

Lower-bound Solutions to the Touring Regions Problem with Polygonal Obstacles and Disk-shaped Regions

Jindřiška Deckerová

Jan Faigl

Abstract—The paper studies lower bounds on the optimal solution of the shortest path connecting a sequence of regions with the presence of obstacles. For disk-shaped regions, a straightforward bound can be based on relaxing the obstacles and using Euclidean distance between the disks or the Second Order Cone Programming (SOCP). However, such lower bounds would be poor if regions are mutually close but reachable by relatively long detours, avoiding the obstacles. Therefore, we analyze the two-disk shortest path problem with three possible cases and use the path in lower bound estimation called TriCase. We show TriCase is a tighter lower bound than using Euclidean and SOCP-based approaches for specific instances. Based on the evaluation results, we propose to combine TriCase and SOCP approaches to determine the lower bound on the shortest path connecting disk-shaped regions in environments with polygonal obstacles. The lower bound is further employed in assessing the quality of a feasible sampling-based solution to the touring regions problem, supporting the viability of the proposed approach for routing problems with neighborhoods and the presence of polygonal obstacles.

I. INTRODUCTION

The studied problem is motivated by routing problems in the polygonal domain [1], [2] to connect regions by shortest path avoiding obstacles [3], [4], which is an underlying problem of various robotic applications [5]. In particular, we focus on finding a tight lower bound solution of the shortest path among obstacles connecting a given sequence of disk-shaped regions [6], modeling sensing ranges in data-collection tasks [7]. Since we study disk-shaped regions, we call the problem the *Touring Regions Problem with Obstacles* (TRPO) to distinguish it from the *Touring Polygons Problem* (TPP) [8] already studied for more than two decades. The TPP is to find the shortest path visiting a sequence of polygonal regions, and it is studied for instances without obstacles with several available approximation algorithms [9], [10], [11]. However, the problem is NP-hard [8], [12] in general.

The TPP is studied for specific types of regions [10], such as convex bodies [13]. A recently published work on disjoint disks [14] demonstrates that the TPP is actively studied from the foundational perspective. Besides, the problem formulation is motivated by practical applications of remote data collection [15], where further challenges are related to finding the shortest path among obstacles in the environment [16].

The authors are with the Dept. of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Technická 2, 166 27 Prague, Czechia. E-mails: {deckejin, faigl.j}@fel.cvut.cz.

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 22-05762S and by the European Union under the project ROBOPROX - Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590).

The *TPP with Obstacles* (TPPO) is addressed in [17] using a discretization schema with the gradient descent algorithm and the mixed-integer program with grid-partitioning to obtain locally optimal solutions and upper bounds on optimal solutions, respectively. The approach depends on the discretization density and grid size, and it scales poorly with the number of regions. A recent work on routing problems with obstacles is a solution of the *Hampered Traveling Salesman Problem with Neighborhoods* (H-TSPN) introduced in [6]. The H-TSPN is a routing problem with disk-shaped target regions and obstacles considered as a single segment. The problem is addressed by finding the shortest paths between two disk-shaped regions among segment barriers using a visibility graph. However, only instances with target regions that are not fully visible to each other are addressed, which allows the authors to assume disks and polygonal barriers are second-order cone representable [6].

In the present work, we propose to generalize the shortest path between two disk regions in environments with polygonal obstacles into three possible cases. The shortest path is employed to determine a lower-bound solution of the TRPO. Since it is based on three cases, as depicted in Fig. 1, the lower bound is referred to as TriCase. Lower bound estimates are essential for solving routing problems when using branch-and-bound techniques [18]. Therefore, we further relax obstacles and determine the lower bound using the Euclidean distance between the disks, regardless of the obstacles. Besides, the optimization model of the *Second Order Cone Programming* (SOCP) formulating the continuous problem is employed for the whole disk sequence without accounting for the obstacles.

Based on the empirical evaluation, the TriCase lower bound provides tighter bounds than the SOCP in scenarios with mutually close target regions but with the shortest path connecting them with significant detours to avoid obstacles. Hence, a combined lower bound is proposed for the quality assessment of feasible solutions to the TRPO determined by sampling-based methods. The evaluation results for solving test instances with and without overlapping regions demonstrate the viability of the presented approach. The contributions of the presented work are considered as follows.

- Analysis of the shortest path between two disk-shaped regions in the environment with polygonal obstacles.
- TriCase lower bound solution to the TRPO with disk regions based on the shortest path between two disks.
- The combined lower bound solution to the TRPO based on the proposed TriCase and SOCP formulation of the TRPO without obstacles.

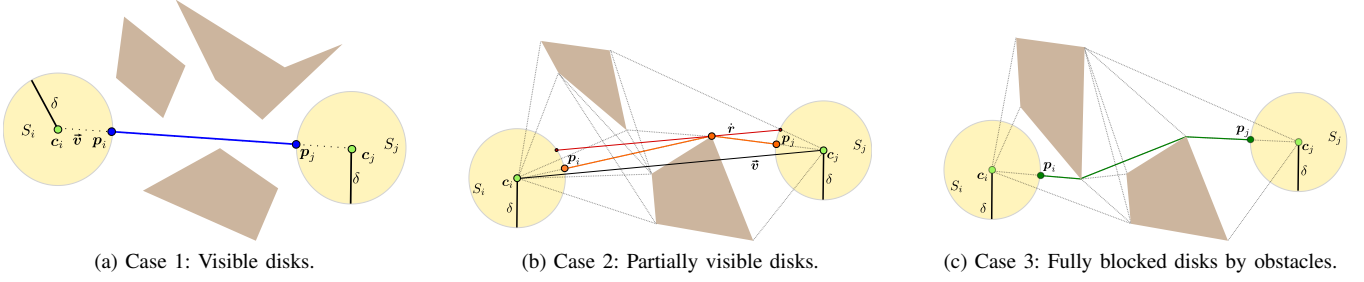


Fig. 1. Three cases of the solution to the shortest path between two disks in the presence of polygonal obstacles.

- Empirical evaluation on lower bound solutions and feasible solutions using sampling-based method.

The remainder of the paper is organized as follows. The studied TRPO is formally introduced in Section II. The analysis of the shortest paths between two disks is presented in Section III, and lower-bound methods are outlined in Section IV. Evaluation results are presented in Section V, and concluding remarks are in Section VI.

II. PROBLEM STATEMENT

The studied *Touring Regions Problem with Obstacles* (TRPO) is to find the collision-free shortest path visiting a sequence of n disk-shaped regions $\mathcal{S}$ in the presence of polygonal obstacles $\mathcal{O}$. Let $\mathcal{S}$ be $\mathcal{S} = (S_1, \dots, S_n)$, where the disk region S_i is centered at $c_i \in \mathbb{R}^2$ and has radius $\delta_i > 0$. Each j obstacle of m polygonal obstacles $\mathcal{O} = \{O_1, \dots, O_m\}$ is represented by l_j vertices, $O_j = \{o_j^1, \dots, o_j^{l_j}\}$, $o_j^l \in \mathbb{R}^2$, forming a polygonal obstacle. Let p_i and p_j be locations in $\mathbb{R}^2$, and $\pi(p_i, p_j)$ be the shortest path among obstacles from p_i to p_j , such as $\pi(p_i, p_j) \cap \mathcal{O} = \emptyset$. Then, $L(p_i, p_j) = \|\pi(p_i, p_j)\|$ denotes the length of the shortest path, and we denote $\|p_i - p_j\|$ for the Euclidean distance between the locations p_i and p_j .

The TRPO stands to determine a set of visiting locations $P = \{p_1, \dots, p_n\}$, where for each $S_i \in \mathcal{S}$, the visiting location p_i is within S_i , $\|p_i - c_i\| \leq \delta_i$, and the sum of lengths of the paths connecting two disks in the sequence $\pi(p_{i-1}, p_i)$, for $1 < i \leq n$, is the shortest possible. We denote the cost of the solution $L(\mathcal{S})$, and the TRPO can be formulated as the minimization Problem 1.

Problem 1 (Touring Regions Problem with Obstacles):

$$L(\mathcal{S}) = \min_P \sum_{i=2}^n \|\pi(p_{i-1}, p_i)\| \quad (1)$$

$$\text{s.t. } P = \{p_1, \dots, p_n\}, \text{ for } 1 \leq i \leq n \quad (2)$$

$$\|p_i - c_i\| \leq \delta_i, \text{ for } 1 \leq i \leq n \quad (3)$$

$$\pi(p_{i-1}, p_i) \cap \mathcal{O} = \emptyset, \text{ for } 1 < i \leq n. \quad (4)$$

A. Lower Bound Solution to the TRPO

We are primarily focused on the lower bound solution to the TRPO, which is assumed to be faster to compute than finding a feasible, eventually optimal, solution of the TRPO. It is motivated by searching for the best sequence regarding a solution to the H-TSPN and similar routing problems. Quick to compute lower bounds might help to reject unpromising sequences quicker than using heuristic feasible solutions [19]. Therefore, we are looking for tight lower bounds with low computational requirements.

The lower bound $LB(\mathcal{S})$ estimate does not necessarily correspond to a feasible solution. However, it is always lower or equal to the cost of the feasible (including optimal) solution $LB(\mathcal{S}) \leq L(\mathcal{S})$. We define $LB(\mathcal{S})$ as a sum of path lengths between the consecutive regions in the sequence $\mathcal{S}$.

Since the lower bound solution may be infeasible, we define two locations for each region S_i (except the starting and terminal ones): an exit location p_i^{out} and an entry location p_i^{in} . These represent the connecting path between two regions. For example, a path from S_{i-1} to S_i is from p_{i-1}^{out} to p_i^{in} . Further, we denote the lower bound path connecting two regions as $\pi_{\text{lb}}(p_i, p_j)$, which might be shorter than the shortest path between the regions S_i and S_j when obstacles are relaxed. Then, the lower bound cost estimate $LB(\mathcal{S})$ of the TRPO with $\mathcal{S}$ regions can be defined as

$$LB(\mathcal{S}) = \sum_{i=2}^n \|\pi_{\text{lb}}(p_{i-1}^{\text{out}}, p_i^{\text{in}})\|, \quad (5)$$

where $\|\cdot\|$ denotes the length of the path in the question.

Note that a straightforward lower bound can be computed using the Euclidean distance between the disks, reduced by the disks' radii, with relaxed obstacles. Besides, we can determine a single location $p_i^{\text{in}} = p_i^{\text{out}}$ using SOCP for a whole sequence of $\mathcal{S}$, again without considering the obstacles. Nevertheless, we intend to employ the shortest collision-free path among obstacles between two disks as the path π_{lb} , which is studied in the following section.

III. TWO-DISKS SHORTEST PATH IN THE PRESENCE OF POLYGONAL OBSTACLES

The shortest path between two disks S_i and S_j in the presence of polygonal obstacles is provided by analyses of three distinguished cases based on the presence of obstacles between them. We follow the idea of the shortest path presented in [6]; however, we explicitly accounted for three cases as depicted in Fig. 1. We further rely on available (and precomputed) visibility graph between the disks' centers instead of representing obstacles as barriers. It yields a relatively straightforward and easy-to-compute shortest path, which is not necessarily the case of the mixed integer SOCP in [6]. The three individual cases are analyzed in the following parts of the section.

A. Case 1: Visible Disks

We say that the disks S_i and S_j are visible if the disks' centers are visible. Thus, the shortest path is a line segment connecting the disks and is a part of the line segment connecting the disks' centers as depicted in Fig. 1a.

The disks' centers' are visible if the line segment $\vec{v} = (c_i, c_j)$ connecting them does not intersect any obstacle, which can be obtained from the visibility graph. Thus, there is no intersection of any obstacles' edges and $\vec{v}$. The visiting locations p_i and p_j to the disks can be determined as

$$p_i = c_i + \frac{\vec{v}}{\|\vec{v}\|} \delta, \quad p_j = c_j - \frac{\vec{v}}{\|\vec{v}\|} \delta. \quad (6)$$

The shortest path connecting the disks is $\pi = (p_i, p_j)$ and the path's length is $L(\pi(p_i, p_j)) = \|p_i - p_j\|$.

B. Case 2: Partially Visible Disks

Two disks S_i and S_j are partially visible if the disks' centers are not visible, and there exists at least one line segment directly connecting the disks; see Fig. 2. In this case, the shortest path π connecting S_i with S_j can be found using the visibility graph G from disks' centers, as in Fig. 1b.

Disks' centers are not visible if an obstacle intersects $\vec{v} = (c_i, c_j)$, which is the line segment connecting the disks' centers. If there exists at least one parallel line segment $\vec{r}$ to $\vec{v}$ that passes through any obstacle vertex and intersects both disks, the regions are partially visible. Then, the shortest path $\pi(c_i, c_j)$ can be found using the visibility graph as follows.

- 1) A visibility graph $G = (V, E)$ is constructed from all obstacles' vertices $o \in \mathcal{O}$ and disks' centers c_i and c_j .
- 2) The shortest path $\pi(c_i, c_j)$ is found in G using Dijkstra's algorithm and consists of c_i , c_j and a set of vertices $\hat{V} = \{o_1^*, \dots, o_k^*\}$, $k \geq 1$ that lies between the visiting locations to S_i and S_j .

We must determine the visiting locations p_i and p_j on the disks' borders to obtain the shortest path between the disks. Using vertices o_1^* and o_k^* , we determine visiting locations as

$$p_i = c_i + \delta \frac{o_1^* - c_i}{\|o_1^* - c_i\|}, \quad p_j = c_j + \delta \frac{o_k^* - c_j}{\|o_k^* - c_j\|}. \quad (7)$$

Note, that if $k = 1$ then $o_1^* = o_k^* = o^*$, which would be the case depicted in Fig. 2.

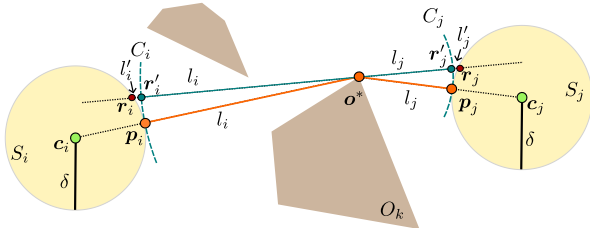


Fig. 2. Intersection points (blue) of two circles (depicted as the only arc in dashed blue) divide line segments into parts with the lengths l'_i , l_i , l_j , and l'_j , where $l'_i > 0$ and $l'_j > 0$.

The path $\pi(p_i, p_j)$ is formed from at least two connected line segments (p_i, o_1^*) , and (o_k^*, p_j) and the path between o_1^* , o_k^* , i.e., $\pi(o_1^*, o_k^*)$ found by search of G . The length of the path is denoted as $L(\pi(p_i, p_j)) = \|p_i - o_1^*\| + L(\pi(o_1^*, o_k^*)) + \|o_k^* - p_j\|$. The path is considered the shortest path since no shorter line segment directly connects the disks visiting the vertices $o_1^*, o_k^* \in \hat{V}$. So, for any other visiting location $r_i \in S_i$, $r_i \neq p_i$ there is no shorter path connecting r_i with any $o_1^* \in \hat{V}$ and any $o_k^* \in \hat{V}$ with any

$r_j \in S_j$, $r_j \neq p_j$ because $\|(r_i - o_1^*)\| + L(\pi(o_1^*, o_k^*)) + \|(o_k^* - r_j)\| \geq L(\pi(p_i, p_j))$.

Proof: The existence of the shortest path can be proved by constructing two circles, C_i and C_j . Let C_i be a circle centered at o_1^* and with the radius $l_i = \|p_i - o_1^*\|$, and C_j be a circle centered at o_k^* with the radius $l_j = \|p_j - o_k^*\|$. Thus, $L(\pi(p_i, p_j)) = l_i + l_j$. The intersection point of C_i and the line segment (r_i, o_1^*) is denoted as r'_i and the intersection point of C_j with (o_k^*, r_j) is $r'_j = C_j \cap (r_j, o_k^*)$. Then, we can express $\|(r_i - o_1^*)\| = \|r_i - r'_i\| + \|r'_i - o_1^*\| = \|r_i - r'_i\| + l_i$ and $\|(o_k^* - r_j)\| = \|o_k^* - r'_j\| + \|r'_j - r_j\| = l_j + \|r'_j - r_j\|$. Now, $\|(r_i - o_1^*)\| + \|(o_k^* - r_j)\| \leq L(\pi(p_i, p_j))$ and since $L(\pi(p_i, p_j)) = l_i + l_j$ we get

$$\|r_i - r'_i\| + l_i + l_j + \|r'_j - r_j\| \leq l_i + l_j$$

that can hold only if r'_i and r'_j intersect the respective disks that is in contradiction with $r_i \neq p_i$ and $r_j \neq p_j$. ■

C. Case 3: Fully Blocked Disks

The disk-to-disk visibility is fully blocked by obstacle if no parallel line segments $\vec{r}$ from the previous case can be constructed. Then, the shortest path π between the disks S_i and S_j can be determined using the visibility graph G , similar to Case 2. Locations p_i and p_j can be obtained as

$$p_i = c_i + \delta \frac{o_1^* - c_i}{\|o_1^* - c_i\|}, \quad p_j = c_j + \delta \frac{o_k^* - c_j}{\|o_k^* - c_j\|}, \quad (8)$$

where o_1^* and o_k^* is the obstacle vertex visited from c_i and c_j , respectively. The length of the path is $L(\pi(p_i, p_j)) = \|p_i - o_1^*\| + \sum_{l=2}^k \|o_{l-1}^* - o_l^*\| + \|o_k^* - p_j\|$, where k is the number of visited obstacle vertices.

Computational Complexity

In all three cases, we rely on an available visibility graph that can be, for n disks and M vertices of the obstacles, determined in $O((n + M)^2)$ using [20]. The query times can be improved for just two-point queries using triangular expansion, such as [21]. However, we further determine the shortest paths between the disks' centers using Dijkstra's algorithm run for each disk that can be bounded by $O(n(n + M)^2 \log(n + M))$. Then, the query for the shortest path between two disks is the access to the precomputed shortest paths between disks' centers that can be done in $O(1)$.

IV. LOWER BOUND METHODS

The shortest path between two disks is employed to determine the lower bound for the TRPO. Since the path is found based on three cases, we call the lower bound TriCase. Besides, we include two relatively straightforward lower bounds based on relaxing obstacles and using only Euclidean distances between the disks. The most straightforward is the usage of the Euclidean distances between the disks' centers in the Euclidean lower bound. Finally, we employ the SOCP model in the SOCP lower bound. All three methods are detailed in the following sections.

A. TriCase Lower Bound

TriCase lower bound is based on the shortest path between two disks. For each of two consecutive regions S_{i-1} and S_i , we determine the lower bound path π_{lb} as the shortest collision-free path between two locations $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$ found as a result of the shortest path. The location $\mathbf{p}_{i-1}$ corresponds to $\mathbf{p}_{i-1}^{\text{out}}$ of S_{i-1} and $\mathbf{p}_i$ corresponds to $\mathbf{p}_i^{\text{in}}$ of S_i . The lower bound $LB(\mathcal{S})$ is then computed directly using the paths' lengths in (5). Hence, for the precomputed shortest paths, the complexity can be bounded by $O(n)$. The lower bound is further referred to as LB_{Tri} .

B. Euclidean Lower Bound

For the Euclidean lower bound, we relax the obstacles and the lower bound path $\pi_{lb}(\mathbf{p}_{i-1}^{\text{out}}, \mathbf{p}_i^{\text{in}})$ between S_{i-1} and S_i lies on the line segment $\vec{v} = (\mathbf{c}_{i-1}, \mathbf{c}_i)$ connecting regions' centers $\mathbf{c}_{i-1}$ and $\mathbf{c}_i$. The locations $\mathbf{p}_{i-1}^{\text{out}}$ of S_{i-1} and $\mathbf{p}_i^{\text{in}}$ of S_i are determined as

$$\mathbf{p}_{i-1}^{\text{out}} = \mathbf{c}_{i-1} + \delta_{i-1} \frac{\vec{v}}{\|\vec{v}\|}, \quad \mathbf{p}_i^{\text{in}} = \mathbf{c}_i + \delta_i \frac{\vec{v}}{\|\vec{v}\|}. \quad (9)$$

The lower bound $LB(\mathcal{S})$ is then directly computed using (5) with the time complexity bounded by $O(n)$. The lower bound is further referred to as LB_{Euclid} .

C. SOCP Lower Bound

The SOCP lower bound is also based on the relaxed obstacles. The lower bound path π_{lb} is determined as the shortest connection of all regions in the sequence by solving the SOCP model. For details, we refer the readers to [18].

Since only one location $\mathbf{p}_i = (x_i, y_i)$ is determined for each S_i , the consecutive lower bound paths $\pi_{lb}(\mathbf{p}_{i-1}, \mathbf{p}_i)$ between two consecutive regions S_{i-1} and S_i are connected, and $\mathbf{p}_i = \mathbf{p}_i^{\text{in}} = \mathbf{p}_i^{\text{out}}$. Once the visiting locations are determined by the SOCP, the lower bound $LB(\mathcal{S})$ is directly determined using (5). Since the number of variables and second-order cone constraints depends only on n , the time complexity can be bounded by $O(n^3)$ [22]. The bound is further referred to as LB_{SOCP} . A single location $\mathbf{p}_i$ per region might yield tighter estimates of LB_{SOCP} compared to TriCase and LB_{Euclid} as it is independent of the radius δ_i .

V. RESULTS

The proposed lower bounds to the TRPO have been empirically evaluated in 32 randomly generated instances of the TRPO. The SOCP-based lower bounds with relaxed obstacles provide relatively tight bounds for instances with sequences where line segments can directly connect the consecutive disks in the sequence. Therefore, we propose using the combined lower bound based on the maximum of the provided estimates. The combined lower bound is then employed to assess the sampling-based solution of the TRPO, providing the relative optimality gap.

All methods are implemented in Python ver. 3.11, and the SOCP is solved using Gurobi solver [23]. The visibility graphs are computed using pyvisgraph library [24]. The evaluation is based on the following performance indicators.

- The relative lower bound $LB_{\text{rel}} = LB/LB_{\text{max}}$, where LB_{max} is the highest lower bound for the particular instance among the computed lower bounds, $LB_{\text{max}} = \max\{LB_{Tri}, LB_{SOCP}\}$.
- The relative gap $\%G = (L' - LB_{\text{max}})/LB_{\text{max}}$ 100% of a feasible solution with the length L' .
- The real computational times T reported in seconds.

For aggregated results on the TRPO instances with the same number of regions n , the indicators are reported as mean values with the standard deviations computed from multiple trials of the instances with the same n .

The rest of the results section is structured as follows. The used instances and their creation are detailed in Section V-A. Lower bound results are in Section V-B, and feasible solutions are reported in Section V-C. The results are further discussed in Section V-D.

A. TRPO Instances

Two types of evaluation of TRPO instances are used. The first instances are based on routing scenarios reported in [3] that include cases where regions are close by the Euclidean distance but reachable by longer shortest paths among obstacles. In particular, we chose polygonal scenarios *dense* with $n = 19$ regions, *jh* with $n = 22$, and *potholes* with $n = 63$. For each scenario, the sequence of visits to n regions $\mathcal{S}$ is determined using 2-Opt [25] for regions' centers and visibility graph.

The second type represents random instances with n regions selected from the set $n \in \{5, 10, 15, 20, 25, 28, 29, 30\}$. The instances are created in random environments with m obstacles $m \in \{12, 13, 14, 18, 20\}$ with up to four variants for the same m denoted as *random*_ $\{a, b, c, d\}$. Each random instance is generated as follows.

- An 100×100 large area is divided into 50 convex polygons using Voronoi diagram [26].
- From the generated polygons, in a random order, we select mutually not connected polygons that form the obstacle set $\mathcal{O}$.
- For each remaining polygon, we determine its center used as a center for the inscribed disk with the radius δ_i . If the disk does not intersect any obstacle $O \in \mathcal{O}$, the disk is added to $\mathcal{S}$.
- The sequence of regions is determined using disks' centers and visibility graph by 2-Opt [25], or random sequences are used.

An example of the instances is depicted Fig. 3.

B. Lower Bounds to the TRPO

First, we evaluate the lower bounds on the scenarios in *dense*, *jh*, and *potholes* polygonal environments. The best lower bound LB_{max} and individual lower bounds are depicted in Table I.

The results suggest the Euclidean distance is a relatively poor lower bound, as expected. However, depending on the shape of the environment, LB_{SOCP} might be a tighter lower bound than LB_{Tri} . It is noticeable for the *potholes* instance with relatively large open areas and dense regions, where

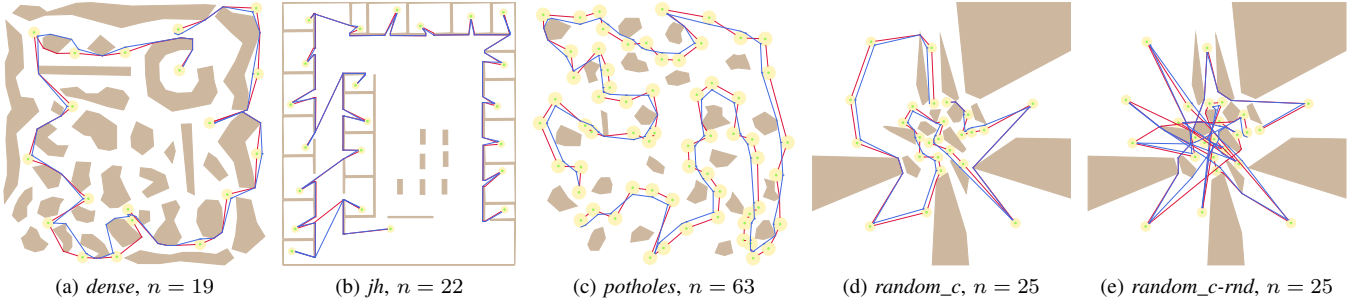


Fig. 3. Example of the TRPO instances with n disk regions in the polygonal environments. The environments *dense*, *jh*, and *potholes* are adopted from [3]. The *random_c* environment is a randomly generated TRPO instance using convex partitioning based on the Voronoi diagram. Each instance is depicted with a feasible solution obtained by *Sampling* ($k = 8$) (blue) and a lower bound solution (red) by TriCase. In the instances (a)–(d), the sequence is determined by 2-Opt using a visibility graph for disks' centers. The instance (e) is a random sequence.

TABLE I

LOWER BOUNDS FOR THE TRPO INSTANCES IN ENVIRONMENTS [3].

Instance	$LB_{\max}$	LB_{Euclid}		LB_{SOCP}		LB_{Tri}	
		LB_{rel}	$T_{\max}[s]$	LB_{rel}	$T_{\max}[s]$	LB_{rel}	$T_{\max}[s]$
<i>dense</i> , $n = 19$	7 703.23	0.85	< 1 ms	0.93	0.01	1.00	< 1 ms
<i>jh</i> , $n = 22$	10 224.31	0.69	< 1 ms	0.76	0.02	1.00	< 1 ms
<i>potholes</i> , $n = 63$	10 679.40	0.64	< 1 ms	1.00	0.03	0.66	< 1 ms

the error caused by passing obstacles is less significant than the reduction of LB_{Tri} caused by the addition of lengths of independent shortest paths between two consecutive regions. Since the computational requirements of LB_{Tri} are negligible when shortest paths are precomputed, the longer lower bound from LB_{Tri} and LB_{SOCP} can be used as the $LB_{\max}$.

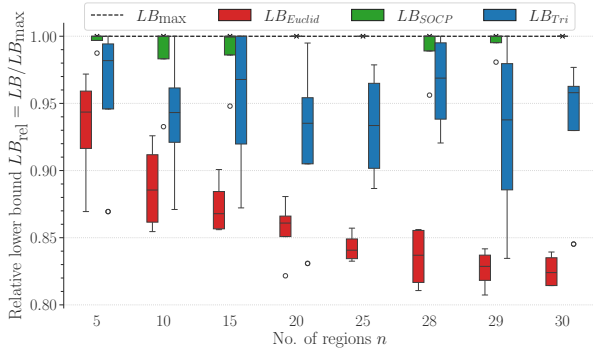


Fig. 4. Relative lower bounds LB_{rel} aggregated amount random TRPO instances with sequences found using visibility graph and 2-Opt.

The impact of the environment shape and specific sequences is studied for aggregated lower bound values computed for the random TRPO instances depicted as relative lower bounds in Figs. 4 and 5. The tightest lower bound $LB_{\max}$ defines the baseline value 1.0. The results suggest LB_{SOCP} and TriCase can vary significantly depending on the sequence. Therefore, the combined $LB_{\max}$ fits all cases.

The mean computational requirements are depicted in Fig. 6, where *VG - init* denotes the required time to determine the visibility graph.

C. Evaluation of Feasible Solution to the TRPO

The studied lower bound to the TRPO is employed in the evaluation of feasible solutions that are obtained by uniform sampling of each disk into k samples.

Hence, the feasible solution is denoted *Sampling* and solutions are found for $k \in \{4, 8, 16\}$. The visibility graph

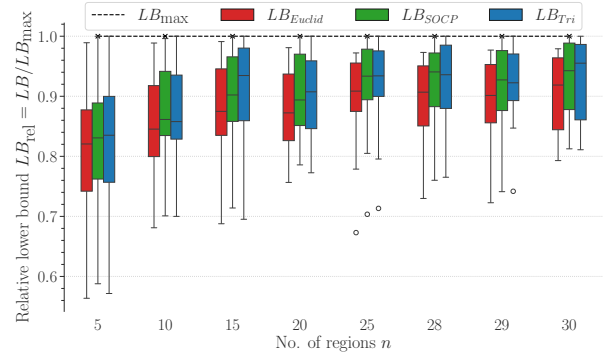


Fig. 5. Relative lower bounds LB_{rel} aggregated amount random TRPO instances with random sequences.

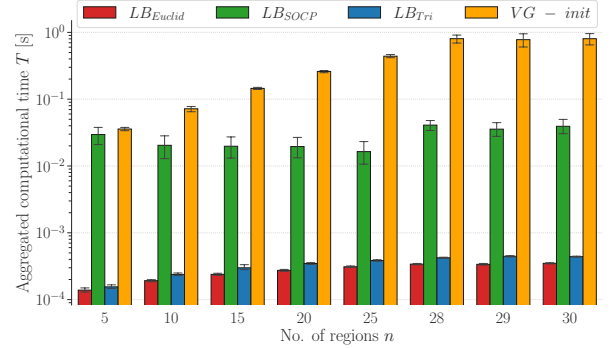


Fig. 6. Mean computational times T for random TRPO instances.

and shortest paths between the samples of two consecutive disks in the sequence are determined. Then, the feasible solution is found in the corresponding search graph using dynamic programming in $O(nk^2)$. Aggregated results using the relative gap are in Fig. 7 together with the computational requirements.

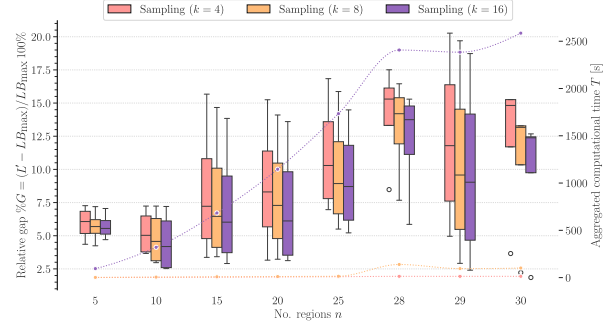


Fig. 7. Summary of the relative gap of the found solutions to the random TRPO instances to the $LB_{\max}$ depicted as the five-point summary.

D. Discussion of Results

The results suggest that the lower-bound methods' performance varies by instance. For sequences found using 2-Opt with straight-line-connectable disks, LB_{SOCP} is typically the tightest. On the other hand, LB_{Tri} provides better results in the created random TRPO instances. These two cases represent the late and early phases, respectively, of solving TSP-like problems in the polygonal domain, where the visit sequence is determined through combinatorial optimization.

However, the results on the feasible solutions in Fig. 7 indicate the lower bound is still relatively low compared to the feasible solution found using $k = 16$. The gap is in the range of 2.5 % up to about 20 %, which provides a ground for further research toward the optimal solution of the TRPO and other routing problems in the presence of obstacles.

All lower bounds are fast to compute. While pre-computing the visibility graph and inter-disk shortest paths adds some overhead, it remains minor compared to the overall cost of evaluating many sequences in TSP-like problems. The prototype Python implementation can be further optimized, e.g., using C++ visibility graph computation [21].

VI. CONCLUSION

We present the lower bound methods for the TRPO, using Euclidean distances and the newly proposed TriCase method based on the shortest path between disks among polygonal obstacles. Combining TriCase with SOCP-based bounds enables effective evaluation of feasible TRPO solutions, e.g., those found by the sampling-based methods. Results support using lower bounds in assessing TRPO sequences within branch-and-bound frameworks and reveal the poor scalability of sampling-based approaches as the number of samples k grows. It highlights the need for more efficient approaches, such as guiding sampling with lower bounds—similar to the *Iterative-Refined Inform Sampling* (IRIS) [27], [19], [28]. The proposed approach is also easily extendable to 3D with potential applications in the uncrewed aerial vehicles [29] and underwater exploration [30].

REFERENCES

- [1] M. Kulich, J. Vidašič, and J. Mikula, "On the travelling salesman problem with neighborhoods in a polygonal world," in *Robotics in Natural Settings*, 2023, pp. 334–345.
- [2] J. Janoš, V. Vonásek, and R. Pěnička, "Multi-goal path planning using multiple random trees," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4201–4208, 2021.
- [3] J. Faigl, V. Vonásek, and L. Přeučil, "Visiting convex regions in a polygonal map," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1070–1083, 2013.
- [4] L. Fanta, "The close enough travelling salesman problem in polygonal domain," Master's thesis, Czech Technical University in Prague, 2021. [Online]. Available: <https://hdl.handle.net/10467/96747>
- [5] D. Debnath, F. Vanegas, S. Boiteau, and F. Gonzalez, "An integrated geometric obstacle avoidance and genetic algorithm tsp model for uav path planning," *Drones*, vol. 8, no. 7, p. 302, 2024.
- [6] J. Puerto and C. Valverde, "The Hampered Travelling Salesman problem with Neighbourhoods," *Computers & Industrial Engineering*, vol. 188, p. 109889, 2024.
- [7] D. J. Gulczynski, J. W. Heath, and C. C. Price, "The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics," in *Perspectives in Operations Research: Papers in Honor of Saul Gass' 80th Birthday*. Springer US, 2006, pp. 271–283.
- [8] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell, "Touring a sequence of polygons," in *Thirty-Fifth Annual ACM Symposium on Theory of Computing*, 2003, pp. 473–482.
- [9] A. Mozafari and A. Zarei, "Touring Polygons: An Approximation Algorithm," in *Combinatorial Algorithms*, 2012, pp. 110–121.
- [10] A. Ahadi, A. Mozafari, and A. Zarei, "Touring a sequence of disjoint polygons: Complexity and extension," *Theoretical Computer Science*, vol. 556, pp. 45–54, 2014-10-30.
- [11] X. Tan and B. Jiang, "Efficient Algorithms for Touring a Sequence of Convex Polygons and Related Problems," in *Theory and Applications of Models of Computation*, 2017, pp. 614–627.
- [12] A. Ahadi, A. Mozafari, and A. Zarei, "Touring Disjoint Polygons Problem Is NP-Hard," in *Combinatorial Optimization and Applications*, 2013, pp. 351–360.
- [13] V. Polishchuk and J. S. Mitchell, "Touring Convex Bodies-A Conic Programming Solution," in *Canadian Conference on Computational Geometry*, 2005, pp. 290–293.
- [14] B. Qi, R. Qi, and X. Chen, "New Approximation Algorithms for Touring Regions," in *39th International Symposium on Computational Geometry (SoCG)*, 2023, pp. 54:1–54:16.
- [15] V. Krátký, P. Petráček, V. Spurný, and M. Saska, "Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2302–2309, 2020.
- [16] J. Deckerová, K. Kučerová, and J. Faigl, "On Improvement Heuristic to Solutions of the Close Enough Traveling Salesman Problem in Environments with Obstacles," in *European Conference on Mobile Robots (ECMR)*, 2023, pp. 1–6.
- [17] W. Ma, X. Liu, and X. Luo, "An Novel Method for Solving Touring-polygons Problem with Obstacles," 2023, Preprint.
- [18] W. P. Coutinho, R. Q. d. Nascimento, A. A. Pessoa, and A. Subramanian, "A branch-and-bound algorithm for the close-enough traveling salesman problem," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 752–765, 2016.
- [19] J. Faigl, P. Váňa, and J. Drchal, "Fast sequence rejection for multi-goal planning with dubins vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6773–6780.
- [20] M. H. Overmars and E. Welzl, "New methods for computing visibility graphs," in *Fourth Annual Symposium on Computational Geometry (SCG)*, 1988, p. 164–171.
- [21] J. Mikula, M. Kulich, and L. Přeučil, "Tfivis: Versatile, reliable, and high-performance tool for computing visibility in polygonal environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 10 503–10 510.
- [22] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001.
- [23] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2025, cited on 2025-01-17. [Online]. Available: <https://www.gurobi.com>
- [24] C. Reksten-Monsen *et al.*, "Pyvisgraph - python visibility graph," 2018, cited on 2025-01-17. [Online]. Available: <https://github.com/TaipanRex/pyvisgraph>
- [25] G. A. Croes, "A method for solving traveling-salesman problems," *Operations research*, vol. 6, no. 6, pp. 791–812, 1958.
- [26] S. Gillies *et al.*, "Shapely: manipulation and analysis of geometric objects," [toblerity.org](https://github.com/Toblerity/Shapely), 2007, cited on 2025-01-17. [Online]. Available: <https://github.com/Toblerity/Shapely>
- [27] J. Faigl, P. Váňa, M. Saska, T. Báča, and V. Spurný, "On solution of the dubins touring problem," in *European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–6.
- [28] O. Toman, "Toward optimal solution of the close enough traveling salesman problem using interval sampling," Master's thesis, Czech Technical University in Prague, 2025. [Online]. Available: <https://hdl.handle.net/10467/123210>
- [29] Q. Kuang, J. Wu, J. Pan, and B. Zhou, "Real-time uav path planning for autonomous urban scene reconstruction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1156–1162.
- [30] Z. Cooper-Baldock, S. R. Turnock, and K. Sammut, "Wake-informed 3D path planning for Autonomous Underwater Vehicles using A* and neural network approximations," *Ocean Engineering*, vol. 332, p. 121353, 2025.