Path Planning among Imprecise Obstacles

Technical Report
January 18, 2016
Department of Computer Science and Information Technology
Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
www.iasbs.ac.ir

Path Planning among Imprecise Obstacles

Zahra Gholami, Mansoor Davoodi, and Sirwe Saeedi

Department of Computer Science and Information Technology,
Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
zahra.gholami87@gmail.com
mdmonfared@iasbs.ac.ir
sirwe.saedi@gmail.com

Abstract. This paper considers path planning problem in presence of imprecise obstacles for a single robot. The workspace including *start* and *end* points named s and e, respectively and its obstacles are considered to be n imprecise segments whose endpoints correspond to one of $2 \times n$ regions on the plane. We study the problem of arranging the obstacles by placing a point inside each region in such a way that maximize or minimize the shortest path distance between s and e. First we study the decision version of the Maximum Shortest Path (Max-SPP) Problem. We prove NP-completeness of the problem if the Euclidean metric is used. Following that, we obtain a similar result by proving the problem with the Manhattan distance as metric. When each imprecise region are modeled as independent disjoint disks, we propose an algorithm for Max-SPP with approximation factor $\frac{1}{2}$. If regions are dependent on separation factor k, then we obtain the approximation factor of $1 - \frac{2}{k+4}$. Furthermore, we show that the Minimum Shortest Path Problem (Min-SPP) is NP-hard for any metric L_p , $p \geq 1$. Finally, in a similar approach to Max-SPP, we propose a parameterized approximation algorithm for Min-SPP.

Keywords: Computational Geometry, NP-hard Problems, Imprecise Data, Uncertainty, Robot Path Planning, Approximation Algorithms.

1 Introduction

Regarding the widespread applications of robotics in everyday life, the necessity of investigation over motion planning problems has currently become evident. It is unsurprising, therefore, that planning a path for robots within a workspace with obstacles has been under investigated as an intriguingly applicable problem. Taking some constraints and properties of robots and workspaces into consideration, various approaches have been suggested for the path planning problem, for instance cell decomposition [21], sampling roadmaps [11] and potential fields [2]. These approaches assume the workspace, data processing, and robot motions completely in precise manner.

In addition to the mechanical constraints of robots, a raft of data is inaccessible due to the different sources of error, such as collecting real data about the world and its dynamical properties. Clearly, these uncertainties make these approaches in precise manner inefficient. Accordingly, imprecision consideration will draw a more complete and accurate picture of path planning.

Generally, in order to model the imprecision in the path planning problem, the probability and fuzzy theories have been used [23]. In contrast, this present study utilizes the less complex geometrical approaches, namely *region-based models* [16], linear parametric geometric uncertainty model (*LPGUM*) [10] and $\lambda - geometry$ [4]. In first model, a set of imprecise regions in the plane such as segments, disks or convex polygons are assumed to be a set of imprecise points. The precise point may appear anywhere in the region with a uniform probability. The goal in the region-based models is finding critical point of each geometric region in order to minimize or maximize a specific values. For example, the problem of finding Minimum Spanning Tree (MST) for some regions as imprecise points, turn out to be the problems of finding the Minimum and Maximum-weight MST [8]. While the region-based models cannot handle dependency among imprecise regions, LPGUM supports such a dependency [20]. The third model is an innovative model for handling a dynamic level of imprecision.

For a sequence of simple polygons and two points s and e, Touring Polygons Problem (TPP) is looking for a tour from s to e so that all polygons are visited in the given order. A more general form of TPP is the Shortest Path Problem (hereafter: SPP) for imprecise points. In this problem, a graph of polygons is given instead of an order of polygons. In directed graph, the traverse between vertices is only allowed through the edges. The aim of SPP is to find a placement of the vertices which minimizes the shortest distance between s and e. The maximum variant of SPP, has been studied which searched for such a placement that maximizing the shortest path distance between s and e.

An imprecise segment is a segment that at least one of its endpoints is a region instead of a point. In addition to s and e, workspace consists of imprecise segments as obstacles. Our goal in Maximum Shortest Path Problem (hereafter: Max-SPP) and Minimum Shortest Path Problem (hereafter: Min-SPP) is placing a point inside each region in order to arrange the obstacles such that the shortest path from s to e becomes maximum and minimum, respectively. In other words, Max-SPP is SPP in continuous space instead of graph.

This paper makes the following contributions:

1. In both Manhattan and Euclidean metrics, we prove NP-completeness for the decision version of *Max-SPP* when the imprecise regions are modeled as segments in the region-based models.
2. When the imprecise regions have been modeled as disjoint disks, we propose an algorithm for Max-SPP with approximation factors $1 - \frac{2}{k+4}$ and $\frac{1}{2}$ in cases where the regions are *k-separable* disks (see Definition 7) and not *k-separable* disks, respectively.
3. We show the NP-hardness of *Min-SPP* for any metric $L_p, p \geq 1$ when the imprecise regions are modeled as segments in the LPGUM.
4. We propose an algorithm for *Min-SPP* with approximation factor $1 + \frac{2}{k}$, when the imprecise regions are *k-separable* disks (see Definition 7).

The rest of paper is organized as follows: Section 2 overviews some of related works. In Section 3, we formulate Max-SPP and Min-SPP as well as other associated problems. Next, in Section 4 we show hardness results for Max-SPP and propose an approximation algorithm for it. Then, Section 5 represents our consequences of Min-SPP. Finally, we conclude our contribution in Section 6.

2 Related Works

Löffler and van Kreveld discussed the convex hull of imprecise points in various types of regions which maximize or minimize area/perimeter of the convex hull [17]. For each variant they either provide an NP-hardness proof or a polynomial-time algorithm.

By taking the parameter dependencies into consideration, Joskowicz et al. [10] introduced LPGUM for describing uncertainties of positions and shapes such as points and lines. They proved that the complexity of basic geometric entities is low-polynomial in the number of dependent parameters. Myers and Joskowicz [19] provided algorithms for the closest pair, diameter and bounding box problems as well as a few efficient algorithms for uncertain range queries in LPGUM.

Dror et al. in [9] showed that for convex and disjoint polygons TPP is solvable in polynomial time. NP-hardness of such a problem has been proved in any metrics, L_p, $p \geq 1$ in case of non-convex polygons (i.e. they are disjoint [1] or overlapping polygons [9]). This problem for non-convex rectilinear axis-aligned polygons under Manhattan metric is known to be polynomially solvable [9]. Moreover, some approximation algorithms are given for TPP in cases the polygons are non-convex [13, 22]. Also, the maximum variant of TPP is explored by Disser et al. in [7] that provided a polynomial time algorithm for computing a maximum placement.

In general, SPP is NP-hard for any metric L_p, $p \geq 1$. Disser et al. in [6] showed that for axis-aligned rectilinear polygons (not necessarily convex) under Manhattan metric, proposing a polynomial time algorithm is feasible. Their study in [7] shows that the problem is hard to approximate for any approximation factor $(1 - \epsilon)$ with $\epsilon < 1/4$, even when the polygons consist of only vertically aligned segments.

3 Problem Formulation

Free Space: In this work, we assume points of s and e to be located within the free space. In the precise manner, the free space is introduced as all points in the workspace that do not belong to any obstacles. However, in the imprecise manner, the free space refers to all points that do not belong to any obstacles for all possible placements. So, there are no placements for which obstacles contain points of s and e. In other words, s and e are not allowed to be located in the obstacles for any possible placement.

For a robot, we consider a workspace containing start point s, end point e in free apace and a set of imprecise segments as obstacles (see Fig. 1 (left)). We

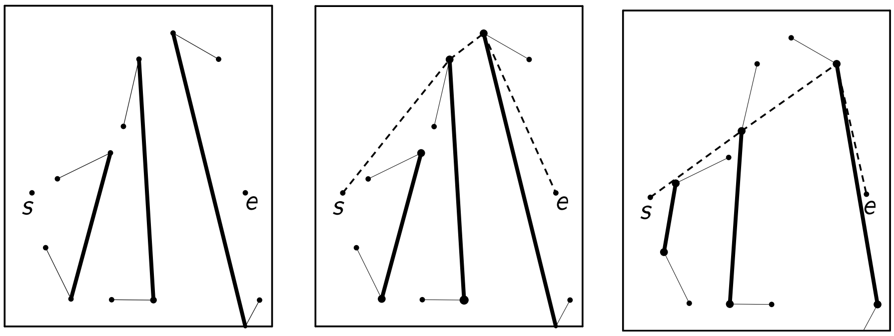

Fig. 1. Left: A workspace containing a set of imprecise segments as obstacles. Middle: An $\mathcal{I}^{max}$ placement for the workspace. Right: An $\mathcal{I}^{min}$ placement for the workspace.

define the imprecise points or regions as a set $\mathcal{R} = \{R_1, R_2, R_3, ..., R_n\}; R_i \subset \mathbb{R}^2, 1 \leqslant i \leqslant n$. where n is the number of obstacles' endpoints in workspace. Suppose $\mathcal{I}$ to be a set of points that we achieve by placing a point or *instance* inside each region of $\mathcal{R}$, like the placement

$$\mathcal{I} = \{I_1, I_2, I_3, \ldots, I_n\}; I_i \in \mathbb{R}_i, 1 \leqslant i \leqslant n \quad (1)$$

If $\mathcal{L}(\mathcal{I})$ refers to the length of the shortest path from s to e for the placement $\mathcal{I}$, then in the **Max-SPP**, the goal is to maximize $\mathcal{L}(\mathcal{I})$ by setting suitable placement like

$$\mathcal{I}^{max} = \{I_1^{max}, I_2^{max}, I_3^{max}, \ldots, I_n^{max}\}; I_i^{max} \in \mathbb{R}_i, 1 \leqslant i \leqslant n \quad (2)$$

$(\mathcal{I})^{max}$ represents a placement which maximizes the shortest path length between s and e. As an example, for the workspace in Fig. 1 (left panel), we have shown an $\mathcal{I}^{max}$ placement in the middle panel.

The Decision Version of Max-SPP:

Input: $\mathcal{R}$ as a set of imprecise points, points of s and e beside a length value of B.

Output: YES if there exists a placement like $\mathcal{I}$ that $\mathcal{L}(\mathcal{I}) \geqslant B$, NO otherwise.

We proceed to minimize $\mathcal{L}(\mathcal{I})$ for a disk robot by placing a dependent or independent single point or instance in each region of $\mathcal{R}$ for $(\mathcal{I})^{min}$ placement. This is indeed the **Min-SPP**. Here $\mathcal{I}^{min}$ represents the placement resulting the minimum possible shortest path (see Fig. 1 (right panel)).

The **Existence Path Problem** (hereafter: **EPP**) is a problem whose answer is YES if there exists a path from s to e for at least a set of $\mathcal{I} \subset \mathcal{R}$, NO otherwise.

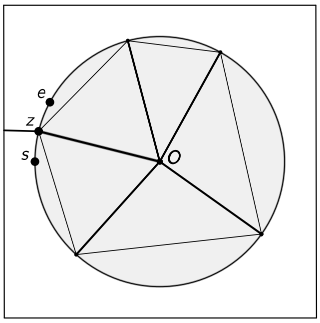

Fig. 2. A circle is divided into arcs in which all the segments passing through o represent obstacles. The points s and e located by some sufficiently small $\epsilon > 0$ above and below the separator point of z.

4 Maximum Shortest Path Problem (Max-SPP)

In this section, with the help of reduction from the SAT problem to Max-SPP, we show the hardness results for Max-SPP. Since the approach for NP-hardness proof of the Largest Convex Hull problem in [17] is not applicable for Max-SPP, we add some crucial obstacles to the workspace.

For a given SAT instance (formula ψ), we construct a Max-SPP instance. For this, we setup $\mathcal{R}(\psi)$ including imprecise obstacles's endpoint. Then, we prove that the decision version of Max-SPP returns YES if and only if the SAT formula ψ is satisfiable.

As illustrated in Fig. 2, for converting the SAT formula to the Max-SPP instance, we divide a circle into $M = c + q$ arcs which c and q are the numbers of clauses and variables in formula ψ, respectively. The c clauses and q variables of ψ are characterized as arcs in the Max-SPP instance. The circle contains one arc for each clause and one arc for each variable as well as two points s, e and M separator points that separate arcs from each other. We locate the points s and e by some sufficiently small $\epsilon > 0$ above and below the separator point of z. In addition, to insert some obstacles we draw segments from circle center at o to all separator points and a segment from point of z to the workspace boundary.

Variable Arcs Configuration: As Fig. 3 (left) shows, for each variable in ψ like v, we have an arc that contains:

(a) An imprecise region named $\overline{lr}$.
(b) A segment parallel to $\overline{lr}$ (shown as $\overline{tf}$).
(c) Two sets of points, P_v and Q_v, with the same number of elements equal to $3c$.

Notably, although the points in P_v corresponding to each variable like v are placed such that they are all on the convex hull of $\{l, r, f\}, P_v$ and Q_v, they are

Fig. 3. left: A variable arc. right: A clause arc.

not on the convex hull of $\{l, r, t\}, P_v$ and Q_v. As shown in Fig. 3 (left), points in Q_v are symmetrical with P_v.

Clause Arcs Configuration: As Fig. 3 (right) shows, for each clause in ψ like c we have an arc contains of point h_c. If the variable v appears in clause c as a positive literal, we connect the point h_c to a member of P_v. If the variable v appears in clause c as a negative literal, we connect the point h_c to a member of Q_v. In this way, the segments as imprecise regions would be produced.

- the connection between the workspace boundary and the separator point of z.
- the connection between the circle center at o and all separator points.
- the connection between the circle center at o and the imprecise regions with an endpoint h_c and the other in P_v or Q_v sets.
- the connection between the circle center at o and the imprecise regions with endpoints at t and f.

Finally, in the workspace constructed by formula ψ, maximizing the shortest path between s and e is equivalent to the sum of the maximized shortest paths between two separator points. So, in order to maximize the shortest path between two separator points (which locate on a single arc) for every variable arc like v, the selected endpoint should be either t together with all points in Q_v or f together with all points in P_v. Moreover, for the optimal placement of $\mathcal{I}^{max}$ point h_c should be selected in each clause arc such as c.

Definition 1 *We define* $\boldsymbol{B}$=$\mathcal{L}(\mathcal{I})$*, if the placement* $\mathcal{I}$ *contains* t*, points in* Q_V *in all variable arcs and points of* h_c *in all clause arcs.*

Theorem 1. *Suppose we are given a workspace contains a set of segment obstacles and a set of imprecise points as obstacles' endpoints. Obstacles are assumed to be arbitrary segments that can have common intersections only at their endpoints. For such a workspace with these imprecise obstacles, Max-SPP is NP-hard under the Euclidean metric and its decision version is NP-complete.*

Proof. We show that for the value of B the decision version of Max-SPP returns YES if and only if the formula ψ is satisfiable.

True or false value for any variable is achievable if there exists a satisfying truth assignment for formula ψ. So, from all segments $\overline{tf}$ we take point t for true variables and f for false variables. These selections form the placement of $\mathcal{I}$ for variable arcs.

If value of variable v is

- **True** (i.e. point t), then for setting placement $\mathcal{I}$ we take points in set Q_v from all variable arcs. In addition, we choose point h_c from each clause arc. In other words, we take point h_c from at least one of segments $\overline{hp_i}, 1 \leqslant i \leqslant n$ (note that $p_i \in P_v$ for some i).
- **False** (i.e. point f), then for setting placement $\mathcal{I}$ we take points in set P_v from all variable arcs. In addition, we choose point h_c from each clause arc. In other words, we take point h_c from at least one of segments $\overline{hq_i}, 1 \leqslant i \leqslant n$ (note that $q_i \in Q_v$ for some i).

Owing to the fact that points of Q_v and P_v are symmetric, we conclude that $\mathcal{L}(\mathcal{I}) = B$. Therefore, the decision version of Max-SPP returns YES for the value of B.

Now, returning YES for the value of B in the decision version of Max-SPP means that there exists a placement $\mathcal{I}$ for which $\mathcal{L}(\mathcal{I}) = B$. Then the placement $\mathcal{I}$ consists of either the mentioned points in Definition 1 or point f and set P_v in the variable arcs instead of t and Q_v. According to the value of shortest path, all the points h_c in clause arcs must be necessarily selected. This implies that according to the variables values there exists a satisfying truth assignment for formula ψ. ■

So far, we showed the NP-hardness of Max-SPP under the Euclidean metric. The question arises as whether the Theorem 1 is still correct under the Manhattan metric or not? As illustrated in Fig. 4, we have the same obstacles as we had in Theorem 1. Now, for maximizing the shortest path from point l to r in a variable arc, we do not need to choose point t together with points in set Q_v nor point f together with points in set P_v. In other words, the points of P_v or Q_v have no effect on the length of the path.

Theorem 2. *Suppose we are given a workspace contains a set of segment obstacles and a set of imprecise points as obstacles' endpoints. Obstacles are assumed to be arbitrary segments that can have common intersections only at their endpoints. For such a workspace with these imprecise obstacles, Max-SPP is NP-hard under the Manhattan metric and its decision version is NP-complete.*

Proof. We prove this theorem in the same way of theorem 1. The only difference is that some proper obstacles are added to the Max-SPP instance in order to make s to e shortest paths equivalent in both the Manhattan and the Euclidean metrics.

In addition, according to Fig. 5 we modify the variable arcs by adding points i_l and i_r in arcs $\widehat{12}$ and $\widehat{23}$. Then, we have two new obstacles by connecting i_l and i_r to the workspace boundary.

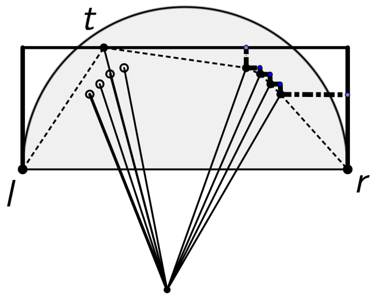

Fig. 4. Solid thick lines represent the shortest path from point l to r under the Manhattan metrics and dashed lines are the shortest path under Euclidean metrics.

Definition 2 *A number of connected points are introduced as a chain. Now, if at least one of the points is a imprecise region then the chain is known as an* ***imprecise chain****.*

Definition 3 *Members of* ***set G*** *are defined as points located in a sufficiently small distances above each point of sets P_v and Q_v. In more accurate words, none of the members of set G can be horizontally aligned with their next and previous points in sets P_v and Q_v (refer to Fig. 5).*

After adding points of set G to the variable arcs, we connect these points to the imprecise region of $\overline{tf}$ in the corresponding arc. Furthermore, we obtain some imprecise chains by connecting the imprecise region $\overline{tf}$ to the circle center at o and the points of set G.

Eventually, by adding the new obstacles to the workspace in Theorem 1, we claim that whether point f together with all points in set P_v or point t together with all points in set Q_v must be selected in order to find the optimal placement in Manhattan metric. ■

Up to this point, we proved in Theorem 1 and 2 that Max-SPP in region-based models under the mentioned conditions is NP-hard. Therefore, one can also prove in the same way that in $\lambda - geometry$ model for any constant value of λ, Max-SPP is NP-hard and its decision version is NP-complete.

4.1 Approximation Algorithm for Max-SPP

Regarding the NP-hardness of Max-SPP, the approximation algorithms could be used for estimating the solution. For approximating the optimal placement of Max-SPP, we focus on those workspaces which their obstacles are just segments and their imprecise endpoints are disjoint disks. Our approximation algorithm simply selects center of disks as placement $\mathcal{I}$ (i.e. as an approximate placement). We have proved that in this algorithm, $\mathcal{L}(\mathcal{I})$ is not smaller than half of that in optimal placement.

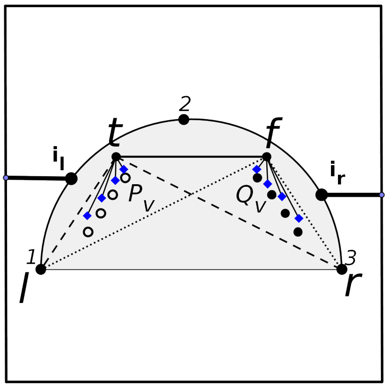

Fig. 5. New variable arc in Manhattan metric.

Definition 4 *Let* $\boldsymbol{\mathcal{L}(\mathcal{I}^{center})}$ *and* $\boldsymbol{SP(\mathcal{I}^{center})}$ *denote respectively the solution value and the shortest path of the approximation algorithm that selects the disks' centers as the placement* $\mathcal{I}$*.*

Definition 5 *We define* $\boldsymbol{SP(\mathcal{I}^{max})}$ *and* $\boldsymbol{\mathcal{L}(\mathcal{I}^{max})}$ *as the shortest path and its length in the optimal placement for Max-SPP, respectively.*

Definition 6 *We suppose* $\boldsymbol{SP'(\mathcal{I}^{max})}$ *is the path from* s *to* e *with the same topology*[1] *as* $SP(\mathcal{I}^{center})$*. Noticeably, this path is not necessarily the shortest path.*

Let $\mathcal{L}(\mathcal{I}^{max})$ and $\mathcal{L}(SP'(\mathcal{I}^{max}))$ stand for the length of paths from s to e for paths $SP(\mathcal{I}^{max})$ and $SP'(\mathcal{I}^{max})$, respectively. Then we have

$$\mathcal{L}(\mathcal{I}^{max}) \leqslant \mathcal{L}(SP'(\mathcal{I}^{max})) \tag{3}$$

Theorem 3. *Consider a workspace such that the imprecise obstacles' endpoints are disjoint disks. Now, in the approximation algorithm assuming the center of all disks as the placement* $\mathcal{I}^{center}$ *for Max-SPP, we have*

$$\frac{1}{2}\mathcal{L}(\mathcal{I}^{max}) \leqslant \mathcal{L}(\mathcal{I}^{center}) \tag{4}$$

Proof. First, we show the correctness of the following equation.

$$\mathcal{L}(SP'(\mathcal{I}^{max})) \leqslant 2\mathcal{L}(\mathcal{I}^{center}) \tag{5}$$

As $SP'(\mathcal{I}^{max})$ and $SP(\mathcal{I}^{center})$ are topologically the same, then we have Fig. 6 for any disks throughout the path $SP'(\mathcal{I}^{max})$ or $SP(\mathcal{I}^{center})$. Where a and b

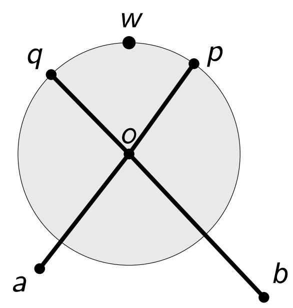

Fig. 6. A sample disk throughout path $SP(\mathcal{I}^{center})$ or $SP'(\mathcal{I}^{max})$.

represent points outside the disk and o refers to center of disk. Point w is also a member of $\mathcal{I}^{max}$ corresponding to the disk throughout the path $SP'(\mathcal{I}^{max})$.

According to the fact that p and q are the farthest points from a and b on the disk's perimeter, respectively, we obtain Eq. (6a). On the other side, since the disks are completely disjoint then $\overline{ao} > \overline{op}$ and $\overline{bo} > \overline{oq}$. Again we emphasize that the points s and e are in free space. So, we acquire:

$$\overline{ao} + \overline{op} \geq \overline{aw} \;\;\&\;\; \overline{bo} + \overline{oq} \geq \overline{bw} \tag{6a}$$

$$\Rightarrow 2\overline{ao} \geq \overline{aw} \;\;\&\;\; 2\overline{bo} \geq \overline{bw} \tag{6b}$$

$$\Rightarrow 2(\overline{ao} + \overline{bo}) \geq \overline{aw} + \overline{bw} \tag{6c}$$

Where $\overline{ao} + \overline{bo}$ and $\overline{aw} + \overline{bw}$ are respectively parts of paths $SP(\mathcal{I}^{center})$ and $SP'(\mathcal{I}^{max})$. Obviously, these two relations are valid for all the disks throughout these paths. Then, Eq. (6c) is valid for all $SP(\mathcal{I}^{center})$ and $SP'(\mathcal{I}^{max})$ subpaths. Consequently, equations (6) and (3) lead us to:

$$2\mathcal{L}SP(\mathcal{I}^{max}) \geq \mathcal{L}(SP'(\mathcal{I}^{max})) \tag{7}$$

$$\Rightarrow \mathcal{L}(\mathcal{I}^{max}) \leq \mathcal{L}(SP'(\mathcal{I}^{max})) \leq 2\mathcal{L}(\mathcal{I}^{center}) \tag{8}$$

■

If the disks are sufficiently far from each other, the approximation factor of the algorithm in Theorem 3 will be improved to more accurate values. So, in the following we prove that the larger the distances between disks, the better the approximation factor we get (i.e. closer to 1).

[1] The sequence of obstacles and their endpoints throughout a path.

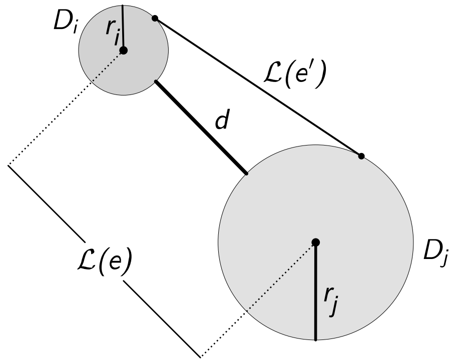

Fig. 7. An edge connects two arbitrary disks in path $SP(\mathcal{I}^{center})$.

Definition 7 *As defined by [14], a given set of disks with the largest radius* r_{max} *are **k-separable** when for the maximum value of* k *the minimum distance between each pair of disks is at least* $k.r_{max}$*.*

Theorem 4. *Consider a workspace such that the imprecise obstacles' endpoints are k-separable disks with* $k > 0$*. Now, in the approximation algorithm assuming the center of all disks as the placement* $\mathcal{I}^{center}$ *for Max-SPP, we have*

$$(1 - \frac{2}{k+4})\mathcal{L}(SP(\mathcal{I}^{max})) \leqslant \mathcal{L}(SP(\mathcal{I}^{center})) \tag{9}$$

Proof. Here, we assume $SP(\mathcal{I}^{max})$, $SP'(\mathcal{I}^{max})$ and $SP(\mathcal{I}^{center})$ are similar to those in theorem 3.

As exhibited in Fig. 7, edge e connects two arbitrary disks D_i and D_j in $SP(\mathcal{I}^{center})$ path. Now, the length of this edge is $\mathcal{L}(e) = d + r_i + r_j$, where r_i and r_j represent the radius of disks D_i and D_j, respectively and d denotes the minimum distance between disks.

In the $SP'(\mathcal{I}^{max})$ path, e' is an edge between two arbitrary points of D_i and D_j. Clearly, we have $\mathcal{L}(e') \leqslant d + 2r_i + 2r_j$ and then:

$$\begin{aligned} &\frac{\mathcal{L}(e)}{\mathcal{L}(e')} = \frac{r_i + r_j + d}{\mathcal{L}(e')} \geqslant \\ &\frac{r_i + r_j + d}{2r_i + 2r_j + d} \geqslant \frac{r_i + r_j + k.r_{max}}{2r_i + 2r_j + k.r_{max}} \geqslant \\ &\frac{r_{max} + r_{max} + k.r_{max}}{2r_{max} + 2r_{max} + k.r_{max}} = \frac{k+2}{k+4} = 1 - \frac{2}{k+4} \end{aligned} \tag{10}$$

Since equations (10) are valid for all disk pairs in the $SP(\mathcal{I}^{center})$ path, throughout the path we have:

$$\begin{aligned} &\frac{\mathcal{L}(SP(\mathcal{I}^{center}))}{\mathcal{L}(SP'(\mathcal{I}^{max}))} \geq \frac{k+2}{k+4} \\ &\Rightarrow \mathcal{L}(SP(\mathcal{I}^{center})) \geq \frac{k+2}{k+4}\,\mathcal{L}(SP'(\mathcal{I}^{max})) \end{aligned} \tag{11}$$

Eventually, from equations (3) and (11) we obtain:

$$\frac{k+2}{k+4}\mathcal{L}(SP(\mathcal{I}^{max})) \leqslant \frac{k+2}{k+4}\mathcal{L}(SP'(\mathcal{I}^{max})) \leqslant \mathcal{L}(SP(\mathcal{I}^{center})) \tag{12}$$

Therefore, we showed that the approximation factor is $1 - \frac{2}{k+4}$; and tends to 1 for larger values of k. ■

Note that we obtain this result when the center of all disks are assuming to be the placement $\mathcal{I}^{center}$. Now, obviously farther disks (i.e. larger value of k) leads the algorithm to more accurate approximation factors (i.e. closer to 1).

5 Minimum Shortest Path Problem (Min-SPP)

Here, we prove that Min-SPP is NP-hard for any metric $L_p\,, p \geq 1$. For this first, we show that the Existence Path Problem (EPP) is reducible to the Min-SPP. In other words, we show that

$$EPP \leqslant_p Min - SPP \tag{13}$$

Then by reducing the 3SAT problem to EPP, we show the NP-hardness of Min-SPP. Now, for prove the (13) reduction with respect to the same inputs for both problems, it is enough to show that the Min-SPP output will lead us to the output of EPP.

Having a set of optimal instances ($\mathcal{I}^{min}$) as the output that minimizes the shortest path, we are able to calculate the length of shortest path by applying associated algorithms. Taking any length values except infinity, means that there exists at least one path from s to e. Hence, the EPP returns YES; otherwise if the path has an infinite length, there would be no path from s to e for any possible $\mathcal{I}$. So, the EPP will return NO.

Theorem 5. *Suppose we are given a workspace contains a set of segment obstacles and a set of horizontal unit-length segments in the LPGUM as imprecise points for some obstacles' endpoints. Obstacles are assumed to be horizontal or vertical segments that can have common intersections only at their endpoints. In such a workspace for a disk robot Min-SPP and EPP are NP-hard for any metric* $L_p\,, p \geq 1$.

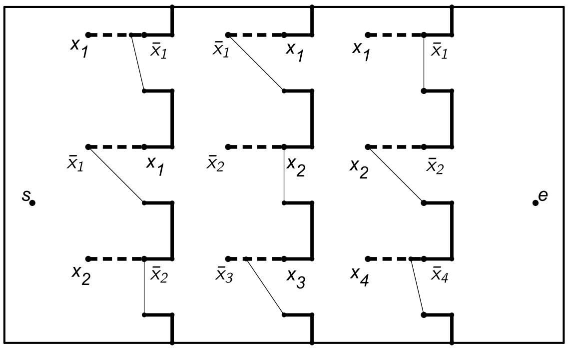

Fig. 8. Conversion of a 3SAT input like $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee x_4)$, to EPP's.

Proof. Here, we will not pursue the details on the LPGUM because this would take us too far from our goal. For more details about this model see [10].

For proving the NP-hardness of EPP, we show that the 3SAT problem is reducible to EPP. To this end, EPP returns YES if and only if the 3SAT formula ψ is satisfiable.

Converting the 3SAT input to the input of EPP should satisfy the following conditions:

1. If at least a variable in one clause takes a true value, then there must be an unobstructed path for the robot from the gate corresponding to this clause.
2. The path from s to e exists in EPP if there exist paths from the gates corresponding to the all clauses in ψ. In other words, the s to e path will be blocked even if the gate of one clause is obstructed.
3. The variable x_i in an arbitrary clause in formula ψ can not take true and false values, simultaneously. It means that variables should be either true or false.

As Fig. 8 shows, for satisfying these three conditions we construct the EPP workspace from each formula ψ as follows. First of all,if there is m clause in formula ψ corresponding to the each clause of $C_j, 1 \leqslant j \leqslant n$, we consider a gate with three entrance. In this configuration, thicker and more thin segments represent precise obstacles and imprecise obstacles with one imprecise endpoint, respectively. Notice that the unit-length segments with endpoints 1 and 2 in the left panel of Fig. 8 show imprecise endpoint of obstacles. We assume a disk robot with the radius of half length of the unit-length segments. So, it passes through a gate when it is totally open.

Secondly, according to Fig. 8, for imprecise segments we place variables on vertices 1 and negated variables on vertices 2. Therefore, all clauses meet condition 1 of converting the input of 3SAT to EPP's. Now, for each formula ψ in 3SAT problem we configure a workspace like Fig. 8, in which s and e are the leftmost and rightmost places, respectively. Considering a gate corresponding to each clause, the two first conditions of conversion would be satisfied. Because the robot have a path if at least an entrance in each gate is open. Also, the robot can migrate from s to e if there is no gate with all three entrances closed.

For satisfying the third condition, we associate all x_is, $1 \leqslant i \leqslant n$ together based on the LPGUM properties where n is the number of variables in formula ψ. In this sense, for example, all x_1s are dependent together and should take the same values.

Therefore, existing a path from s to e means that there exists at least one unobstructed entrance corresponding to a variable in each clause.

By assigning the true value to the variables whose entrance is open, we can find a true assignment in the corresponding formula ψ. On the other side, if the path from s to e does not exist, there is at least a clause like $C_j, 1 \leqslant j \leqslant n$ that all its corresponding entrances are obstructed. Then, there is no variable in C_j ables to take a true value. So, the formula ψ is not satisfiable.

Eventually, since the existence or non-existence of the path from s to e does not depend on the any metric $L_p, p \geq 1$, then we conclude that EPP is NP-hard for all such metrics. ■

Therefore, regarding the NP-hardness of EPP for any metric $L_p, p \geq 1$ through horizontal and vertical segment obstacles, the Min-SPP is also NP-hard under the above-mentioned conditions.

5.1 Approximation Algorithm for Min-SPP

We have a workspace which the endpoints of its imprecise segment obstacles are k-separable disks. We show the approximation algorithm in section 4.1 that members of approximate placement are selected at center of disks have $1 + \frac{2}{k}$ approximation factor for Min-SPP.

Theorem 6. *Consider a workspace such that the imprecise obstacles' endpoints are k-separable disks with $k > 0$. Now, in the approximation algorithm assuming the center of all disks as the placement $\mathcal{I}^{center}$ for Min-SPP, we have*

$$(1 + \frac{2}{k})\,\mathcal{L}(\mathcal{I}^{min}) \geq \mathcal{L}(\mathcal{I}^{center}) \tag{14}$$

Proof. We assume that $\mathcal{L}(\mathcal{I}^{center})$ and $SP(\mathcal{I}^{center})$ are introduced as in Definition 4. We define $SP(\mathcal{I}^{min})$ and $\mathcal{L}(\mathcal{I}^{min})$ as the shortest path and its length in the optimal placement for Min-SPP, respectively. Now, by introducing $SP'(\mathcal{I}^{center})$ as a path from s to e with the same topology as $SP(\mathcal{I}^{min})$ that selects the instances of $\mathcal{I}^{center}$ for $\mathcal{I}$, we will have:

$$\mathcal{L}(\mathcal{I}^{center}) \leqslant \mathcal{L}(SP'(\mathcal{I}^{center})) \tag{15}$$

Suppose two arbitrary disks along the path of $SP'(\mathcal{I}^{center})$ connected by e. As Fig. 7 shows, the length of this link is: $\mathcal{L}(e) = d + r_i + r_j$

Let e' be the link of D_i and D_j disks in $SP(\mathcal{I}^{min})$. Then we have: $\mathcal{L}(e') \geq d$

Therefore,

$$\begin{aligned}&\frac{\mathcal{L}(e')}{\mathcal{L}(e)} = \frac{\mathcal{L}(e')}{d + r_i + r_j} \geq \frac{d}{d + r_i + r_j}\\ &\Rightarrow \frac{d}{d + r_i + r_j} \geq \frac{k.r_{max}}{k.r_{max} + r_i + r_j} \geq \frac{k.r_{max}}{k.r_{max} + r_{max} + r_{max}} = \frac{k}{k+2}\end{aligned} \tag{16}$$

Eventually, for each pair of disks throughout $SP(\mathcal{I}^{center})$, Eq. (16) is valid. Thus,

$$\begin{aligned}&\frac{\mathcal{L}(\mathcal{I}^{min})}{\mathcal{L}(SP'(\mathcal{I}^{center}))} \geq \frac{k}{k+2}\\ &\Rightarrow \mathcal{L}(\mathcal{I}^{min}) \geq \frac{k}{k+2}\mathcal{L}(SP'(\mathcal{I}^{center}))\\ &\Rightarrow \mathcal{L}(\mathcal{I}^{min}) \geq \frac{k}{k+2}\mathcal{L}(SP'(\mathcal{I}^{center}))\\ &\Rightarrow \frac{k+2}{k}\mathcal{L}(\mathcal{I}^{min}) \geq \mathcal{L}(SP'(\mathcal{I}^{center}))\end{aligned} \tag{17}$$

And according to Eq. (15) and (17) we conclude that:

$$\mathcal{L}(\mathcal{I}^{center}) \leqslant \mathcal{L}(SP'(\mathcal{I}^{center})) \leqslant \frac{k+2}{k}\mathcal{L}(\mathcal{I}^{min}) \tag{18}$$

■

Analogous to Max-SPP, here we have proved that the approximation factor of this algorithm is $1 + \frac{2}{k}$. It obviously means that the farther the distance between disks, the closer the approximation factor to 1.

6 Conclusion

In this paper, we modeled the imprecise points by using some geometric approaches and proved that the Maximum Shortest Path Problem (Max-SPP) for a point robot is NP-hard and its decision version is NP-complete. For this proof, we considered the obstacles to be segments and their endpoints to be imprecise points modeled as segments. Remarkably, the obstacles can only be intersected at their endpoints. In addition, we presented an approximation algorithm with approximation factors of $1/2$ and $1 - \frac{2}{k+4}$ for disk and *k-separable* disk as imprecise points, respectively.

Furthermore, We studied the Minimum Shortest Path Problem (Min-SPP) in LPGUM. We proved that for a disk robot, Min-SPP is NP-hard for any metric $L_p\,, p \geq 1$, even when the imprecise regions are modeled as segments. Finally, we presented a parameterized approximation algorithm for this problem with approximation factor $1 + \frac{2}{k}$.

A possible future work includes the investigation of the hardness of Max-SPP and Min-SPP for different shapes which imprecise points could be modeled with.

References

1. Ahadi, A., Mozafari, A., Zarei, A.:Touring disjoint polygons problem is NP-hard. In Combinatorial Optimization and Applications (pp. 351-360). Springer International Publishing (2013)
2. Barraquand, Jerome, Bruno Langlois, and Jean-Claude Latombe. "Numerical potential field techniques for robot path planning." Systems, Man and Cybernetics, IEEE Transactions on 22.2 (1992): 224-241.
3. Choset, H., M., Ed.: Principles of robot motion: Theory, Algorithms, and Implementation MIT press (2005)
4. Davoodi, M., Mohades, A., Sheikhi, F., Khanteimouri, P.: Data Imprecision under λ-Geometry Model. Information Sciences, 126-144 (2015)
5. Davoodi, M., Khanteimouri, P., Sheikhi, F., Mohades, A.: Data Imprecision under λ-Geometry: Finding the Largest Axis-Aligned Bounding Box. In: 27th European Workshop on Computational Geometry, pp. 135-139 (2011)
6. Disser, Y., Mihalk, M., Montanari, S., Widmayer, P. (2014). Rectilinear Shortest Path and Rectilinear Minimum Spanning Tree with Neighborhoods. In Combinatorial Optimization (pp. 208-220). Springer International Publishing.
7. Disser, Y., Mihalk, M., Montanari (2015). Max Shortest Path for Imprecise Points. In EuroCG.
8. Dorrigiv, R., Fraser, R., He, M., Kamali, S., Kawamura, A., López-Ortiz, A., and Seco, D.: On Minimum and Maximum-weight Minimum Spanning Trees with Neighborhoods. In: Approximation and Online Algorithms. Springer Berlin Heidelberg, pp. 93-106 (2013)
9. Dror, M., Efrat, A., Lubiw, A., Mitchell, J. S. (2003, June). Touring a sequence of polygons. In Proceedings of the thirty-fifth annual ACM symposium on Theory of computing (pp. 473-482). ACM.
10. Joskowicz, L., Ostrovsky-Berman, Y., Myers, Y.: Efficient Representation and Computation of Geometric Uncertainty: The linear parametric model. Precision engineering, 34(1), 2-6 (2010)
11. Latombe, Jean-Claude. Robot motion planning. Vol. 124. Springer Science & Business Media, 2012.
12. LaValle, S., M.: Planning Algorithms. Cambridge university press (2006)
13. Li, F., Klette, R. (2007, March). Rubberband algorithms for solving various 2D or 3D shortest path problems. In Computing: Theory and Applications, 2007. ICCTA'07. International Conference on (pp. 9-19). IEEE.
14. Lichtenstein, D.: Planar Formulae and Their Uses. SIAM journal on computing, 11(2), 329-343 (1982)
15. Löffler, M., van Kreveld, M.: Largest Bounding Box, Smallest Diameter, and Related Problems on Imprecise Points. In: Algorithms and Data Structures, Springer Berlin Heidelberg, pp. 446-457 (2007)
16. Löffler, M.: Data Imprecision in Computational Geometry. PhD thesis, Utrecht University (2009)
17. Löffler, M., van Kreveld, M.: Largest and smallest convex hulls for imprecise points. Algorithmica, 56(2), 235-269 (2010)
18. Myers, Y., Joskowicz, L.: Uncertain Geometry with Dependencies. Proceedings of the 14th ACM symposium on solid and Physical Modeling, 159-164 (1998)
19. Myers, Y., and Joskowicz, L.: Point Distance and Orthogonal Range Problems with Dependent Geometric Uncertainties. In: 14th ACM Symposium on Solid and Physical Modeling, pp. 61-70. ACM (2010)

20. Myers, Y., Joskowicz, L.: Uncertain Lines and Circles with Dependencies. Computer-Aided Design, 45.2, 556-561 (1998)
21. O'Rourke, J.: Computational Geometry in C. Cambridge university press (1998)
22. Pan, X., Li, F., Klette, R. (2010). Approximate shortest path algorithms for sequences of pairwise disjoint simple polygons. In CCCG (pp. 175-178).
23. Surmann, H., Huser, J., Wehking, J.: Path Planning far a Fuzzy Controlled Autonomous Mobile Robot. In: Fifth IEEE International Conference on Fuzzy Systems, Vol. 3, pp. 1660-1665. IEEE (1996)