An approximation algorithm for d_1-optimal motion of a rod robot with fixed rotations

M.A. Abam M. Ghodsi
Computer Engineering Department
Sharif University of Technology, Tehran, Iran

abam@ce.sharif.edu ghodsi@sharif.edu

December, 2002

Abstract

Given a translating and rotating rod robot in a plane with the presence of polygonal obstacles and two initial and final placements of the rod, the d_1-optimal motion planning problem is defined as finding a collision-free motion for the rod such that the orbit length of a fixed but arbitrary point F on the rod is minimized. In this paper, we study a special case of this problem in which the rod can translate freely, but can only rotate under pre-specified given angles around F. We will first characterize the d_1-optimal motion of the robot under the given conditions, and then present an $1 + \epsilon$ approximation algorithm for finding the optimal path. The running time of the algorithm is bounded by a polynomial in problem parameters to be specified in some details in the paper.

Keywords: Approximation algorithm, Robot motion planning, Rod robot, d_1-optimal motion

Glossary of main notations

$\Re$: The robot

F: A fixed point on $\Re$

$\overrightarrow{AB}$: A rod robot

$\{\alpha_1, \ldots, \alpha_k\}$: Specified orientations

$\Re_{\alpha_i}(x, y)$: The rod with its fixed point at point (x, y) and its orientation as α_i

$C(\Re)$: Configuration space of $\Re$

$\mathrm{R}^2 * \{\alpha_1, \cdots, \alpha_k\}$: the $C(\Re)$ for a rotating and translating $\Re$ with fixed rotations

$C_{forb}(\Re)$: Forbidden space of $\Re$

$C_{free}(\Re)$: Free space of $\Re$

$C_{obstacle}(P_i)$: forbidden space of obstacle P_i

$\mathcal{O} = \{P_1, \ldots, P_\ell\}$: Set of obstacles

n_i: Number of edges of P_i

$n = \sum_{i=1}^{\ell} n_i$: Total number of edges of all obstacles

$S = (s, \theta_s)$: Initial placement of the robot

$T = (t, \theta_t)$: Final placement of the robot

m: A motion

X_m: Orbit of the point X on $\Re$ in m

R: Real numbers

ϵ: Arbitrary positive number

L: Bound of the number of bits for each integer

$\underline{x}$: reflections of a point, arc, or a shape x about the origin

$\overset{\frown}{B_i B_j}$: An arc (a part of a disk) between B_i and B_j

$q[(p, \alpha)]$: a new point reached by rotating point q under α degrees around the origin and translating under $\vec{p}$

$\mathcal{S}_i$: set of all points (x, y) such that $(x, y, \alpha_i) \in C(\Re)$

TFS_i: Translating forbidden space in $\mathcal{S}_i$

$BTFS_i$: The boundary of TFS_i

RFS_i^j : Rotating forbidden space which is the forbidden space in $\mathcal{S}_i$ when rotating to α_j.

$BRFS_i^j$: The boundary of RFS_i^j

1 Introduction

The problem of moving a robot amidst obstacles has received good attention of some researchers. Most published results in this area are concerned with the feasibility of motion planning and few is known to have studied the optimal motion for some simple planar cases where the robot is either a disk or a rod (a directed line segment). In this paper we study the optimal motion of a rod robot when the robot can only rotate under pre-specified angles around a fixed but arbitrary point on the rod.

If F is a fixed point on the rod, the curve traced by F in any continuous motion m of the rod is called *the orbit of F*. There are several objective functions to measure a motion of the rod and the common one, so-called d_n function, is the average orbit lengths of n ($n \geq 1$) evenly distributed points on the rod. In particular, d_1 function is the orbit length of one fixed but arbitrary point on the rod. The goal in the d_n-optimal motion problem is to find the motion m having the least $d_n(m)$ of all collision-free motions between two given initial and final placements of the rod.

The d_2-optimal motion problem is to minimize the average lengths of the orbits of the two endpoints of the rod. In the absence of obstacles, this problem is called Ulam 's problem and was solved by Icking *et al.* [5]; they proved that there is an optimal motion between the initial and final placements consisting of either at most 3 rotations, or at most 2 rotations and one straight-line motion, or one rotation between 2 straight-line motions. But the same problem in the presence of obstacles remains open. O'Rourke [6] obtained a polynomial-time algorithm for the d_∞-optimal motion problem restricted to either pure translation and/or rotation by ± 90 degrees. The d_1-optimal motion problem where F is in relative interior of the rod was proved to be NP-hard [1], and when F is at one endpoints of the rod has unknown complexity.

Two approximation algorithms for this problem was presented in [1] and [2]. Since our result is based on those in these papers, some details of these algorithms are explained in Section 4 of this paper. Papadimitriou and Silverberg [7] studied the d_1-optimal problem when F is at one endpoint of the rod and its motion is restricted to straight lines between obstacle vertices. They obtained an $O(n^4 \lg n)$ time algorithm for this problem where n is the total number of obstacle edges. Sharir [8] improved this algorithm and obtained an $O(n^3\alpha(n)\lg^2 n)$ time algorithm where $\alpha(n)$ is the extremely slowly growing function of inverse of Ackermann's.

In this paper, we consider a restricted case of the d_1-optimal problem where the directed rod can only be positioned in one of the allowable k orientations of $\{\alpha_1, \cdots, \alpha_k\}$, and can translate between any two rotations. We assume that $\alpha_i \in [0, 2\pi)$ and $\alpha_1 < \alpha_2 < \cdots < \alpha_k$. More precisely, the rod can change its orientation from α_i to α_j by rotating counter-clockwise under $\alpha_j - \alpha_i$ around point F. We also assume that our work space is a 2-dimensional environment which includes a set of disjoint convex polygonal obstacles.

Our result includes characterization of the optimal motion and an $1+\epsilon$ approximation algorithm for finding this motion. The algorithm guarantees to find a motion m between any two given initial and final placements of the rod satisfying $d_1(m) \leq (1+\epsilon)d_1(m^*)$, where m^* is the optimal motion. The running time of our algorithm is bounded by a polynomial in n, the total number of obstacle edges, k the number of allowed rod rotations, L a bound of the number of bits for each input integer, and $\frac{1}{\epsilon}$ where ϵ is an arbitrary positive number.

The rest of the paper is organized as follows. In section 2, we present the basic definitions and some known relevant results. Section 3 describes the structure of the optimal motion and section 4 describes our approximation algorithm.

2 Notations and basic definitions

Let $\Re$ be a translating robot moving around a 2-dimensional environment work space that consists of a set $\mathcal{O} = \{P_1, \cdots, P_\ell\}$ of disjoint convex polygonal obstacles. The number of edges of P_i, or its size, is assumed to be n_i, and n is the problem size ($n = \sum_{i=1}^{\ell} n_i$). $\Re$ is considered a directed unit-sized vector $\overrightarrow{AB}$. F is one arbitrary fixed point on the rod which is called its reference point. A placement of translating robot $\Re$ can be specified by the coordinates of a fixed point F; $\Re(x, y)$ means that $\Re$ is placed with its reference point at (x, y). In the same way, a placement of a translating and rotating robot $\Re$ can be specified by the coordinates of reference point $((x, y))$ and the orientation of the robot (α) and denoted by $\Re_\alpha(x, y)$.

The parameter space of $\Re$ is usually called its *configuration space* and is denoted by $C(\Re)$. A point p in $C(\Re)$ corresponds to a certain placement of $\Re$ in the work space. For a translating and rotating robot, $C(\Re)$ is a 3-dimensional $\mathrm{R}^2 * [0, 2\pi)$ space. The points in $C(\Re)$ that correspond to placements where the robot intersects one of the obstacles in $\mathcal{O}$ is not permitted to be parts of any motion. The set of these points is called *forbidden space* and it is denoted by $C_{forb}(\Re)$. The rest of $C(\Re)$ is called *free space* and is denoted by $C_{free}(\Re)$.

A path of the robot is a curve in configuration space and every placement along the path maps to the corresponding point in $C(\Re)$. A collision-free path maps to a curve in the free space. We can map obstacle P_i to a set of points p in $C(\Re)$ such that $\Re(p)$ intersects P. The resulting set is called *forbidden space of obstacle* P_i and is denoted by $C_{obstacle}(P_i)$. Obviously $C_{forb}(\Re)$ is the union of $C_{obstacle}(P_i)$ for all obstacles.

The Minkowski sum of the two sets $S_1, S_2 \in \mathrm{R}^2$, denoted by $S_1 \oplus S_2$, has been defined as:

$$S_1 \oplus S_2 = \{p + q : p \in S_1, q \in S_2\},$$

where $p + q$ is the vector sum of two vectors p and q [3]. The following lemma is used to express $C_{obstacle}(P_i)$ in terms of Minkowski sum.

Lemma 1 [3] *Let $\Re$ be a planar translating robot and let P be a planar obstacle. Then, the $C_{obstacle}(P)$ is $P \oplus \underline{\Re}(0,0)$.*

As mentioned, we can specify a placement of the rod robot by a pair of $Z = (p, \alpha)$ where $p \in \mathrm{R}^2$ is the coordinates of F and α is the orientation of the rod at that point. For each placement $Z = (p, \alpha)$ of the rod and any point q in the Euclidean plane we define $q[Z]$ to be a transformation into a new plane by rotating q under α degrees around the origin followed by a translation under $\vec{p}$. Each motion m is specified by a function $m : [0, 1] \to C(\Re)$. For any point X on $\Re$, let $X_m : [0, 1] \to \mathrm{R}^2$, called the orbit of X, denotes the function $X_m(x) = X[m(x)]$. In other words, $X[m(x)]$ denotes the curve traced by point X on the rod in the work space under the motion m.

When the rod can translate and lie in any one of the allowable directions in $\{\alpha_1, \cdots, \alpha_k\}$, the configuration space is $\mathrm{R}^2 * \{\alpha_1, \cdots, \alpha_k\}$. In this case, we will define the motion m precisely as follow.

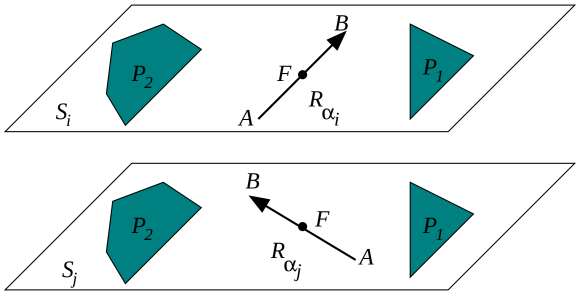

Figure 1: Two distinct placements of the rod.

A motion m of the rod from placement $S = (s, \theta_s)$ to placement $T = (t, \theta_t)$ $(s, t \in \mathrm{R}^2$ and $\theta_s, \theta_t \in \{\alpha_1, \cdots, \alpha_k\})$ is a function of the form $m : [0, 1] \to C(\Re)$, where $m(0) = S$ and $m(1) = T$ and F_m, the orbit of F, is a continuous and rectifiable function. We denote this motion by $S \stackrel{m}{\to} T$, the length of which is defined as

$$d_1(m) = sup_{(a_0=0<a_1<\cdots<a_j=1)} \sum_{i=0}^{j-1} ||\overline{F[m(s_j)]F[m(s_{j+1})]}||,$$

where the supremum is taken over all finite subdivisions $a_0 < a_1 < \cdots < a_j$ of interval $[0, 1]$ and $||.||$ denotes Euclidean length. We define the d_1 function between any two placements S and T to be $d_1(S, T) = inf_{S \stackrel{m}{\to} T} d_1(m)$ where the infimum is taken over all collision-free motions m between S and T. If m is a motion from S to T and $d_1(S, T) = d_1(m)$, we say m is a d_1-optimal motion.

3 Structure of the optimal motion

Because the forbidden space has important role in characterizing the optimal motion and is used in our algorithm, we first specify its structure.

3.1 Forbidden space

The configuration space $(C(\Re) = \mathrm{R}^2 * \{\alpha_1, \cdots, \alpha_k\})$ is a 3-dimensional space consisting of k planes $\{\mathcal{S}_1, \cdots, \mathcal{S}_k\}$ where $\mathcal{S}_i$ is the set of points corresponding to all placements where the orientation of the rod is α_i. Precisely, $\mathcal{S}_i = \{(x, y, \alpha) | \alpha = \alpha_i\}$. Without loss of generality, the points of $\mathcal{S}_i$ is denoted by two parameters (x, y). Let us denote the rod with orientation α_i as $\Re_{\alpha_i}$. Two placements of the rod is illustrated in Figure 1.

The rod performs a translating motion when moves around in each plane and does a rotating motion when moves from one plane to another. The rod should not intersect any obstacles during rotations or translations. Therefore, there are two forbidden spaces in each plane:

1. *Translating forbidden space*, denoted by TFS_i, is defined as

$$TFS_i = \{(x, y) \in \mathcal{S}_i : \Re_{\alpha_i}(x, y) \cap (\cup_{h=1}^{\ell} P_h) \neq \emptyset\}.$$

   The boundary of TFS_i is denoted by $BTFS_i$.

2. *Rotating forbidden space*, denoted by RFS_i^j $(i \neq j)$, is defined as

$$RFS_i^j = \{(x, y) \in \mathcal{S}_i : \exists \alpha \in [\alpha_i, \alpha_j], \Re_{\alpha}(x, y) \cap (\cup_{h=1}^{\ell} P_h) \neq \emptyset\}.$$

   The boundary of RFS_i^j is denoted by $BRFS_i^j$.

Thus there is one translating forbidden space and $k - 1$ rotating forbidden spaces in each plane. By Lemma 1 $TFS_i = \underline{\Re}_{\alpha}(0, 0) \oplus (\cup_{h=1}^{\ell} P_h)$.

Lemma 2 *Complexity of $BTFS_i$ is $O(n)$ and can be computed in $O(n \lg^2 n)$ time.*

Proof: Since the complexity of $\Re$ 2 (constant) and all obstacles are convex, this lemma is a simple conclusion from a lemma in [3]. □

We now concentrate on computing $BRFS_i^j$. To move from plane $\mathcal{S}_i$ to plane $\mathcal{S}_j$, the robot rotates counter-clockwise around F under $\alpha_j - \alpha_i$. During this rotation, the rod sweeps the area as illustrated in Figure 2(a). This area is denoted by $disk_{ij}$ and consists of two parts: a lower part of the disk with radius $||\overline{AF}||$ denoted by $ldisk_{ij}$, and an upper part of the disk with radius $||\overline{BF}||$ denoted by $udisk_{ij}$. We thus have,

$$RFS_i^j = \underline{disk}_{ij} \oplus (\cup_{h=1}^{\ell} P_h) = (\underline{udisk}_{ij} \oplus \cup_{h=1}^{\ell} P_h) \cup (\underline{ldisk}_{ij} \oplus \cup_{h=1}^{\ell} P_h).$$

If $udisk_{ij}$ is convex $(|\alpha_i - \alpha_j| \leq \pi)$, the following algorithm describes how we can compute $\underline{udisk}_{ij} \oplus P_h$:

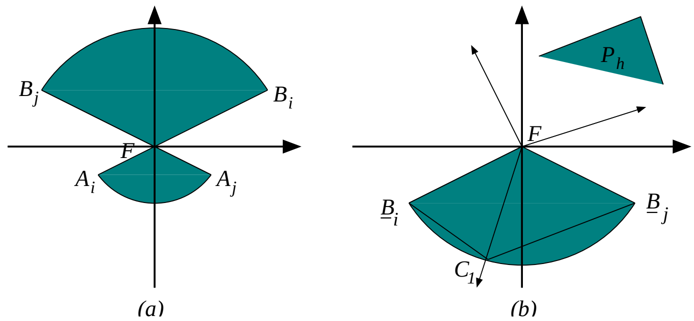

Figure 2: (a) The area swept by the rod in a rotation (b) Minkowski sum of a convex polygon and a part of a disc

1. Find the reflection of $udisk_{ij}$ about the origin. Let $\underline{B}_i$ and $\underline{B}_j$ be the reflections of B_i and B_j about the origin, respectively.

2. Compute the intersections of arc $\widehat{\underline{B}_i\underline{B}_j}$ with outer normals of edges of P_h drawn from the origin. Let $C_1, \cdots, C_g$ denote the intersection points (see Figure 2(b)). Since each normal vector intersects arc $\widehat{\underline{B}_i\underline{B}_j}$ at most once, then $g \leq n_h$. The resulting polygon $C = F\underline{B}_iC_1 \cdots C_g\underline{B}_j$ is convex.

3. Compute $P_h \oplus C$ with the algorithm described in [3].

4. Replace edges of $P_h \oplus C$ corresponding (parallel and equal) to $\overline{\underline{B}_iC_1}, \overline{C_1C_2}, \cdots, \overline{C_g\underline{B}_j}$ with the respective arcs of $\widehat{\underline{B}_iC_1}, \widehat{C_1C_2}, \cdots, \widehat{C_g\underline{B}_j}$.

If $udisk_{ij}$ is non-convex, $udisk_{ij}$ is partitioned into two convex parts u_1 and u_2. Minkowski sum of P_h and $disk_{ij}$ is now the union of two convex sets $u_1 \oplus P_h$ and $u_2 \oplus P_h$.

Lemma 3 *The algorithm correctly computes $\underline{udisk}_{ij} \oplus P_h$. Its time complexity and the complexity of $\underline{udisk}_{ij} \oplus P_h$ are both $O(n_h)$.*

Proof: Assume that $udisk_{ij}$ is convex. It is easy to observe that the first step of the algorithm correctly computes the reflection of $udisk_{ij}$ about the origin. Let $v_1, \cdots, v_{n_h}$ be the vertices of P_h and let $\vec{d_1}, \cdots, \vec{d_{n_h}}$ be the outer normals of $v_1v_2, \cdots, v_{n_h}v_1$ drawn form origin. Since the order of outer normals is the same as the order of edges of P_h, some consecutive outer normals intersect $\widehat{\underline{B}_i\underline{B}_j}$. Without loss of generality, we assume that $\vec{d_1}, \vec{d_2}, \cdots,$ and $\vec{d_g}$ intersects $\widehat{\underline{B}_i\underline{B}_j}$ at $C_1, \cdots, C_g$. It is known that the extreme point on

Minkowski sum in direction $\vec{d}$ is the sum of the extreme points on operands of Minkowski sum in the same direction. If $\vec{d}$ does not intersect $\underline{B_i}\overset{\frown}{}\underline{B_j}$, the extreme point on $\underline{udisk}_{ij}$ in direction $\vec{d}$ (one of F, $\underline{B}_i$, or $\underline{B}_j$) is the extreme point on C in the same direction, the sum of the extreme points on P_h and $\underline{udisk}_{ij}$ in the direction $\vec{d}$ is equal to sum of the extreme points on P_h and C in the same direction which are correctly computed in the third step of the algorithm. Now let $\vec{d}$ intersect $\overset{\frown}{\underline{B}_i\underline{B}_j}$ at G. Without loss of generality, let E be between C_1 and C_2. The extreme point on P_h in direction $\vec{d}$ is v_2, then the sum of the extreme points in direction $\vec{d}$ is $E + v_2$. For all directions between $\vec{d_1}$ and $\vec{d_2}$, the extreme points on P_h are v_2 and also E moves on $\overset{\frown}{C_1C_2}$. This means that we should replace the edge corresponding to v_1v_2 in $P_h \oplus C$ with the arc that is equal to $\overset{\frown}{C_1C_2}$ (the fourth step of the algorithm.)

Since the complexity of C is $O(n_h)$, the complexity of $P_h \oplus C$ is $O(n_h)$. Therefore, the complexity of $P_h \oplus \underline{udisk}_{ij}$ is $O(n_h)$.

It is straightforward to see that the first step of the algorithm requires $O(1)$ time and the second and fourth steps require $O(n_h)$ time. By Minkowski sum properties, the third step requires $O(n_h)$ time. Thus the complexity of the algorithm is $O(n_h)$. $\square$

Similar algorithm can be used to compute $P_h \oplus \underline{ldisk}_{ij}$. To compute their union, we can use a simple divide and conquer approach. We thus get the following theorem.

Theorem 4 *The complexity of $BRFS_i^j$ is $O(n^2)$ and it can be computed in $O(n^2 \lg^2 n)$ time.*

Proof: The number of edges of $P_h \oplus \underline{udisk}_{ij}$ is $O(n_h)$ for each h and $\sum_{h=1}^{\ell} n_h = n$. In the worst case, if each edge intersect all other edges, it is partitioned into $O(n)$ segments (line segments and arcs.) Since the total number of edges is $O(n)$, the total number of segments is $O(n^2)$. Therefore, the complexity of BRFS_i^j is $O(n^2)$.

Let $T(n)$ be the time needed by the algorithm. Since the merge step requires $O(n^2 \lg n)$ (the overlay algorithm presented in [3]), we have $T(n) = T(n/2) + O(n^2 \lg n)$ which results in $T(n) = O(n^2 \lg^2 n)$. $\square$

$BTFS_i$ consists of only line segments and $BRFS_i^j$ consists of line segments and arcs with radii $||\overline{FA}||$ or $||\overline{FB}||$. Other simple and useful properties are:

1. Since $\underline{\Re}_\alpha(0,0) \oplus udsik_{ij} \subset \underline{disk}_{ij} \oplus P_h$, we have $\mathrm{TFS}_i \subset \mathrm{RFS}_i^j$, i.e., the translating free space is a subset of the rotating free space.

2. Since $disk_{ij} = disk_{ji}$, we have $BRFS_i^j = BRFS_j^i$.

The rod can change its orientation form α_i to α_j by counter-clockwise rotating under $\alpha_j - \alpha_i$ or by clockwise rotation under $2\pi + \alpha_i - \alpha_j$. We have assumed the former case, but if the latter case is allowed as well, $BRFS_i^j$ is the intersection of Minkowski sum of obstacles and the swept area formed by clockwise rotation and Minkowski sum of obstacles and the swept area formed by counter-clockwise rotation. This does not change the results mentioned in previous theorems.

3.2 Characterization of the optimal motion

The following theorem summarizes the characterizations of optimal motion m between two placements $S = (s, \theta_s)$ and $T = (t, \theta_t)$.

Theorem 5 *Any optimal motion m can be transformed into a motion m' such that $F_{m'} = m_0 m_1 \cdots m_q$ is a polygonal path whose vertices, except those that the rod rotates around them, are s, t, and vertices of $BTFS_i$ $(i = 1, \cdots, k)$. The vertices that the rod rotates around them are s and the rest are on $BRFS_i^j$ $(i, j = 1, \cdots, k)$.*

Proof: Since m is optimal, each sub-motion of m is also locally optimal. Therefore, the sub-motion of m contained in $\mathcal{S}_i$ is polygonal paths whose inner vertices are, by theorem in [3], vertices of $BTFS_i$. Using this, we can say that F_m consists of a finite sequence $m_0 m_1 \cdots m_q$ where $m_h m_{h+1}$ is a line segment and $m_0 = s, m_q = t$.

Let m_r be the vertex in which the rod changes its orientation from α_i to α_j. If m_r does not belong to $BRFS_i^j$ and s, then $m_{r-1} m_r$ is parallel to $m_r m_{r+1}$. Otherwise, as illustrated in Figure 3, we can find a shorter path. This is because there is a disc centered at m_r avoiding RFS_i^j and RFS_j^i ($RFS_i^j = RFS_j^i$) and the sub-motion of m inside the disc can be replaced by a straight line segment and one rotation around one endpoint of this segment. Obviously this path is shorter. Therefore, since $m_{r-1} m_r$ and $m_r m_{r+1}$ are collinear , the rotation around m_r can be performed later at either $BRFS_i^j$ or s. □

The following theorem describes the vertices where the rod rotates around them.

Theorem 6 *If m_r is a vertex at which the rod changes its orientation from α_i to α_j, one of the following holds:*

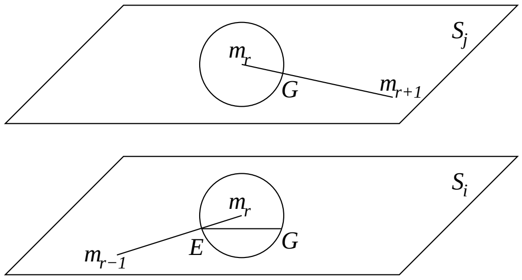

Figure 3: Performing rotations on s and $BRFS_i^j$

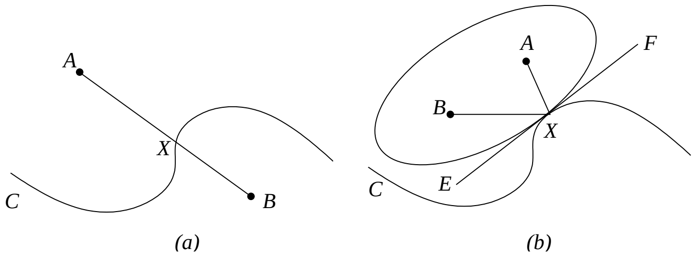

Figure 4: Shortest path between A and B passing through curve C

1. *m_r is a vertex of $BRFS_i^j$,*

2. *$m_{r-1}, m_r, and m_{r+1}$ are collinear, or*

3. *$m_r m_{r+1}$ is the reflection of $m_{r-1}m_r$ on one edge of $BRFS_i^j$.*

Proof: Let C, A, B be a curve and two distinct points in the plane, respectively. Let X be a point on C such that $||\overline{AX}|| + ||\overline{BX}||$ is minimum. If C intersects the segment AB in point D, it is obvious that $X = D$ (Figure 4(a)). Otherwise, consider an ellipse whose focuses are A, B and whose fixed value is $||\overline{AB}||$. We increase the fixed value of the ellipse till it touches the curve C. It is obvious that X is the touching point which is either one endpoint of C or an interior point of C. In the latter case, the ellipse and C are tangent at X, which means that the tangent line to the ellipse at X is tangent to C. As illustrated in Figure 4(b), we know “AXF = “BXE, i.e., BX is the reflection of AX on C.

Let m_r be on edge e of BRFS_i^j. $m_{r-1}m_r$ avoids BTFS_i and $m_r m_{r+1}$ avoids BTFS_j. Let's assume that e is a curve. Since $||\overline{m_{r-1}m_r}|| + ||\overline{m_r m_{r+1}}||$ is minimum, using above characterization, one of three features of the theorem must hold. □

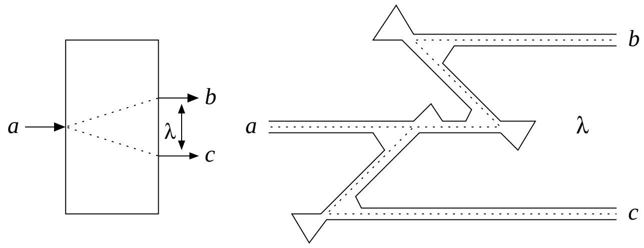

Figure 5: Schematic and detail description of wide beam spliter module

4 Approximation algorithm

As indicated in section 1, the d_1-optimal motion problem when F is a relative interior point of the rod is NP-hard. The environment used in reduction 4CNF-satisfiability to decision version of d_1-optimal motion problem consists of an assembly of certain prefabricated modules. These are fabricated from just one basic module which is illustrated in Figure 5. It is easy to see the rod can lie in two particular orientations in this module. Therefore, We can conclude that our problem is also NP-hard when F is any relative interior point of the rod. We don't know whether or not our problem is NP-hard when F is at one endpoint of the rod.

The approximation algorithm presented in [1] uses special placements in which one endpoint of the rod lies on an edge of obstacles and the rod is perpendicular to it. This algorithm can not be applied to our problem because these special placements may not happen in this case. The output motion m of this algorithm satisfies the following condition:

$$d_1(m) \leq (1+\epsilon)d_1(m^*) + O(n^2\epsilon'),$$

where ϵ, ϵ' are two positive numbers and m^* is the optimal motion. The running time of the algorithm is

$$O(n^4\alpha(n)\frac{L - \lg \epsilon'}{\lg(1+\epsilon L^{-L})}).$$

This algorithm was improved in [2] by introducing pseudo ϵ-approximation algorithm and converting it into ϵ-approximation algorithm. The pseudo ϵ-approximation algorithm satisfies

$$d_1(m) \leq d_1(m^*_R) + \epsilon R$$

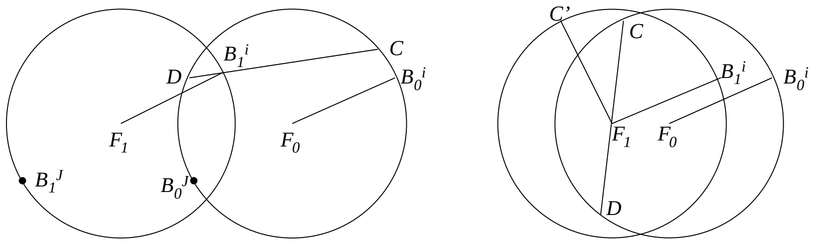

Figure 6: Intersection of the sector corresponding to the border arc CD of $BRFS_i^j$ and $BTFS_i$

where $R > 0$ is arbitrary number and m_R^* is the d_1-optimal when the orbit of F is restricted to a disk of radius R centered at s. If R is large enough then $m_R^* = m^*$.

For presentation of our approximation algorithm, we assume that F is the endpoint A of the rod. We will later show how the algorithm can be changed for the case where F is relative interior point of the rod. We first prove the following lemma that has an important role in our algorithm.

Lemma 7 *If F is at one endpoint of the rod and $\frown{CD}$ is an arc of $BRFS_i^j$, the intersection of $BTFS_i$ and the sector corresponding to $\frown{CD}$ (the area surrounded by $\frown{CD}$ and the chord CD), except C and D, is empty.*

Proof: $\frown{CD}$ is produced by placing $\Re_{\alpha_i}$ with its reference point at F_0 belonging to obstacles of work space and rotating around F_0 under $\alpha_j - \alpha_i$. $\frown{CD}$ is a part of this arc. Let B_0^i be the other endpoint of $\Re_{\alpha_i}$ when its reference point is placed at F_0 and let B_0^j be the other endpoint of $\Re_{\alpha_j}$ when its reference point is placed at F_0. $\frown{CD}$ is a part of $\frown{B_0^i B_0^j}$. Now consider $\Re_{\alpha_i}(F_1)$, F_1 being one point on the obstacles of work space. The other endpoint of $\Re_{\alpha_i}$ in this placement is denoted by B_1^i. By contradiction, we assume that the intersection of $\Re_{\alpha_i}(F_1)$ and the sector corresponding to $\frown{CD}$, except C and D, is not empty. Notice that $\{F_0, F_1\} \oplus \underline{disk}_{ij} \in \mathrm{RFS}_i^j$ and $\{F_0, F_1\} \oplus \underline{\Re}_{\alpha_i}(0,0) \in \mathrm{TFS}_i$.

Since $\frown{CD}$ is a border arc and the intersection of $F_1 B_1^i$ and sector corresponding to $\frown{CD}$ is not empty, two cases can occur: 1) B_1^i is inside the sector, and 2) F_1 is inside the sector. Figure 6 depicts the worst case where these points are interior points of the chord CD.

In the first case, since $\frown{CD}$ is a part of $\frown{B_0^i B_0^j}$ and $\frown{B_0^i B_0^j}$ is as long as $\frown{B_1^i B_1^j}$, $\frown{B_1^i B_1^j}$ is longer than $\frown{CD}$; hence $\frown{B_1^i B_1^j}$ intersects $\frown{CD}$ which contradicts the fact that $\frown{CD}$ is a border

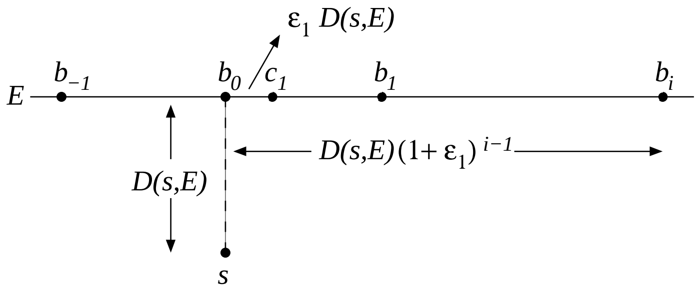

Figure 7: Breaking up line segment E into smaller segments

arc.

In the second case, two circles are homothetic with homothetic center at infinity. Let C' be the homothetic point of C; hence $F_0F_1||CC'$. $F_1C' \in \mathrm{RFS}_i^j$. It is straightforward to observe that F_1C' intersects $\overset{\frown}{CD}$, which is again a contradiction. $\square$

The main idea of our algorithm that was also used in [1] and [7], is to break up edges of BRFS_i^j into smaller segments. The algorithm includes the following steps:

1. Compute BRFS_i^j and BTFS_i for all $i, j = 1, \cdots, k$. Let μ be the total number of edges of these borders.

2. Divide each computed edge E into a set of segments by introducing points such as $b_0, b_{\pm 1}, b_{\pm 2}, \cdots$. Each segment is of the form $\sigma = [p, q]$ where p, q are consecutive introduced points on E. Throughout this algorithm, parameter ϵ_1 is taken to be $\frac{\epsilon}{8\mu}$ and $D(\sigma, \sigma')$ denotes the shortest distance between two segments σ and σ' (each point is a segment of length 0). Notice that each segment is either line segment or arc. If E is a line segment (Figure 7):

   (a) Let b_0 be the closet point on E to the starting point s, i.e., $D(s, E) = ||\overline{sb_0}||$. b_0 is either at one endpoint of E or at the projection of s upon E.

   (b) Let b_i be at distance $D(s, E)(1 + \epsilon)^{i-1}$ $(i > 0)$ from b_0 on edge E. b_{-i} is the reflection of b_i about b_0.

   (c) Segment $[b_0, b_1]$ is uniformly subdivided into segments of length $\epsilon_1 D(s, E)$, thus introducing points $c_1, c_2, \cdots$.

If E is an arc and s is an exterior point of the circle containing E (the radius r of this circle is $||\Re|| = 1$):

(a) Let b_0 be the closet point on E to s. b_0 is either at one endpoint of E or at intersection of the circle containing E and the line passing through s and the center of this circle.

(b) Let b_i be the point on E such that $|| \widehat{b_0 b_i} || = D(s,E)(1+\epsilon_1)^{i-1}$ $(i > 0)$. b_{-i} is reflect of b_i about the line passing through s and the center of circle containing E.

(c) The arc $[b_0, b_1]$ is uniformly subdivided into arcs of length $\epsilon_1 D(s,E)$, thus introducing points $c_1, c_2, \cdots$.

(d) The arc $[b_i, b_{i+1}]$ is uniformly subdivided into arcs of length $\frac{r}{2}|| \widehat{b_i b_{i+1}} ||$.

In last step, if the arc $[b_i, b_{i+1}]$ is uniformly subdivided into arcs of length $\frac{r}{4}|| \widehat{b_i b_{i+1}} ||$, these steps can be applied to divide arc E when s is an interior point of the circle containing E.

3. Construct the visibility graph G. The nodes of G are the segments from step 2 and also vertices of BTFS_i $(i = 1, \cdots, k)$, t in target plane and s in planes such as $\mathcal{S}_j$ where BRFS^j_h (let θ_s equal α_h) does not contain s. Two segments σ and σ' contained in S_i are visible to each other when there exists $x \in \sigma$ and $x' \in \sigma'$ such that the line segment $[x, x']$ avoids the interior of TFS_i. The edges of G are pairs (σ, σ') of nodes such that σ and σ' are either in the same plane and visible to each other or corresponding segments on BRFS^j_i and BRFS^i_j that are equal (Notice that $\mathrm{BRFS}^j_i = \mathrm{BRFS}^i_j$)

4. For each edge (σ, σ'), compute a nominal cost. If σ and σ' are in the same plane, this cost is Euclidean distance between the midpoints of σ and σ'. Otherwise, it is zero.

5. Apply Dijkstra 's shortest path algorithm to the weighted graph G. Let $\sigma_1, \cdots, \sigma_q$ be all nodes lying on shortest path. The output of the algorithm is a sequence of points $u_1, w_1, \cdots, u_q, w_q$ where $u_j, w_j \in \sigma_j$, $u_1 = s$, $w_q = t$ and u_j, w_{j+1} are visible to each other.

Lemma 8

1. *If σ is a segment, its length satisfies $||\sigma|| \leq \epsilon_1 D(s,\sigma)$.*

2. *There are at most $O(\frac{L}{\lg(1+\epsilon_1)})$ segments on each edge E and $O(\frac{\mu L}{\lg(1+\epsilon_1)})$ segments on all edges.*

3. *μ is bounded by $O(n^2k^2)$.*

Proof:

1. Let σ belong to edge E. If one endpoint of σ is c_i for some i, the length of σ equals $\epsilon_1 D(s,E)$. since $D(s,E) \leq D(s,\sigma)$, we have:

$$||\sigma|| = \epsilon_1 D(s,E) \leq \epsilon_1 D(s,\sigma).$$

   Let E be a line segment and $\sigma = \overline{b_h b_{h+1}}$. We have:

$$||\sigma|| = b_{h+1} - b_h = D(s,E)(1+\epsilon_1)^h - D(s,E)(1+\epsilon_1)^{h-1} = \epsilon_1 D(s,E)(1+\epsilon_1)^{h-1}.$$

   Since $D(s,\sigma) = ||\overline{sb_h}|| \geq ||\overline{b_0 b_h}|| = D(s,E)(1+\epsilon_1)^{h-1}$, then $||\sigma|| \leq \epsilon_1 D(s,\sigma)$.

   Let E be an arc and s be a exterior point of the circle containing E. The arc $\overset{\frown}{b_h b_{h+1}}$ is divided into segments with $\frac{r}{2}||\ \overset{\frown}{b_h b_{h+1}}\ ||$ long. If σ is one of them, we have:

$$D(s,\sigma) \geq ||\overline{b_0 b_h}|| \geq \frac{r}{2} D(s,E)(1+\epsilon_1)^{h-1} = \frac{1}{\epsilon_1}||\sigma||.$$

   And finally if s is an interior point of the circle containing E, We have:

$$D(s,\sigma) \geq ||\overline{sb_h}|| \geq \frac{1}{2}||\overline{b_0 b_h}|| \geq \frac{r}{4}||\ \overset{\frown}{b_0 b_h}\ || = \frac{r}{4} D(s,E)(1+\epsilon_1)^{h-1} = \frac{1}{\epsilon_1}||\sigma||.$$

2. Since the number of bits of each input integer is bounded by L, the maximum length of E is 2^L and minimum distance between two different points is 2^{-L}. Consequently, we have:

$$D(s,E)(1+\epsilon_1)^{h-1} \leq 2^L \Rightarrow h \leq \frac{L - D(s,\sigma)}{log(1+\epsilon_1)} + 1 \leq \frac{2L}{(1+\epsilon_1)} + 1.$$

The segment b_0b_1 is divided into $\frac{1}{\epsilon_1}$ segments and the segment b_ib_{i+1} $(i > 0)$ is divided at most into $\frac{1}{r}$ segments. Therefore, the number of segments on E is $O(\frac{1}{\epsilon_1} + \frac{L}{r\lg(1+\epsilon_1)})$. Since $r = 1$ and $\epsilon_1 \geq log(1 + \epsilon_1)$, the number of segments on E is $O(\frac{L}{log(1+\epsilon_1)})$. With μ edges, the total number of segments is $O(\frac{\mu L}{\lg(1+\epsilon_1)})$.

3. There are k^2 rotating forbidden spaces with complexity $O(n^2)$ and we have k translating forbidden spaces with complexity $O(n)$. Consequently, the complexity of forbidden spaces is $O(k^2n^2)$ in total, i.e., $\mu = O(k^2n^2)$.

$\square$

Theorem 9 *The output motion m of the algorithm between two placements (s, θ_s) and (t, θ_t) satisfies the following:*

$$d_1(m) \leq (1 + \epsilon)d_1((s, \theta_s), (t, \theta_t)).$$

The running time of the algorithm is bounded by $O(\frac{n^6k^3L^2}{\lg^2(1+\epsilon_1)})$.

Proof: Let m (m^*) pass through $\sigma_1, \cdots, \sigma_q$ $(\sigma'_1, \cdots, \sigma'_{q'})$. More precisely, the motion m (m^*) is a sequence $s, u_1, w_1, \cdots, u_q, w_q, t$ $(s, u'_1, w'_1, \cdots, u'_{q'}, w'_{q'}, t)$ where $u_h, w_h \in \sigma_h$ $(u'_h, w'_h \in \sigma'_h)$ and w_h and u_{h+1} $(w'_h$ and $u'_{h+1})$ are visible to each other. Let $m_1, \cdots, m_q$ $(m'_1, \cdots, m'_{q'})$ be the midpoints of $\sigma_1, \cdots, \sigma_q$ $(\sigma'_1, \cdots, \sigma'_{q'})$. The following inequality can be derived from steps 4 and 5 of the algorithm:

$$||\overline{sm_1}|| + \cdots + ||\overline{m_qt}|| \leq ||\overline{sm'_1}|| + \cdots + ||\overline{m'_{q'}t}||.$$

It is easy to see the following inequalities always hold:

$$d_1(m) \leq ||\overline{sm_1}|| + \cdots + ||\overline{m_qt}|| + 2(||\sigma_1|| + \cdots + ||\sigma_q||),$$

$$||\overline{sm'_1}|| + \cdots + ||\overline{m_{q'}t}|| \leq d_1(m^*) + 2(||\sigma'_1|| + \cdots + ||\sigma'_{q'}||.$$

We know that:

$$||\sigma|| \leq \epsilon_1 D(s, \sigma) \leq \epsilon_1 d_1(m^*).$$

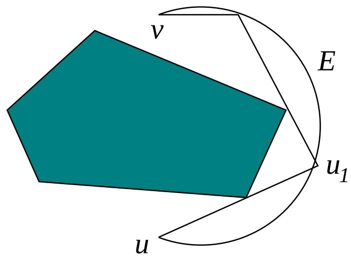

Figure 8: Partitioning the border arc E into smaller arcs

Consequently, we have, $d_1(m) \leq d_1(m^*) + 4(q+q')d_1(m^*)$.

The motions m and m^* pass through σ at most once, otherwise if σ is an arc via lemma 6 and if σ is a line segment via knowing that the shortest path between two points is a line segment whose endpoints are these points, we can simply find a shorter path. Therefore $q, q' \leq \mu$ and since $\epsilon_1 = \frac{\epsilon}{8\mu}$, we conclude that, $d_1(m) \leq (1+\epsilon)d_1(m^*)$.

Computing BTFS_i $(i = 1, \cdots, k)$ requires $O(kn \lg^2 n)$ time and computing BRFS_i^j $(i, j = 1, \cdots, k)$ requires $O(k^2 n^2 \lg^2 n)$ time. The required time to determine whether or not σ_1 and σ_2 are visible to each other is $O(n^2)$. Since there are $O(\frac{\mu L}{k \lg(1+\epsilon_1)})$ segments in each plane, and $\mu = O(k^2 n^2)$, the required time for constructing the visibility graph is $O(\frac{k^3 n^6 L^2}{\lg^2(1+\epsilon_1)})$. Finding the shortest path can be done in $O(\frac{\mu L}{\lg(1+\epsilon_1)} \lg(\frac{\mu L}{\lg(1+\epsilon_1)}) + k\frac{\mu^2 L^2}{k^2 \lg^2(1+\epsilon_1)})$ time by Dijkkstra 's algorithm. In total, the running time of the algorithm is $O(\frac{k^3 n^6 L^2}{\lg^2(1+\epsilon_1)})$. We know $\lg^2(1+\epsilon_1) \geq \epsilon_1^{2+\delta}$. Therefore, the running time of the algorithm is $O(\frac{k^{7+\delta} n^{10+\delta} L^2}{\epsilon_1^{2+\delta}})$. □

If F is a relative interior point of the rod, a new step can be added to the algorithm in between steps 1 and 2 which includes dividing each arc edge E of BRFS_i^j into arc edges such as E_1 such that intersection of the sector $\mathcal{S}_{E_1}$ corresponding to E_1 and interior points of TFS_i is empty. This can be performed as follow: Assume that u and v are endpoints of E and E is in left side of $\overrightarrow{uv}$ (Figure 8). Find vertices of BTFS_i lying inside S_E, among these vertices find vertex w such that the angle between $\overrightarrow{uw}$ and X-axis is minimum. Let u_1 be the other intersection of E and the line passing through u and w, the sector corresponding to arc $\widehat{uu_1}$ does not have interior intersection with TFS_i. We can now recursively perform these steps for arc $\widehat{u_1 v}$. Since the number of vertices of BTFS_i is bounded by $O(n)$, E is divided into at most $O(n)$ edges. Therefore, μ is bounded by $O(n^3 k^2)$.

5 Conclusion

In this paper we considered the problem of characterizing and computing the optimal motions for a translating and rotating rod robot that can only rotate under the pre-specified fixed number of angles and around a fixed but arbitrary point on rod. The latter condition in a new restriction is for the known d_1-optimal problem. We characterized the d_1-optimal motion of the robot under the given conditions, and then presented an $1+\epsilon$ approximation algorithm for finding the optimal motion of the rod. The running time of the algorithm was shown to be bounded by a polynomial in problem parameters.

References

[1] T. Asano, D. Kirkpatrick and C. K. Yap, *d_1 optimal motion for a rod*, In Proc. 12th Annu. ACM Sympos. Comput. Geom., 252-263, 1996.

[2] T. Asano, D. Kirkpatrick and C. K. Yap, *Pseudo approximation algorithms, with application to optimal motion planning*, Symposium on Computational Geometry, Barcelona, Spain, 170-178, 2002.

[3] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and its Applications*, Chapter 2, 13, 15, Springer, 1997.

[4] J. Choi, J. Sellen and C. Yap, *Approximate Eucllidean Shortest Path in 3-space*, In Proc. 10th Annu. ACM Sympos. Comput. Geom., 41-48, 1996.

[5] C. Icking, G. Rote, E. Welzl and C. Yap, *Shortest paths for line segment*, Algorithmica, 10: 182-200, 1993.

[6] J. O'Rourke, *Finding a shortest ladder path : a special case*, IMA Preprint Series 353, Institute for Mathematic and its Applications, University of Minnesota, 1987.

[7] C. H. Papadimitriou and E. B. Silverberg, *Optimal piecewise linear motion of an object among obstacles*, Algorithmica, 2:523-287, 1987.

[8] M. Sharir, *A note on the Papadimitrio-Silverberg algorithm for planning optimal piecewise-linear motion of a ladder*, NYU Robotics report, No. 188, 1989.