On Improvement Heuristic to Solutions of the Close Enough Traveling Salesman Problem in Environments with Obstacles

Jindřiška Deckerová Kristýna Kučerová Jan Faigl

***Abstract*— In this paper, we present a novel improvement heuristic to address the Close Enough Traveling Salesman Problem in environments with obstacles (CETSP$_{\text{obs}}$). The CETSP$_{\text{obs}}$ is a variant of the Traveling Salesman Problem (TSP), where the goal is to find a sequence of visits to given disk-shaped regions together with the points of visits to the regions. We address challenging instances in a polygonal domain with polygonal obstacles, where the final path connecting the regions must be collision-free. We propose a novel `Post-Optimization` procedure using Mixed Integer Non-Linear Programming (MINLP) to improve existing heuristic solutions to the CETSP$_{\text{obs}}$. We deploy the method with existing heuristic solvers, and based on the presented evaluation results, the proposed `Post-Optimization` significantly improves the heuristic solutions of all examined solvers and makes them competitive regarding the solution quality. The statistical evaluation reveals that the sequence found using relatively sparse sampling of the disk regions yields the best solutions among the evaluated solvers. The results support the benefit of the proposed MINLP-based solution to the continuous optimization of the CETSP$_{\text{obs}}$.**

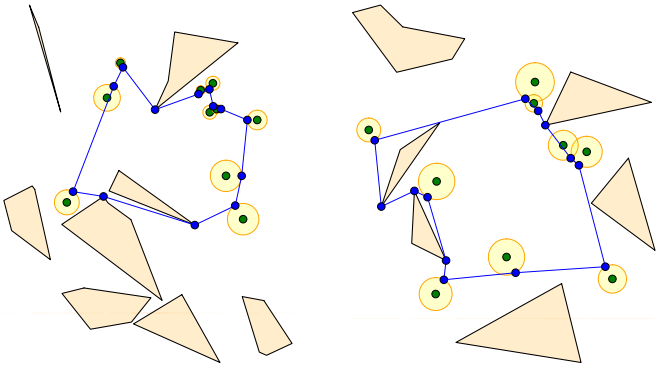
I. Introduction

The studied problem is motivated by *multi-goal path planning* [1] that is a robotic variant of the well-known combinatorial *Traveling Salesman Problem* (TSP) [2], where paths connecting the given set of locations are collision-free among possible obstacles in the environment. In the TSP, we search for a cost-efficient closed-loop tour visiting the locations, and we thus determine an optimal sequence of visits to the locations. Hence, the TSP represents a suitable problem formulation for various robotic sequencing tasks [3]. Furthermore, in remote data collection missions [4], [5], it is sufficient to visit a close region around the particular location and thus save the travel cost. In such scenarios, the TSP becomes the *TSP with Neighborhoods* (TSPN), where we need to determine the optimal sequence to visit the regions and also the optimal point of the visit to each region.

The neighborhoods in the TSPN can be represented as continuous regions [4], [6], [7], [8], or as clusters of regions [9], [10], [11], [12], or as clusters of locations [13]. In general, the TSPN is an APX-hard [14], and many heuristic approaches [15], [16], [17], [18], and approximation algorithms [19], [20], [14], [21] have been proposed. Further, the TSPN with disk-shaped neighborhoods has been introduced as the *Close Enough TSP* (CETSP) in [4]. Although exact methods have been proposed to solve the CETSP [22], [23],

The authors are with the Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic. {deckejin|kucerkr1|faiglj}@fel.cvut.cz

The presented work has been supported by the Czech Science Foundation (GAČR) under the research project No. 22-05762S.



(a) `t10_005` (b) `t9_005`

Fig. 1. Instances of the CETSP$_{\text{obs}}$ with found solutions depicted in blue.

they do not account for possible obstacles, and connections between the regions are only straight line segments with the length determined as the Euclidean distance between points of visits to the regions.

In this paper, we address the robotic variant of the CETSP in the polygonal domain with polygonal obstacles, further referred to as the CETSP$_{\text{obs}}$; see examples of instances in Fig. 1. When compared to the CETSP, the main challenge of the CETSP$_{\text{obs}}$ is determining collision-free paths between points of visits to the regions that can be arbitrarily located in the regions while finding the optimal sequence. Thus, the shortest paths connecting the regions need to be determined quickly, as many queries can be expected during the optimization of the sequence and points of visits to the regions. For the regular TSP with point locations or sampled regions to a discrete set of points, visibility graph can be constructed [24], [25] for shortest path queries; however, it is not the case of the CETSP$_{\text{obs}}$ with continuous regions.

Only two approaches explicitly address the CETSP$_{\text{obs}}$ (to the best of the authors' knowledge). The first is based on the shortest-path approximation employed in an unsupervised learning-based solution of the TSP in the polygonal domain [26], [27]. The second is the GLNSC [28] based on the decomposition of the CETSP$_{\text{obs}}$ to the continuous optimization of the CETSP and the point-to-point optimization using Delaunay triangulation. Besides, the discretized variant of the CETSP$_{\text{obs}}$ can be solved as the *Generalized TSP* (GTSP) [13] using pre-computed shortest paths among the obstacles and each sampled location of the regions. However, the optimal solution of such a discretized instance would be only the approximate solution of the original CETSP$_{\text{obs}}$ depending on the sampling density.

We propose to address approximations of existing so-

lutions to the CETSP$_{\text{obs}}$ by the `Post-Optimization` procedure that improves any existing heuristic solution. The procedure is based on formulating the problem as *Mixed Integer Non-Linear Programming* (MINLP) and exploits the given sequence of visits to the regions. We employed the procedure to existing solvers GLNSC [28] and unsupervised learning of the *Self-Organizing Map* (SOM) [27]. In addition, we adopted GTSP-based approach [18] to the *Generalized TSPN* (GTSPN), which first determines the sequence of visits to the regions' centers and then computes the points of visits using the local iterative optimization. Based on the empirical evaluation, the proposed `Post-Optimization` procedure improves solutions found by the existing solvers and makes the sampling-based GTSP the best-performing solver.

The rest of the paper is organized as follows. The CETSP$_{\text{obs}}$ is formally defined in Section II. The examined SOM and GTSP-based solvers are briefly described in Section III. The proposed `Post-Optimization` procedure is presented in Section IV. The results of the empirical evaluation are summarized in Section V, and the paper is concluded in Section VI.

II. Problem Statement

The studied CETSP$_{\text{obs}}$ is to find the shortest multi-point path that visits each of the n disk-shaped regions $\mathcal{S} = \{S_1, \ldots, S_n\}$ while avoiding m polygonal obstacles $\mathcal{O} = \{O_1, \ldots, O_m\}$. Each region $S_i \in \mathcal{S}$ is defined by its center $\boldsymbol{c}_i \in \mathbb{R}^2$, radius $\delta_i \geq 0$, and it is entirely inside the free space of the polygonal domain. A polygon obstacle $O_j \in \mathcal{O}$ is defined by a sequence of l_j vertices represented as points in $\mathbb{R}^2$, $O_j = (\boldsymbol{o}_j^1, \ldots, \boldsymbol{o}_j^{l_j})$, $\boldsymbol{o}_j^t \in \mathbb{R}^2$, for $1 \leq t \leq l_j$.

A solution of the CETSP$_{\text{obs}}$ is defined by a sequence Σ of visits to regions together with the points of visits $\mathcal{P}$ further referred to as *waypoints*. The final multi-point path is formed by a sequence of (shortest) paths among obstacles connecting $\mathcal{P}$ according to Σ. Hence, for the purpose of finding a path among obstacles connecting two waypoints, we consider a set of obstacles' points $\mathcal{Q}$ denoting the vertices of the obstacles' borders. Thus, the multi-point path is denoted $(\Sigma, \mathcal{P}, \mathcal{Q})$, where the terms can be defined as follows.

- Σ – *Sequence of visits* defining the order of visits to the regions: $\Sigma = (\sigma_1, \ldots, \sigma_n)$, $\sigma_i \neq \sigma_j$ for $i \neq j$.
- $\mathcal{P}$ – *Waypoints* are the points of visits to the regions: $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$, $\boldsymbol{p}_i \in \mathbb{R}^2$. For each waypoint $\boldsymbol{p}_i$, it holds $\|\boldsymbol{c}_i - \boldsymbol{p}_i\| \leq \delta_i$.
- $\mathcal{Q}$ – *Obstacles' points* forming the final path connecting $\mathcal{P}$ according to Σ, $\mathcal{Q} = \cup_{i=1}^{n} \{\boldsymbol{q}_i^0, \ldots, \boldsymbol{q}_i^{k_i}\}$, where $k_i \geq 0$ denotes the number of obstacles' points of the path connecting consecutive waypoints $\boldsymbol{p}_{\sigma_i}$ and $\boldsymbol{p}_{\sigma_{i+1}}$. Note that for a closed multi-point path, $\boldsymbol{p}_{\sigma_1}$ is the consecutive waypoint of $\boldsymbol{p}_{\sigma_n}$.

The length $\mathcal{L}^*$ of the path between two waypoints $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ can be defined according to the number of obstacles' points k_i. If the straight line connection of the waypoints is collision-free, the length is directly the Euclidean distance $\mathcal{L}^*(\boldsymbol{p}_i, \boldsymbol{p}_j) = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|$ and $k_i = 0$; for $k_i = 1$, it is

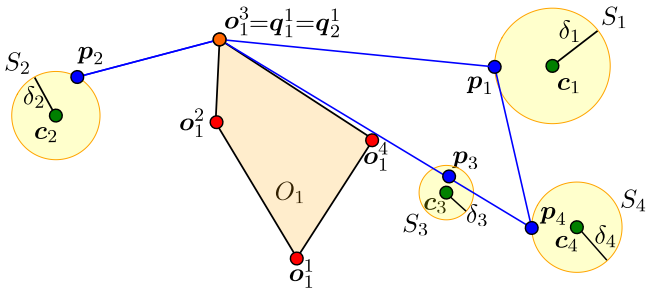


Fig. 2. A solution of the CETSP$_{\text{obs}}$ instance with $n = 4$ regions and one obstacle, $m = 1$. Regions' centers are small green disks. Vertices of the obstacles are small red disks, while obstacles' points $\mathcal{Q}$ are in orange. The determined waypoints $\mathcal{P}$ are visualized as small blue disks.

$$\mathcal{L}^*(\boldsymbol{p}_i, \boldsymbol{p}_j) = \|\boldsymbol{p}_i - \boldsymbol{q}_i^1\| + \|\boldsymbol{q}_i^1 - \boldsymbol{p}_j\|; \text{ otherwise}$$

$$\mathcal{L}^*(\boldsymbol{p}_i, \boldsymbol{p}_j) = \|\boldsymbol{p}_i - \boldsymbol{q}_i^1\| + \sum_{l=1}^{k_i - 1} \|\boldsymbol{q}_i^l - \boldsymbol{q}_i^{l+1}\| + \|\boldsymbol{q}_i^{k_i} - \boldsymbol{p}_j\|. \quad (1)$$

The used notation is visualized in an example of the solution instance in Fig. 2. The CETSP$_{\text{obs}}$ is formulated as the optimization problem in Problem 1.

Problem 1 (CETSP with polygonal domain (CETSP$_{\text{obs}}$)):

$$\mathcal{L}^* = \min_{\Sigma, \mathcal{P}, \mathcal{Q}} \mathcal{L}^*(\boldsymbol{p}_{\sigma_n}, \boldsymbol{p}_{\sigma_1}) + \sum_{i=1}^{n-1} \mathcal{L}^*(\boldsymbol{p}_{\sigma_i}, \boldsymbol{p}_{\sigma_{i+1}}) \quad (2)$$

s.t.

$$\Sigma = (\sigma_1, \ldots, \sigma_n), \sigma_i \neq \sigma_j \text{ if } i \neq j, 1 \leq \sigma_i \leq n, \quad (3)$$

$$\mathcal{P} = \{\boldsymbol{p}_{\sigma_1}, \ldots, \boldsymbol{p}_{\sigma_n}\}, \quad \boldsymbol{p}_i \in \mathbb{R}^2, \quad (4)$$

$$\|\boldsymbol{p}_{\sigma_i} - \boldsymbol{c}_{\sigma_i}\| \leq \delta_{\sigma_i} \quad \forall i \in \{1, \ldots, n\}. \quad (5)$$

III. Background

The studied CETSP$_{\text{obs}}$ is solved using existing heuristics and applying the proposed `Post-Optimization` procedure to their provided solution. In addition to the GLNSC [28] that directly solves the addressed CETSP$_{\text{obs}}$, an unsupervised learning approach has been proposed to solve the CETSP$_{\text{obs}}$ with polygonal regions in [27]. Besides, the GTSP-based approach [18] can be utilized to solve a discretized variant of the CETSP$_{\text{obs}}$. Therefore, the two additional methods are briefly overviewed with the relatively straightforward modifications for the CETSP$_{\text{obs}}$ to make the paper self-contained.

A. SOM-based Unsupervised Learning for the CETSP$_{\text{obs}}$

The unsupervised learning approach presented in [27] is based on the SOM for the TSP [29] and has been deployed in the polygonal domain in [26] using an approximation of the shortest path based on the underlying convex partitioning of the polygonal domain. Although there are multiple improvements of the SOM-based unsupervised learning for various routing problems, such as [30], [31], [32] and its generalization *Growing Self-Organizing Array* (GSOA) [17] deployed in [18], we directly utilize the available solver [27].

The unsupervised learning [27] is an iterative procedure in which a ring of $2n$ nodes (representing the multi-point path) is adapted to the regions during learning epochs. For each region, the closest node of the ring is determined as a winner node. Then, the winner node is adapted (moved) toward the region together with its neighboring nodes with the decreasing power of adaptation based on the neighboring function. The adaptation's power is controlled by the learning gain decreased every learning epoch to converge the ring to a stable solution. Note that the adaptation (movement) is along the shortest paths (or their approximation) among obstacles. Besides, the regions are examined in a random order in each epoch to avoid local minima [26].

After a finite number of epochs, the ring represents a multi-point path as each region has a unique winner node because of inhibition of the winners for each epoch [27]. Since the ring is represented as an array of nodes, the sequence of visits to the regions can be retrieved by traversing the ring. Besides, the winner node is associated with the point of the visit to the polygonal region.

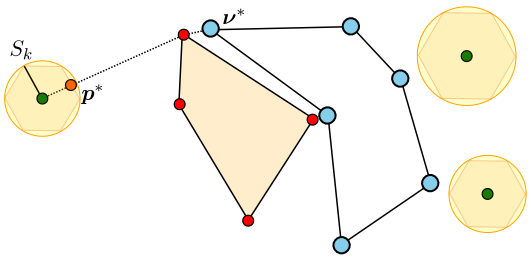


Fig. 3. Illustration of the winner node ν^* for the region S_k and its point of visit to the region $\boldsymbol{p}^*$ determined in the SOM solver [27]. The ring of nodes is represented as connected small blue disks.

The main modification of [27] for the herein addressed $\text{CETSP}_{\text{obs}}$ with disk-shaped regions is to represent each disk as the polygonal region with l vertices $\boldsymbol{p}_i^l$. However, unsupervised learning can still benefit from continuous regions. It is because the point of the visit to the region $\boldsymbol{p}^*$ is determined as the point on the region's boundary that intersects the shortest path between a node ν^* and disk's center, see Fig. 3. If the winner node is already inside the region, which can be caused by the adaptation of other nodes, its position is used as $\boldsymbol{p}^*$. The final multi-point path is retrieved by traversing the ring and connecting the associated points to the winner nodes. The reader is referred to [27] or [26] for further details on the utilized unsupervised learning.

B. GTSP-based Solver to the $\text{CETSP}_{\text{obs}}$

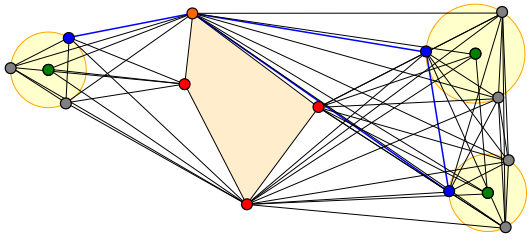


Fig. 4. An example of the GTSP-based solution of the $\text{CETSP}_{\text{obs}}$ using visibility graph. The found solution is depicted in blue.

The GTSP-based solver [18] has been proposed to solve a continuous variant of the GTSPN by discretization to the GTSP using regions' centers and deploying the heuristic GTSP solver [33] to determine the sequence of visits. Deploying the GTSP-based solver to the $\text{CETSP}_{\text{obs}}$ is straightforward. The disk-shaped regions are discretized into a finite set of samples on the disks' boundaries. Then, the visibility graph [34] is employed to determine the shortest paths between the samples; see Fig. 4. The following four steps summarize the usage of the GTSP-based solver [18].

- *Step 1.* Sample each region S_i into l samples Ξ on the region's border.
- *Step 2.* Construct visibility graph $\mathcal{G}$ in the polygonal domain for the samples Ξ.
- *Step 3.* Create an instance of the GTSP for the GTSP solver [33] using samples Ξ as a set of locations and the shortest paths between samples determined with $\mathcal{G}$ as the lengths between sets.
- *Step 4.* Use the GLKH solver [33] to find a sequence of visits and $\mathcal{G}$ to determine the solution $(\Sigma, \mathcal{P}, \mathcal{Q})$.

IV. Proposed Post-Optimization Procedure

The proposed `Post-Optimization` procedure is based on the MINLP mathematical model to find locally optimal solutions of the studied problem using the given sequence of visits Σ from some feasible solution $(\Sigma, \mathcal{P}, \mathcal{Q})$. The optimization idea is to minimize the path connecting the waypoints; however, we need to account for the obstacles' points through which a path among obstacles connects the waypoints. Therefore, in the MINLP model, we have two types of waypoints. The first waypoints are denoted $\mathcal{P}$, further also called the *disks' waypoints*, and are being optimized according to the problem statement in Section II. The second type of waypoints are the obstacles' points $\mathcal{Q}$, further referred to as the *obstacles' waypoints*.

We do not need to include all obstacles' points in the model, but only those connected with a region's waypoint by a straight line segment in the multi-point path. A connection between two consecutive obstacles' points (vertices) is guaranteed to be collision-free (e.g., using a visibility graph), and we do not change the topology of the multi-point path. Thus, depending on the number of obstacles' vertices of the path connecting two consecutive waypoints p_i and p_j, we add zero, one q_i^1 or two obstacles' waypoints q_i^1 and $q_i^{k_i}$ as defined in (1). Furthermore, if an obstacle's point (vertex) is included in two (or multiple) paths, such as the (orange) vertex in Fig. 4, the point is added to the model as the obstacle waypoint multiple times. Thus, the number of waypoints n' in the model can be $n' \geq n$.

Since all the waypoints have disk-shaped regions in the MINLP formulation, we consider zero disk's radius for obstacles' waypoints, and we get a sequence of regions $\mathcal{S}'$. Hence, a position of the waypoint with $\delta_i = 0$ is thus not effectively optimized in the MINLP solution. The model is summarized in Model 1 with the following variables.

- *Decision variables* $\boldsymbol{x} \in \mathbb{R}^{n' \times 2}$ represent the optimized waypoints $\mathcal{P}_{\text{opt}} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n'})$.

- *Auxiliary variables* $\boldsymbol{f} \in \mathbb{R}^{n'}$ and $\boldsymbol{w} \in \mathbb{R}^{n' \times 2}$ used to minimize the squared difference of two consecutive waypoints (6–8).
- Further *auxiliary variables* $\boldsymbol{v} \in \mathbb{R}^{n' \times 2}$ used to ensure that each waypoint $\boldsymbol{p}_i$ is within δ_i distance from the particular region's center $\boldsymbol{c}_i$ (9–10).

Model 1 (MINLP model):

$$\min_{\boldsymbol{x} \in \mathbb{R}^{n' \times 2}} \sum_{i=1}^{n'} f_i \tag{6}$$

s.t.

$$f_i^2 \geq \boldsymbol{w}_i^T \, \boldsymbol{w}_i, \qquad \forall i \in \{1, \ldots, n'\} \tag{7}$$

$$\boldsymbol{w}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i, \qquad \forall i \in \{1, \ldots, n'-1\} \tag{8}$$

$$\delta_i^2 \geq \boldsymbol{v}_i^T \boldsymbol{v}_i, \qquad \forall i \in \{1, \ldots, n'\} \tag{9}$$

$$\boldsymbol{v}_i = \boldsymbol{x}_i - \boldsymbol{c}_i, \qquad \forall i \in \{1, \ldots, n'\} \tag{10}$$

In solving the created Model 1, we aim to optimize the position of the disks' waypoints within the particular disk. However, the optimized position might yield a collision of the straight line segment connecting two consecutive waypoints (regions of $\mathcal{S}'$) and an obstacle. Therefore, three constraints are added if and only if there is an obstacle O_j between two consecutive regions of $\mathcal{S}'$ as follows.

The first constraint

$$\boldsymbol{d}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \tag{11}$$

uses *auxiliary variables* $\boldsymbol{d}_i \in \mathbb{R}^{n' \times 2}$ to express a straight line segment of two consecutive waypoints as the difference in coordinates. The second and third constraints are for l_j obstacle's vertices

$$-d_{i,2}\, o^l_{j,1} + d_{i,1}\, o^l_{j,2} + d_{i,2}\, x_{i,1} - d_{i,1}\, x_{i,2} \leq M\, y_{i,j} \tag{12}$$

and

$$-d_{i,2}\, o^l_{j,1} + d_{i,1}\, o^l_{j,2} + d_{i,2}\, x_{i,1} - d_{i,1}\, x_{i,2} \geq -M\,(1 - y_{i,j}) \tag{13}$$

for $1 \leq l \leq l_j$ to ensure that the straight line segment expressed as $\boldsymbol{d}_i$ does not intersects the obstacle O_j represented by a sequence of points $O_j = (\boldsymbol{o}^1_j, \ldots, \boldsymbol{o}^{l_j}_j)$. The constraints express that two waypoints are on the same half-plane. Therefore, only one of the constraints (12) or (13) is activated in the model. That is achieved by using the *Big-M* method (we use $M = 100\,000$) and *binary variables* $\boldsymbol{y} \in \{0,1\}^{n'}$ are used to activate the particular constraints.

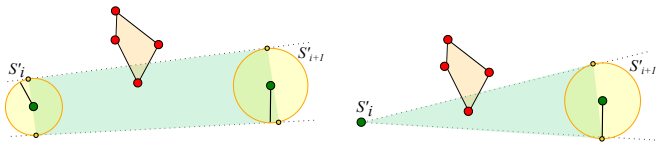
The proposed `Post-Optimization` is based on the MINLP model's construction, summarized in Algorithm 1. Adding constraints (11–13) corresponds to Lines 9, 11 and 12 of Algorithm 1, respectively. Note that the implementation of isObstacleBetween() depends on the type of the regions as a determination of an obstacle between two disk regions or between a disk and a point (disk region with zero radius for the obstacle's waypoint), as depicted in Fig. 5.

The proposed improvement procedure is a relatively straightforward adjustment of the waypoints within the disk-shaped regions. The procedure has been applied to the existing solutions of the CETSP$_{\text{obs}}$ and examined empirically. The results are reported in the following section.

Algorithm 1: `Post-Optimization` of the given CETSP$_{\text{obs}}$ solution (Σ, $\mathcal{P}$, $\mathcal{Q}$)

Input: $\mathcal{S} = \{S_1, \ldots, S_n\}$ – a set of the regions.
Input: $\mathcal{O} = \{O_1, \ldots, O_m\}$ – a set of the obstacles.
Input: (Σ, $\mathcal{P}$, $\mathcal{Q}$) – Σ is a sequence of visits to $\mathcal{S}$ with the corresponding waypoints $\mathcal{P}$ and obstacles' points $\mathcal{Q}$.
Output: (Σ, $\mathcal{P}_{\text{opt}}$, $\mathcal{Q}$) – Optimized solution.

1 $\mathcal{S}' \leftarrow ()$ `// Sequence ordered by` Σ
2 **for** σ_i `in` Σ **do**
3 $\mathcal{S}' \leftarrow \text{insert}(\mathcal{S}', S_{\sigma_i})$
4 **for** l `in` $0 : k_{\sigma_i}$ **do**
5 $\mathcal{S}' \leftarrow \text{insert}(\mathcal{S}', S(\boldsymbol{c} = \boldsymbol{q}^l_{\sigma_i}; \delta = 0))$ `// Insert obstacle's point as a new region with zero radius.`
6 $\mathcal{M} \leftarrow \text{createModel}(\mathcal{S}')$ `// According to Model 1 with decision variables` $\boldsymbol{x}$
7 **forall** consecutive regions $(S_i, S_{i+1}) \in \mathcal{S}'$ `and` $O_j \in \mathcal{O}$ **do**
8 **if** isObstacleBetween$((S_i, S_{i+1}), O_j)$ **then**
9 $\mathcal{M} \leftarrow \text{addConstraint}(\mathcal{M}, \boldsymbol{d}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i)$
10 **for** l `in` $1 : l_j$ **do**
11 $\mathcal{M} \leftarrow \text{addConstraint}(\mathcal{M}, -d_{i,2}\, o^l_{j,1} + d_{i,1}\, o^l_{j,2} + d_{i,2}\, x_{i,1} - d_{i,1}\, x_{i,2} \leq M\, y_{i,j}$
12 $\mathcal{M} \leftarrow \text{addConstraint}(\mathcal{M}, -d_{i,2}\, o^l_{j,1} + d_{i,1}\, o^l_{j,2} + d_{i,2}\, x_{i,1} - d_{i,1}\, x_{i,2} \geq -M\,(1 - y_{i,j}))$
13 $\mathcal{P}_{\text{opt}} \leftarrow \text{solveMINLP}(\mathcal{M})$ `// Extract the optimized disks' waypoints`
14 **return** $(\Sigma, \mathcal{P}_{\text{opt}}, \mathcal{Q})$



(a) Two disk regions.

(b) Disk with zero radius of the obstacle's waypoint and disk region.

Fig. 5. Visualization of the detecting obstacles between two consecutive regions. For two disk regions (left), the tangents are determined from the connection of the disks' centers. The disk has zero radius for the obstacle waypoint; thus, tangents are determined from the cone. Each obstacle's vertice must be on one side of the tangents, ensuring no obstacle between the regions.

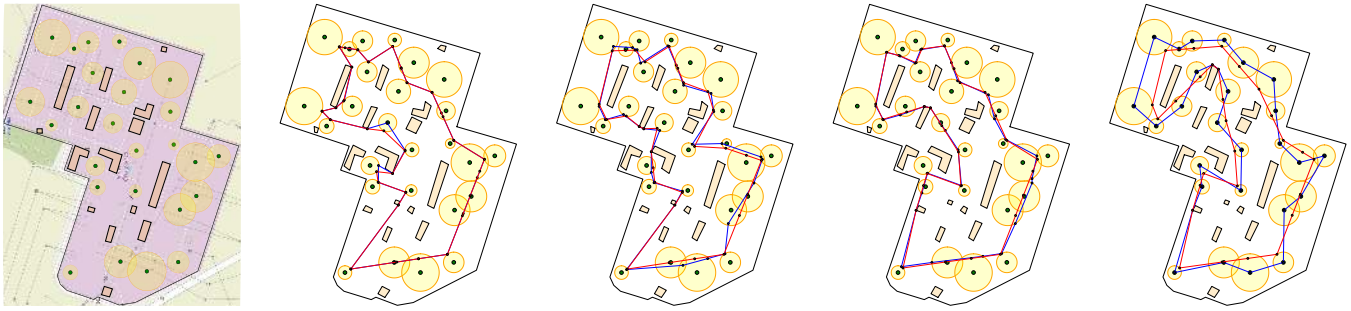
V. Empirical Evaluation

The proposed `Post-Optimization` procedure has been evaluated with the existing GLNSC [28], SOM [27], and GTSP [18] adjusted as described in Section III. All the methods are examined with and without the `Post-Optimization` procedure. The optimized solutions are denoted as GLNSC$^+$, SOM$^+$, and GTSP$^+$. The evaluation has been performed for a set of 32 randomly generated instances of the CETSP$_{\text{obs}}$, and one instance based on a real data collection scenario using a wheeled vehicle in an electrical substation depicted in Fig. 6. Each instance is named `tn_x`, where $n \in \{5, 6, 7, 8, 9, 10, 26\}$ denotes the number of regions, and x denotes the instance label, where $x \in \{1, \ldots, 6\}$ for $n = 5$ and $n = 8$, and $x \in \{1, \ldots, 5\}$ for each $n \in \{6, 7, 9, 10\}$. The SOM-based and GTSP-based solvers depend on the discretization l, selected to $l = 6$ providing the best trade-off between the computational requirements

TABLE I

PERFORMANCE INDICATORS OF THE EXAMINED $\text{CETSP}_{\text{obs}}$ SOLVERS INSTANCES AGGREGATED BY THE NUMBER OF NODES n.

Instance	GLNSC [28]		GLNSC$^+$		SOM ($l=6$) [27]		SOM$^+$ ($l=6$)		GTSP ($l=6$)		GTSP$^+$ ($l=6$)		GTSP ($l=1$)		GTSP$^+$ ($l=1$)	
	%PDB	%PDM	%PDB	%PDM	%PDB	%PDM	%PDB	%PDM	%PDB	%PDM	%PDB	%PDM	%PDB	%PDM	%PDB	%PDM
t5	**0.00**	32.75	**0.00**	**31.98**	1.66	35.49	0.08	32.91	1.37	33.98	**0.00**	32.67	15.38	51.96	**0.00**	33.13
t6	**0.00**	38.35	**0.00**	**38.33**	0.72	40.97	**0.00**	38.76	0.40	38.94	**0.00**	**38.33**	19.59	55.69	**0.00**	38.99
t7	**0.00**	48.91	**0.00**	48.91	0.44	51.91	**0.00**	49.51	0.25	49.39	**0.00**	**48.60**	15.99	66.04	0.16	48.91
t8	**0.00**	27.20	**0.00**	27.20	0.70	25.60	**0.00**	23.98	0.58	22.94	0.08	**22.53**	16.07	38.23	**0.00**	23.59
t9	**0.00**	15.74	**0.00**	15.74	1.67	17.73	**0.00**	16.15	0.96	16.08	**0.00**	**15.45**	18.11	30.40	1.15	16.84
t10	0.56	20.68	**0.00**	20.14	1.69	20.70	0.56	18.93	1.65	19.40	0.56	**18.48**	17.12	40.11	0.56	21.41
t26	**0.00**	1.58	**0.00**	**1.09**	1.95	4.46	**0.00**	2.74	2.46	2.46	1.89	1.89	30.89	30.89	8.47	21.92



(a) Map of an electrical substation with buildings as obstacles

(b) GLNSC, $\mathcal{L} = 1507.44$, $\mathcal{L}^+ = 1480.38$

(c) SOM ($l=6$), $\mathcal{L} = 1517.5$, $\mathcal{L}^+ = 1465.32$

(d) GTSP ($l=6$), $\mathcal{L} = 1470.69$, $\mathcal{L}^+ = 1462.57$

(e) GTSP ($l=1$), $\mathcal{L} = 1878.87$, $\mathcal{L}^+ = 1557.07$

Fig. 6. A map of an electrical substation utilized as an real-world $\text{CETSP}_{\text{obs}}$ instance `t26_001`. Solutions of this instance $\text{CETSP}_{\text{obs}}$ instance found by the particular solver are depicted in blue, and the optimized solutions by the `Post-Optimization` procedure are in red.

and solution quality among $l \in \{6, 12, 24, 64, 128\}$. Besides, we include the GTSP solver with $l = 1$ using the disks' centers in the evaluation to highlight the benefit of the proposed `Post-Optimization` to improve the solution quality even for sparse sampling.

The proposed `Post-Optimization` procedure and the GTSP solver are implemented in `Julia v1.7.` using `JuMP` and the MINLP solver `Juniper` [35]. The GLNSC and SOM are implemented in `C++`, and the GLNSC uses SOM-based initialization in a fast mode [28]. Each solver was executed for 20 trials on the `Intel i7-9700` processor running at 3 GHz, and two performance indicators are used for the evaluation. The *solution quality* %PDB for each instance is measured as the percentage deviation from the best overall solution $\mathcal{L}^*_{\text{best}}$ of the best solution $\mathcal{L}^*$ among all performed trials of the particular method $\%\text{PDB} = (\mathcal{L}^* - \mathcal{L}^*_{\text{best}})/\mathcal{L}^*_{\text{best}}\,100\%$. The *solution robustness* %PDM for each instance is measured as the percentage deviation from the best overall solution $\mathcal{L}^*_{\text{best}}$ of the mean solution value $\bar{\mathcal{L}}^*$ among all performed trials of the particular method $\%\text{PDM} = (\bar{\mathcal{L}}^* - \mathcal{L}^*_{\text{best}})/\mathcal{L}^*_{\text{best}}\,100\%$. Besides, we report the computational times T in milliseconds.

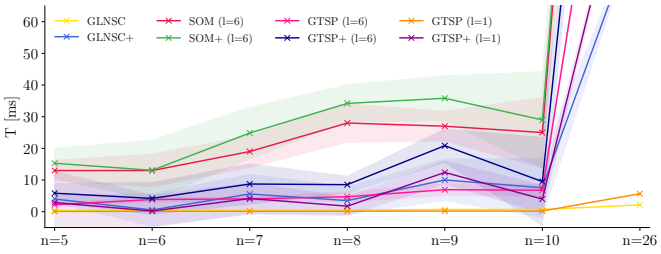


Fig. 7. Median computational times T aggregated from instances with the size n with standard deviation visualized as area around the medians.

The aggregated results are reported in Table I. The results support the expected improvement of the $\text{CETSP}_{\text{obs}}$ solutions and make the examined solvers competitive regarding the solution quality. Although the robustness varies and there is no clear winner, regarding the %PDB, the best-performing method is GLNSC, which can be considered the most complex algorithm. On the other hand, solutions of the very straightforward GTSP with $l = 1$, which can be solved as an instance of the TSP, are significantly improved by the proposed `Post-Optimization`.

The computational requirements of all solvers are exponential with n; see Fig. 7. Here, it is worth noting that the GLNSC method requires preprocessing the input instances by creating supporting structures, which is not included in the presented results, and similarly for the SOM-based solver. However, in both cases, the preprocessing time is competitive to the reported times, but the GLNSC becomes very demanding for larger instances.

TABLE II

STATISTICAL EVALUATION RESULTS OF THE $\text{CETSP}_{\text{obs}}$ SOLVERS.

a_1: GLNSC$^+$ a_2: GLNSC [28]	=	a_1: GLNSC [28] a_2: SOM ($l=6$) [27]	+	a_1: GLNSC [28] a_2: GTSP ($l=6$)	-	a_1: GLNSC [28] a_2: GTSP ($l=1$)	+
a_1: SOM$^+$ ($l=6$) a_2: GLNSC$^+$	=	a_1: SOM ($l=6$) [27] a_2: SOM$^+$ ($l=6$)	+	a_1: SOM ($l=6$) [27] a_2: GTSP ($l=6$)	-	a_1: SOM ($l=6$) [27] a_2: GTSP ($l=1$)	+
a_1: GTSP$^+$ ($l=6$) a_2: GLNSC$^+$	=	a_1: GTSP$^+$ ($l=6$) a_2: SOM$^+$ ($l=6$)	+	a_1: GTSP$^+$ ($l=6$) a_2: GTSP ($l=6$)	+	a_1: GTSP ($l=6$) a_2: GTSP ($l=1$)	+
a_1: GTSP$^+$ ($l=1$) a_2: GLNSC$^+$	-	a_1: GTSP$^+$ ($l=1$) a_2: SOM$^+$ ($l=6$)	-	a_1: GTSP$^+$ ($l=1$) a_2: GTSP$^+$ ($l=6$)	-	a_1: GTSP$^+$ ($l=1$) a_2: GTSP ($l=1$)	+

Symbols $+$, $-$, and $=$ denote the method a_1 provides statistically better, worse, or similar results than the method a_2, respectively.

We further report a statistical comparison of the solvers using the *Wilcoxon Signed Rank Test* [36], where the null hypothesis H_0 is that the solvers a_1 and a_2 provide solutions with statistically similar costs. H_0 is rejected if the obtained p-values are less than 0.001. In the statistical evaluation depicted in Table II, the symbol $=$ denotes a_1 performs similarly to a_2, or $+$ and $-$ if it performs better and worse, respectively, depending on the average solution cost. The results further support the statistically significant improvement of the solutions by the proposed `Post-Optimization` procedure. The GTSP-based solver with $l = 6$ performs best, and SOM is competitive with the GLNSC.

VI. Conclusion

We propose the `Post-Optimization` procedure to improve the heuristic solutions of the $\text{CETSP}_{\text{obs}}$. The procedure is based on the MINLP model to optimize the waypoints, and additional constraints are added to account for the polygonal obstacles. The procedure is employed with three solvers, the GLNSC, currently the only direct method to the $\text{CETSP}_{\text{obs}}$ with disk-shaped regions, and two existing heuristics straightforwardly modified for the disk-shaped regions. Based on the evaluation results, the proposed procedure improves all the found solutions and makes the methods competitive. Furthermore, based on a statistical comparison of the found solutions, the best-performing method is the GTSP with just six samples per each disk region. The sequence of visits to the disks is thus found on the discretized instance of the $\text{CETSP}_{\text{obs}}$, and the MINLP model enables finding the optimal solution of the continuous optimization part of the $\text{CETSP}_{\text{obs}}$ for that sequence. Hence, the proposed `Post-Optimization` represents a groundwork toward an optimal solution for finding the sequence using a branch-and-bound method, similar to the developed solvers to the CETSP without obstacles.

References

[1] S. Alatartsev, S. Stellmacher, and F. Ortmeier, "Robotic Task Sequencing Problem: A Survey," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 2, pp. 279–298, 2015.

[2] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ, USA: Princeton University Press, 2007.

[3] F. Suárez-Ruiz, T. S. Lembono, and Q.-C. Pham, "Robotsp – a fast solution to the robotic task sequencing problem," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1611–1616.

[4] 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*. Springer, 2006, pp. 271–283.

[5] 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.

[6] B. Yuan, M. Orlowska, and S. Sadiq, "On the Optimal Robot Routing Problem in Wireless Sensor Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 9, pp. 1252–1261, 2007.

[7] O. Tekdas, V. Isler, J. H. Lim, and A. Terzis, "Using mobile robots to harvest data from sensor fields," *IEEE Wireless Communications*, vol. 16, no. 1, pp. 22–28, 2009.

[8] M. Dunbabin and L. Marques, "Robots for Environmental Monitoring: Significant Advancements and Applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[9] I. Gentilini, F. Margot, and K. Shimada, "The travelling salesman problem with neighbourhoods: MINLP solution," *Optimization Methods and Software*, vol. 28, no. 2, pp. 364–378, 2013.

[10] K. Vicencio, B. Davis, and I. Gentilini, "Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2985–2990.

[11] A. Kovács, "Integrated task sequencing and path planning for robotic remote laser welding," *International Journal of Production Research*, vol. 54, no. 4, pp. 1210–1224, 2016.

[12] J. Deckerová, P. Váňa, and J. Faigl, "Combinatorial lower bounds for the generalized traveling salesman problem with neighborhoods," 2022, (in review).

[13] G. Laporte, A. Asef-Vaziri, and C. Sriskandarajah, "Some applications of the generalized travelling salesman problem," *Journal of the Operational Research Society*, vol. 47, no. 12, pp. 1461–1467, 1996.

[14] M. de Berg, J. Gudmundsson, M. J. Katz, C. Levcopoulos, M. H. Overmars, and A. F. van der Stappen, "TSP with neighborhoods of varying size," *Journal of Algorithms*, vol. 57, no. 1, pp. 22–36, 2005.

[15] W. K. Mennell, "Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem," Ph.D. dissertation, University of Maryland, 2009.

[16] S. Alatartsev, M. Augustine, and F. Ortmeier, "Constricting insertion heuristic for traveling salesman problem with neighborhoods," in *International Conference on Automated Planning and Scheduling (ICAPS)*, 2013, pp. 2–10.

[17] J. Faigl, "GSOA: growing self-organizing array - unsupervised learning for the close-enough traveling salesman problem and other routing problems," *Neurocomputing*, vol. 312, pp. 120–134, 2018.

[18] J. Faigl, P. Váňa, and J. Deckerová, "Fast heuristics for the 3-d multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2439–2446, 2019.

[19] A. Dumitrescu and J. S. B. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," *Journal of Algorithms*, vol. 48, no. 1, pp. 135–159, 2003.

[20] E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *Discrete Applied Mathematics*, vol. 55, no. 3, pp. 197–218, 1994.

[21] K. Elbassioni, A. V. Fishkin, and R. Sitters, "Approximation algorithms for the Euclidean traveling salesman problem with discrete and continuous neighborhoods," *International Journal of Computational Geometry & Applications*, vol. 19, no. 2, pp. 173–193, 2009.

[22] B. Behdani and J. C. Smith, "An integer-programming-based approach to the close-enough traveling salesman problem," *INFORMS Journal on Computing*, vol. 26, no. 3, pp. 415–432, 2014.

[23] 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.

[24] H. Huang and A. V. Savkin, "Viable path planning for data collection robots in a sensing field with obstacles," *Computer Communications*, vol. 111, pp. 84–96, 2017.

[25] B. Banyassady, M.-K. Chiu, M. Korman, W. Mulzer, A. Van Renssen, M. Roeloffzen, P. Seiferth, Y. Stein, B. Vogtenhuber, and M. Willert, "Routing in polygonal domains," *Computational Geometry*, vol. 87, p. 101593, 2020.

[26] J. Faigl, M. Kulich, V. Vonásek, and L. Přeučil, "An application of the self-organizing map in the non-euclidean traveling salesman problem," *Neurocomputing*, vol. 74, no. 5, pp. 671–679, 2011.

[27] 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.

[28] L. Fanta, "The close enough travelling salesman problem in the polygonal domain," Master's thesis, CTU in Prague, 2021.

[29] S. Somhom, A. Modares, and T. Enkawa, "A self-organising model for the travelling salesman problem," *Journal of the Operational Research Society*, pp. 919–928, 1997.

[30] J. Faigl, "On the performance of self-organizing maps for the non-euclidean traveling salesman problem in the polygonal domain," *Information Sciences*, vol. 181, pp. 4214–4229, 2011.

[31] J. Faigl and G. A. Hollinger, "Autonomous data collection using a self-organizing map," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1703–1715, 2018.

[32] J. Faigl, "Data collection path planning with spatially correlated measurements using growing self-organizing array," *Applied Soft Computing*, vol. 75, pp. 130–147, 2019.

[33] K. Helsgaun, "GLKH," 2013, [cited 29 Jun 2022]. [Online]. Available: http://akira.ruc.dk/~keld/research/GLKH/

[34] M. H. Overmars and E. Welzl, "New methods for computing visibility graphs," in *Symposium on Computational Geometry*. ACM, 1988, pp. 164–171.

[35] O. Kröger, C. Coffrin, H. Hijazi, and H. Nagarajan, "Juniper: An open-source nonlinear branch-and-bound solver in julia," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer International Publishing, 2018, pp. 377–386.

[36] A. Arcuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," in *International Conference on Software Engineering*, 2011, pp. 1–10.