O'. Indeed, by definition of R, the circle centered at O' is free, and any circle (of unit radius) centered at a point on $\Gamma^+[I_\Gamma, O')$ is not free. Similarly, the center of $\mathcal{C}''$ is the first intersection point between Γ and R starting at M_Γ and moving along Γ^+. Using the circle-shooting data structure by Agarwal and Sharir [2], R can be preprocessed in $O(n \log n)$ time, so that $\mathcal{C}'$ and $\mathcal{C}''$ can be computed in $O(\log n)$ time. □

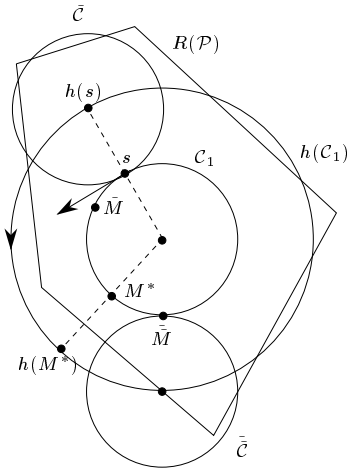

Figure 8. CRAP

By Lemmas 5.4 and 5.5, we can compute, in $O(\log n)$ time, two candidates for the circle supporting segment $\bar{\bar{C}}_1$. We can similarly compute two candidates for the circle supporting segment $\bar{\bar{C}}_2$. By Lemma 4.2, this gives us $O(1)$ candidate paths, for which we may check the feasibility in $O(\log n)$ time. Hence, given two edges e_i and e_j of $\mathcal{P}$, we can compute in $O(\log n)$ time, an optimal path of type $C_I\bar{\bar{C}}S\bar{C}_i\bar{C}_jS\bar{\bar{C}}C_F$, where $\bar{C}_i$ and $\bar{C}_j$ are tangent to e_i and e_j, respectively.

In the cases where the optimal path is of type (B.iii) with only one $\bar{\bar{C}}$-segment in Π_I or Π_F, we get similar results. For example, if an optimal path is of type $C_I\bar{\bar{C}}_1S\bar{C}_i\bar{C}_jSC_F$, then $\bar{\bar{C}}_1$ and $\bar{C}_i$ are free, and $\bar{\bar{C}}_1$ is supported by the $\mathcal{C}'$ or $\mathcal{C}''$ defined above. Thus we get the following lemma.

Lemma 5.6 *Let e_i, e_j be edges of $\mathcal{P}$. In $O(\log n)$ time, we can compute an optimal path of type $\Pi_I\bar{C}_i\bar{C}_j\Pi_F$ where $\Pi_I \in \{C_I\bar{\bar{C}}S, C_IS, C_I, S\}$, $\Pi_F \in \{S\bar{\bar{C}}C_F, SC_F, C_F, S\}$ and where $\bar{C}_i$ and $\bar{C}_j$ are tangent to e_i and e_j, respectively.*

Now we describe how to find a suitable set of pairs of edges $\mathcal{E}$ such that if an optimal path from I to F is of type (B.iii) (i.e., $\Pi_I\bar{C}_i\bar{C}_j\Pi_F$), then the pair of edges (e_i, e_j) tangent to $\bar{C}_i$ and $\bar{C}_j$ is in the set $\mathcal{E}$.

From [1], we know that if an optimal path from I to F is of type $\Pi_I\bar{C}_i^+\bar{C}_j^-\Pi_F$ such that $\bar{C}_i$ and $\bar{C}_j$ are nonterminal, then $\mathcal{C}_I^+$ intersects $\bar{\mathcal{C}}_j$ (the circle supporting $\bar{C}_j$), and $\mathcal{C}_F^-$ intersects $\bar{\mathcal{C}}_i$ (the circle supporting $\bar{C}_i$). Thus, the center of $\bar{\mathcal{C}}_j$, which is at most distance 1 from the boundary of the polygon, is at most distance 3 from I. Since centers of $\bar{\mathcal{C}}_i$ and $\bar{\mathcal{C}}_j$ are distance 2 apart, they are each distance less than 5 from I. Thus, edges e_i and e_j are distance less than 6 from I. By symmetry, they are also distance less than 6 from F. Therefore, we can consider $\mathcal{E}$ to be the set of pairs of edges of $\mathcal{P}$ that are distance less than 6 from I and F. Let k denote the number of edges of $\mathcal{P}$ distance less than 6 from I and F. Then $|\mathcal{E}| = k^2$, and $\mathcal{E}$ can be computed in $O(k^2)$ time. Lemma 5.6 then gives:

Lemma 5.7 *An optimal path of type (B.iii) can be computed in $O(k^2 \log n)$ time.*

Putting everything together, we obtain the following.

Theorem 5.8 *Given a convex polygon $\mathcal{P}$, an initial configuration I, and a final configuration F, an optimal path from I to F inside $\mathcal{P}$ can be computed in time $O((n + k^2)\log n)$, where k is the number of edges of $\mathcal{P}$ at distance less than 6 from I and F.*

Proof: We have shown in the previous subsections that the Dubins paths and the optimal paths of type (A.i), (A.ii), (B.i), and (B.ii) can be computed in $O(n \log n)$ time, while paths of type (B.iii) can be computed in $O(k^2 \log n)$ time. Choosing the shortest among all those paths yields an optimal path. □

6 Conclusion

Our classification of path types in a convex polygon yields a fast algorithm for computing an optimal path. An interesting question is whether the running time can be improved to $O(n \log n)$ by proving additional properties of paths of type (B.iii). Our results show that even for a convex polygon, optimal paths between two configurations can be rather complex. Such complex paths may be difficult to track by a mobile robot. Furthermore, they may arise as artifacts of a tightly constricted environment. A direction for future research is to seek a realistic notion of feasibility that rejects hard-to-follow paths, while admitting fast computation of optimal paths.

Acknowledgment

This paper arose from research begun at the International Workshop on Bounded Radius of Curvature, organized by S. Whitesides and held at the Bellairs Research Institute of McGill University, March 9-16, 1997.

References

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

[2] P. K. Agarwal and M. Sharir. Circle shooting in a simple polygon. *J. Algorithms*, 14:69–87, 1993.

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

[4] J.-D. Boissonnat, A. Cérézo, and J. Leblond. Shortest paths of bounded curvature in the plane. *Internat. J. Intell. Syst.*, 10:1–16, 1994.

[5] J.-D. Boissonnat and S. Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 242–251, 1996.

[6] J. Canny, B. R. Donald, J. Reif, and P. Xavier. On the complexity of kinodynamic planning. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 306–316, 1988.

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

[8] B. Chazelle and L. J. Guibas. Fractional cascading: I. A data structuring technique. *Algorithmica*, 1:133–162, 1986.

[9] B. R. Donald and P. Xavier. A provably good approximation algorithm for optimal-time trajectory planning. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 958–963, 1989.

[10] 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.

[11] S. Fortune and G. Wilfong. Planning constrained motion. *Ann. Math. Artif. Intell.*, 3:21–82, 1991.

[12] D. Halperin, L. Kavraki, and J.-C. Latombe. Robotics. In *CRC Handbook of Computational Geometry* (J. Goodmand and J. O'Rourke, eds.), pages 755–778. CRC Press, Boca Raton, NY, 1997.

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

[14] P. Jacobs, J.-P. Laumond, and M. Taix. Efficient motion planners for nonholonomic mobile robots. *In Proc. of the IEEE/RSJ Internat. Workshop on Intell. Robots and Systems*, pages 1229–1235, 1991.

[15] 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.

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

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

[18] J.-P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. *In Proc. of the Internat. Joint Conf. on Artif. Intell.*, pages 1120–1123, 1987.

[19] J.-P. Laumond, P. Jacobs, M. Taix, and R. Murray. A motion planner for nonholonomic mobile robots. *IEEE Trans. on Robot. and Autom.*, 10:577–593, 1994.

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

[21] B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proc. 9th IEEE Internat. Conf. Robot. Autom.*, pages 2533–2540, 1992.

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

[23] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.*, 145(2), 1990.

[24] J. Reif and H. Wang. The complexity of the two dimensional curvature-constrained shortest-path problem. In *Proc. 3rd Workshop on the Algo. Found. of Robotics* (P. K. Agarwal, L. E. Kavraki, and M. Mason, eds.). A. K. Peters, Wellesley, MA, 1998.

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

[26] S. Sekhavat, P. Švestka, J.-P. Laumond, and M. Overmars. Multi-level path planning for non-holonomic robots using semi-holonomic subsystems. In *Workshop on the Algo. Found. of Robotics* (J.-P. Laumond and M. Overmars, eds.), pages 79–96. A. K. Peters, Wellesley, MA, 1996.

[27] J. Sellen. Planning paths of minimal curvature, *In Proc. of the IEEE Internat. Conf. Robot. and Autom.*, 1995.

[28] J. Sellen. Approximation and decision algorithms for curvature constrained path planning: A state-space approach. In *Proc. 3rd Workshop on the Algorithmic Foundations of Robotics* (P. K. Agarwal, L. E. Kavraki, and M. Mason, eds.). A. K. Peters, Wellesley, MA, 1998.

[29] H. J. Sussmann. Shortest 3-dimentional paths with a prescribed curvature bound. In *Conf. on Decision & Control*, pages 3306–3311, 1995.

[30] H. Wang and P. K. Agarwal. Approximation algorithms for curvature constrained shortest paths. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 409–418, 1996.

[31] G. Wilfong. Motion planning for an autonomous vehicle. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 529–533, 1988.

[32] G. Wilfong. Shortest paths for autonomous vehicles. *In Proc. of the IEEE Internat. Conf. Robot. and Autom.*, pages 15–20, 1989.