Planning and Executing Optimal Non-Entangling Paths for Tethered Underwater Vehicles

Seth McCammon and Geoffrey A. Hollinger

***Abstract*— In this paper, we present a method to improve the navigation of tethered underwater vehicles by computing optimal paths that prevent their tethers from becoming entangled in obstacles. To accomplish this, we define the Non-Entangling Travelling Salesperson Problem (NE-TSP) as an extension of the Travelling Salesperson Problem with a non-entangling constraint. We compute the optimal solution to the NE-TSP by constructing a Mixed Integer Programming model, leveraging homotopy augmented graphs to plan an optimal trajectory through a set of inspection points, while maintaining a non-entangling guarantee. To avoid the computational expense of computing an optimal solution to the NE-TSP, we also introduce several methods to compute near-optimal solutions. In a set of simulated trials, our method was able to plan optimal non-entangling paths through a variety of environments. These results were then validated in a set of pool and field trials using a Seabotix vLBV300 underwater vehicle. The paths generated by our method were then compared to human-generated paths.**

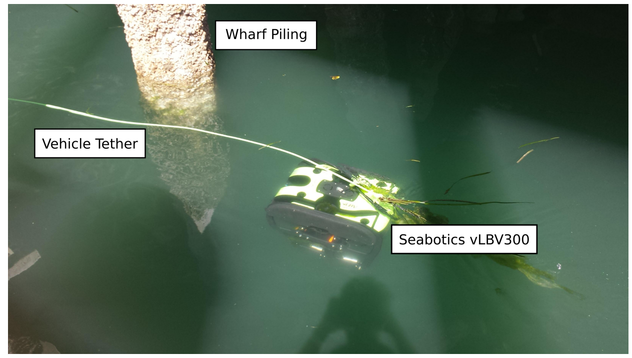


Fig. 1: Seabotix vLBV-300 Vehicle completing a wharf inspection in Newport, Oregon. Using our method, the robot plans a path to inspect the wharf's eight pilings in a non-entangling manner.

I. INTRODUCTION

Inspection of underwater devices is a time-consuming task, which either requires a human diver to physically inspect the device or a human controlled Remotely Operated Vehicle (ROV) to perform the inspection. This task can be made easier through the addition of autonomy to the ROV. Offshore energy production devices, such as Wave Energy Converters (WECs), need routine maintenance and inspection to prevent the buildup of marine life as well as mechanical wear and tear. WECs are often arranged in multi-device arrays to maximize their performance [1]. These environments can be complex, with many obstacles. Control of an ROV in such an environment can be difficult, and automating portions of it, such as navigation to an inspection site, would significantly ease the control burden on the operator.

Underwater inspection involves navigating a ROV or an Autonomous Underwater Vehicle (AUV) on a path which passes through a series of goal points and returns to the start. At each point of interest along the path, the robot may need to stop to make an observation or take a sample, as seen in Fig. 1. A tether connecting the robot to a continuous power supply can extend the mission duration of an AUV indefinitely [2]. The tether also provides a reliable communications link with a base station and a safety mechanism, preventing the robot from being lost at sea. However, a tether is not without drawbacks. Tethers limit the operational range of the robot, requiring them to stay within some distance of the base station. This range is further limited by the presence of obstacles, as the tether can become wrapped around them. In severe cases of entanglement, where the robot is unable to disentangle itself from the obstacle, it can even prevent the robot's recovery.

In this paper, we introduce the Non-Entangling Travelling Salesperson Problem (NE-TSP), which extends the traditional Travelling Salesperson Problem (TSP) by adding a non-entangling constraint. To solve the NE-TSP, we propose a novel method for planning optimal non-entangling paths by solving a Mixed Integer Programming (MIP) model. Since computing an optimal solution to the NE-TSP can be computationally expensive, we also introduce a heuristic method for reducing the search space of the MIP, enabling it to compute near-optimal paths more quickly.

The remainder of the paper is organized as follows. In Section II we examine prior work and the mathematical concept of a homotopy class. Section III formally defines the problem of computing a non-entangling path. We propose our solution to this problem in Section IV, as well as a heuristic for simplifying the robot's travel graph. Finally, in Section V, we compare our algorithm with a stochastic optimization method and a greedy approach as well as against a human in a set of field trials.

II. BACKGROUND

For autonomous underwater inspection tasks, close proximity to obstacles has led previous research to employ untethered AUVs to avoid the entanglement risk posed by a tethered vehicle [3]. Where a tether is used, existing research has focused on the length of the tether as the primary constraint in planning paths [4]. We consider the

This work was funded by Department of Energy grant DE-EE-0006816 and support from the W.M. Keck Foundation.

S. McCammon and G. Hollinger are with the Robotics Program of the School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR. (e-mail: {mccammos, geoff.hollinger}@oregonstate.edu)

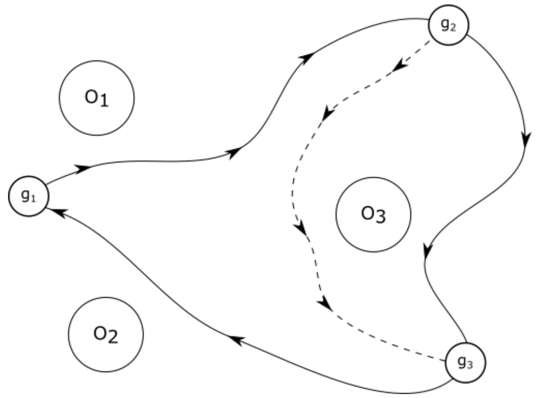


Fig. 2: An example of a trajectory modification which avoids tether entanglement. By modifying the path subsection between g_2 and g_3 (shown with a solid line) to the dotted line, the overall path length may be increased; however, the entanglement with O_3 is eliminated, as O_3 is no longer inside the bound of the trajectory.

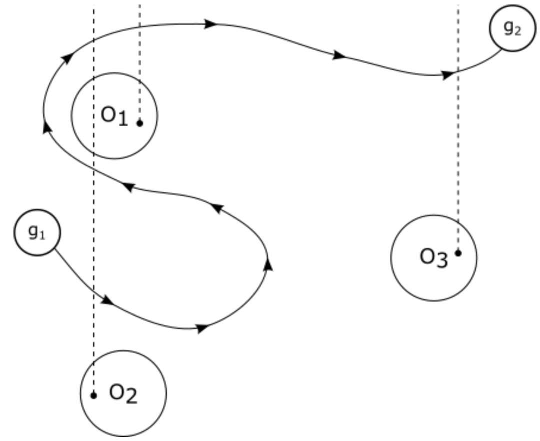


Fig. 3: Demonstration of h-signature Calculation. 1) Representative Points and their rays are constructed within obstacles O_1, O_2, and O_3. 2) Path between g_1 and g_2 is traced and intersections with rays from (1) are recorded. "O_2, O_2^{-1}, O_2, O_1, O_3" 3) h-signature reduced to "O_2, O_1, O_3".

additional constraint of avoiding entanglement as the robot plans through multiple goal points. While it is possible for the robot to reverse its path to avoid any entanglement, as was done in [5] to obtain maximum coverage by a tethered robot, such a behavior can lead to lengthy paths which reduce the overall area which can be inspected in a reasonable amount of time.

Planning for tethered robots has been framed as a topological planning problem, where the topology of the free space that the robot can plan through has been altered by the presence of obstacles. Multiple methods for planning in this space have been suggested, namely homotopy augmented graphs [6], [4], and filtrations of simplicial complexes [7].

In recent work, Mixed Integer Programming (MIP) has proven to be a flexible and powerful tool for computing optimal paths subject to constraints. MIP-based methods have been employed in orienteering-style extensions of the Travelling Salesperson Problem (TSP). In [8], the authors develop a MIP model to compute an optimal tour for a tourist collecting time-varying reward, while subject to a limited travel budget. MIP-based planning approaches have also been employed to plan optimal paths for autonomous aerial and marine vehicles while avoiding obstacles [9], [10].

In this paper, we build on the preliminary version of our method presented in our prior workshop paper [11]. Our method improves upon previous work by incorporating entanglement constraints into the planning process of a complete inspection tour. This enables us to plan tours which avoid entanglement while simultaneously minimizing total path length.

A. Homotopy Classes

A homotopy class describes a set of curves between two points. Two curves share the same homotopy class (i.e. are homotopic) if they share the same end points and one can be continuously deformed into another without encountering any obstacles. For example, in Fig. 2, the dotted and solid curves between g_2 and g_3 do not share the same homotopy class, although they do share the same endpoints, since the continuous deformation between them passes through O_3.

In order to characterize homotopy classes, Bhattacharya et al. [12] developed a descriptor, called an h-signature, which uniquely describes a homotopy class given a start and end point. The h-signature is computed by selecting representative points inside each obstacle, then drawing a parallel ray from each point. To determine the h-signature of a curve, the curve is traced, beginning at its start point. Each time the trace intersects one of the rays, a symbol corresponding to the ray and direction of intersection is added to the h-signature. This process is demonstrated in Fig. 3. A positive crossing of the ray emanating from the nth obstacle is considered to be from left to right, and is denoted as O_n. The inverse crossing, from right to left, is denoted as O_n^{-1}. The h-signature is then reduced by removing adjacent elements with opposite signs along the same ray. This process is repeated until no more elements can be removed. The resulting h-signature is a homotopy invariant which uniquely identifies the homotopy class of a curve.

III. PROBLEM DEFINITION

To plan non-entangling paths through the world, we need to first define an entanglement. Since a tour of goal points consists of a loop starting and ending at g_1, obstacles in the world may be divided into two sets: those contained within the bound of the tour (the interior set), and those outside the bound (the exterior set). Any obstacles in the interior set are considered to be entangled in the tether, while obstacles in the exterior set are non-entangled. This is illustrated in Fig. 2, where a path modification moves O_3 from the interior set to the exterior set. There exists a simple test for whether a given trajectory is entangled in any obstacles. We compute the h-signature of the trajectory by combining the h-signature of each of its sub-paths, and then reducing the combined h-signature as described in Section II-A. If the resulting h-signature is empty, the trajectory is non-entangling.

A. Non-Entangling Travelling Salesperson Problem

The problem of planning optimal non-entangling paths can be seen as an extension of the TSP, with the additional constraint that the path does not cause the tether to entangle any obstacles. This extension is the Non-Entangling Travelling Salesperson Problem.

An instance of the NE-TSP consists of a map of the world which contains n obstacles $O = \{o_1, o_2, ..., o_n\}$. To allow for application to many representative environments, each obstacle o_i is defined as a vertical projection from a circle on $\mathbb{R}^2$ to $\mathbb{R}^3$. The map also contains m goals $G = \{g_1, g_2, ..., g_m\}$ where $g_i \in \mathbb{R}^3$. The initial deployment point of the robot is also its first goal g_1. A trajectory T is a complete circuit of these goal points, ultimately returning to the initial deployment point after visiting the final point in T.

A solution to the NE-TSP consists of a trajectory T^* which satisfies

$$T^* = \underset{T}{\operatorname{argmin}}\{L_T | \textit{h-signature}(T) = \emptyset \text{ , } |T| = m\}, \quad (1)$$

where L_T is the length of the path needed to traverse all points in T and $|T|$ is the number of elements in T. T^* is the minimum-length, non-entangling trajectory which passes through all $g_i \in G$.

To compute the optimal solution to this problem, we propose a Mixed-Integer Programming model which can compute T^* given a set of goals G and obstacles O.

IV. ALGORITHM

A. Homotopy Augmented Graph

To guarantee the construction of non-entangling paths, we construct a homotopy augmented graph based at the robot's deployment point. Formulated by Bhattacharya et al. [12], a homotopy augmented graph allows the robot to plan the shortest path between two points in a given homotopy class. The homotopy augmented graph, $\mathcal{G}_{Aug} = \{V_{Aug}, E_{Aug}\}$, is constructed by augmenting a prior graph $\mathcal{G} = \{V, E\}$ with another dimension, h, which indicates the homotopy class of the path between a given vertex in $\mathcal{G}_{Aug}$ and the base point of the graph. Thus a vertex $v_i \in V_{aug}$ consists of the spatial location of v_i, as well as the homotopy class of the path between it and the base node. Using the vertices in $\mathcal{G}_{Aug}$, we can construct an augmented trajectory T_{Aug} which augments T with the homotopy classes of each of its elements.

We build on the idea of the homotopy augmented graph given in [12] by employing an extension of the Probabalistic Roadmap (PRM*) [13], [14] in place of the grid-based graph. PRMs provide a probabalistically complete graph-based map of an environment by taking a number of samples of the free space, and connecting nearby samples with traversable edges. PRM* extends the PRM by providing a principled method for determining which pairs of samples should be connected with edges. A more detailed explanation of PRM and PRM* can be found in [13]. Leveraging PRM* allows us to more easily span a 3-Dimensional environment, such as the underwater domain. However, since obstacles are projected from $\mathbb{R}^2$ to $\mathbb{R}^3$, we can compute entanglements and homotopy classes on the projection in $\mathbb{R}^2$, while distances between points and the resultant trajectory for the robot remain in $\mathbb{R}^3$.

By constructing the homotopy augmented graph using the AUV's deployment point as a starting point, we can ensure that each point in the graph is reachable by the robot. This is accomplished by only adding points to the graph which are within range of the robot's tether. Furthermore, during the construction of the homotopy augmented graph, we add a non-looping constraint, thus each path through the graph is both feasible and non-entangling.

Once constructed, paths can be planned over the homotopy-augmented graph using standard graph-based planning algorithms, such as A*, which are both complete and optimal. This, combined with the probabilistic completeness guarantees of PRM*, ensures that our proposed method retains the same probabilistic completeness guarantees as PRM*. Additionally, any paths generated on the homotopy augmented graph will be optimal with respect to the graph.

B. Mixed Integer Model

To compute the optimal non-entangling path for the robot over the PRM*, we model the NE-TSP with a Mixed-Integer Program. For each goal point $g_i \in G$, there is a corresponding set of all homotopy classes which reach the goal without exceeding the tether length constraint $\mathcal{H}_i = \{h_1, h_2, ...h_{z_i}\}$ with $z_i \geq 1$. Each $h_j \in \mathcal{H}_i$ corresponds to an augmented goal vertex $v_{g_i,h_j} \in V_{Aug}$. We define the set of homotopy-augmented goals V_h as the union of these vertices for all $g_i \in G$:

$$\forall g_i \in G, V_i = \{v_{g_i,h_1}, v_{g_i,h_2}, ..., v_{g_i,h_{z_i}}\},$$

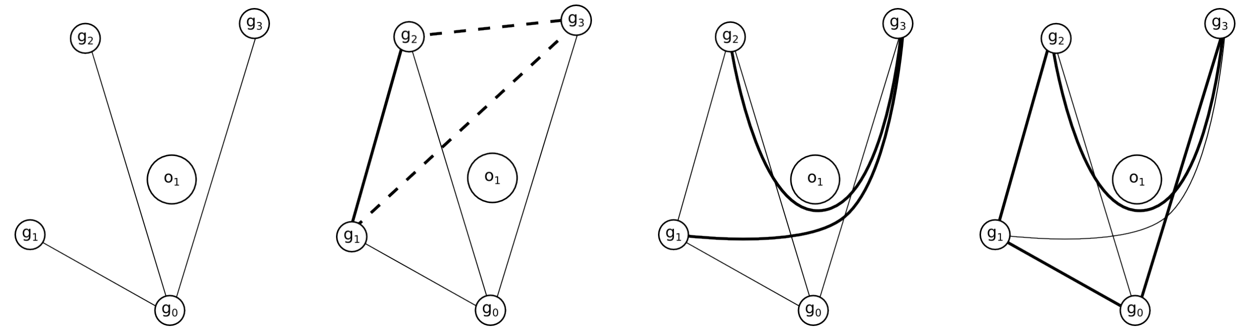
$$V_h = \cup_{i=1}^{m} V_i,$$

where $T_{Aug} \subseteq V_h$.

To fully define our MIP model, we need to determine the order that the goals are visited and by which homotopy class each goal is visited. We introduce two sets of binary decision variables, one to describe each of these two determinations. To solve for the homotopy class h_j of each goal, g_i, for each v_{g_i,h_j} let there be a corresponding $x_{i,j}$, where $x_{i,j} = 1$ if and only if v_{g_i,h_j} is visited by the robot and $x_{i,j} = 0$ otherwise.

The second portion of the solution, the order in which the goals are visited, is developed by determining which edges $e_{g_i,h_j,g_l,h_k} \in E_{Aug}$ are included in T. An edge e_{g_i,h_j,g_l,h_k} is the path segment between two vertices, v_{g_i,h_i} and v_{g_l,h_k}. Let $y_{i,j,k,l} = 1$ if and only if the robot travels the edge e_{g_i,h_j,g_l,h_k} and $y_{i,j,k,l} = 0$ otherwise. Each edge also has a corresponding distance d_{g_i,h_j,g_l,h_k} , which is the shortest-path distance in the homotopy-augmented graph between v_{g_i,h_i} and v_{g_l,h_k}.

To complete its tour of G, the robot will visit each $g_i \in G$ exactly once. Correspondingly, there is only one homotopy class at g_i that the robot will visit. This constraint can be



(a) Initial Graph Edges (b) Non-Entangling Graph Edges (c) Complete Graph (d) TSP Solution

Fig. 4: Heuristic Map Generation method. The shortest edges between the goals $\{g_1, g_2, g_3\}$ and the base point, g_0 are added to the map in 4a. In Fig. 4b, direct paths (shown in bold) are added, while indirect paths (dashed) are replaced with the bold paths shown in Fig. 4c. Finally, the TSP tour of the resulting map is shown in 4d.

modelled with the following summation:

$$\forall i, \sum_{j=1}^{z_i} x_{i,j} = 1. \quad (2)$$

In the classic TSP model, for each vertex in the graph, there is an incoming and outgoing edge. This can be captured in a MIP model with a simple degree-2 constraint which requires two unique edges to connect each vertex. However, in our model, this simple constraint fails when presented with the homotopy augmentation at each vertex. Since not every vertex v_{g_i,h_j} will be included in the final solution, a vertex may have either degree-0 or degree-2, depending on whether or not it is visited during the tour. Furthermore, the homotopy class of both the incoming and outgoing edges must be the same at v_{g_i,h_j} for the trajectory to be continuous. This is accomplished using the following constraint:

$$\forall i, \sum_{k,i\neq k}^{m} \sum_{j=1}^{z_i} \sum_{l=1}^{z_k} y_{i,j,k,l} \times x_{i,j} = 2. \quad (3)$$

Since only one $x_{i,j}$ for a given i is a part of the solution, as is given in the constraint shown in Equation 2, Equation 3 ensures that only edges which correspond with $x_{i,j}$ can be used. The first term enforces the degree-2 constraint, limiting the total number of edges connecting to a given node, while the second term ensures that any v_{g_i,h_j} can have either degree-2 or degree-0.

The final constraint in the MIP model eliminates subtours, ensuring that the solution consists of exactly one tour which visits each goal rather than multiple disjoint subtours. We implement this constraint as follows:

$$\sum_{i,k,i\neq k}^{m} \sum_{j=1}^{z_i} \sum_{l=1}^{z_k} y_{i,j,k,l} \leq |S \cap T_{Aug}|, \forall S \subset V_h, S \neq \emptyset, \quad (4)$$

Since it is impractical to compute all proper and nonempty subsets S of V_h, in practice the constraint modelled in Equation 4 is implemented using a lazy constraint. When a potentially valid solution is found, we determine whether it is a single tour, or multiple disjoint subtours. If the solution does contain subtours, a constraint is added disallowing the potential solution as a valid one.

To solve the MIP model, we employed the Gurobi Mixed Integer Solver [15]. The Gurobi solver utilizes a branch-and-bound method to converge to the optimal solution to any MIP. The solver also has the anytime property, meaning that at any point before convergence to the optimal solution, the incumbent solution (i.e. the best potential solution found) will be a valid, though sub-optimal, solution to the model. In Section V, we compare the optimal solution to the anytime solution generated after 2 minutes of computation. The optimality of our method is by construction. Equations 2 - 4 fully define the NE-TSP. As a result, the optimal solution on the homotopy augmented graph to these constraints also is the optimal solution to the NE-TSP.

C. Reduced MIP Heuristic

Solving the NE-TSP optimally requires a search over not only the combinatorial space of goal point orderings, but also the space of homotopy classes. To reduce this search space, we propose a heuristic which selects homotopy classes which are likely, though not guaranteed, to be a part of the optimal solution.

This is accomplished by creating a subgraph of the homotopy augmented-graph, the process of which is shown in Fig. 4. The graph is initialized with vertices at each $g_i \in G$. The shortest path to the base point, g_0, is computed, along with its corresponding h-signature, shown in Fig. 4a. Then, for each other pair of vertices, we attempt to construct a direct connecting edge if the direct edge shares a homotopy class with the path between the vertices which passes through g_0. If no such edge exists, shown by the dashed lines in Fig. 4b, an indirect edge which corresponds with the shortest path that does share the same homotopy class as the path through g_0 is added.

With this reduced graph, we remove the need for the MIP model to make a decision about which homotopy class to use for a given goal, reducing the overall search space the solver will have to search. The computation time is further reduced by eliminating a set of decision variables and their corresponding constraints. The constraint defined in Equation 2 is eliminated entirely, and the constraint defined by Equation 3 is reduced to a simple linear constraint. By computing the optimal TSP solution over the reduced graph, we compute a near-optimal solution to the NE-TSP.

The path that results from this method is still non-entangling, since no obstacles are contained within each loop in the graph. By only adding edges, direct or indirect, to the graph if they share a homotopy class with a known, non-entangling path, we guarantee that any complete tour in the graph will be non-entangling.

D. Simulated Annealing

The final method we examine to solve the NE-TSP is simulated annealing. Simulated annealing is a stochastic optimization algorithm, which performs search in multidimensional space and is robust to entrapment in local optima [16]. Initialized with some random state, x, at each iteration of the algorithm, a successor state x' is created by mutating x through some function. This successor state is compared to the previous state with an evaluation function. If the mutated state has the higher score, it becomes the new state. If it has a lower score, it becomes the new state with probability:

$$p = e^{-(s-s')/\theta}, \tag{5}$$

where s and s' are the scores of the state and mutated state, and θ is a scaling factor that decreases with subsequent iterations. To begin, x is initialized with a randomly generated T_{Aug}, which is a random ordering of the goal points and their corresponding homotopy classes.

During the mutation step of the simulated annealing process, the first point in this trajectory remains fixed, reflecting the assumption that the robot is tethered to a fixed base station. At each iteration of the optimization process, a trajectory can undergo one of the two types of mutations, chosen at random. The first of these, goal-swapping, swaps the order of two goals on the trajectory:

$$T' = \{g_1, ..., g_{i-1}, \mathbf{g_j}, g_{i+1}, ..., g_{j-1}, \mathbf{g_i}, g_{j+1}, ..., g_m\}.$$

This mutation can either raise or lower the overall trajectory length and entanglement of the path. To ensure that the resultant path is entanglement-free, we use a second method of mutation, path-inversion. During path-inversion, the homotopy class of one element in T_{aug} is changed. The result of a path-inversion mutation is shown in Fig. 2.

V. RESULTS

A. Simulations

We tested our method in a series of simulated tests, comparing the optimal MIP solution to the anytime MIP solution generated after 2 minutes of computation, the simulated annealing solution, and our heuristic solution. For each method aside from the optimal MIP solution, the computation time of the solution was limited to 2 minutes. We also compare to a greedy-backtracking method, in which the robot iteratively travels the closest point that does not violate the tether length constraint. Because of the behavior of the tether, this may not be the closest point in Euclidean space. After travelling to the final point, the robot returns to the start location by retracing its path. This ensures that the resulting tour will be non-entangling.

TABLE I: Comparison of computation time for optimal MIP solution. Each element contains average time to convergence and number of trials completed in 5 minutes (in parentheses).

	Tether Length			
# Goals	200	250	300	350
5	0.16 s, (10)	0.66 s, (10)	2.43 s, (10)	8.47 s, (10)
10	0.65 s, (10)	12.73 s, (10)	47.86 s, (7)	74.73 s, (8)
15	8.07 s, (8)	4.69 s, (2)	186.48 s, (3)	115.28 s, (3)
20	16.20 s, (9)	40.08 s, (2)	0.80 s, (1)	65.57 s, (1)

The results of these simulations are shown in Fig. 5. For each set of tether length and number of goal points, the methods were compared over 20 randomly generated worlds. In each of these worlds, up to 20 circular obstacles are randomly placed in an environment 500 units per side. The homotopy augmented graph is then constructed on a PRM* built with 1000 samples, taken uniformly over the non-obstructed space, with the base point randomly selected. The goal points were randomly selected from coordinates accessible within the homotopy augmented graph. Since each method requires the use of the homotopy augmented graph, the construction time of this graph is not included in the overall planning time. Simulations were done using Python on a Quad-Core Intel i7-2620M laptop processor clocked at 2.70GHz with 8GB of RAM.

In Fig. 5a to Fig. 5c we show a comparison of path lengths between the methods of computing a non-entangling path. In all three tests, the MIP-based methods (optimal MIP, MIP 2-Min anytime, reduced MIP heuristic) outperform simulated annealing and greedy methods. Both MIP 2-Min anytime and heuristic maintain an average path length within 5% of the optimal path length found by allowing the MIP solver to converge. As the number of goals and the tether length increase, performance across all 4 methods begin to degrade. However, it is apparent that the MIP-based methods maintain their level of performance in the larger environments far better than either the greedy or simulated annealing approach. Though initially close, as the environment gets large, the greedy method begins to outperform simulated annealing. This can be attributed to the restricting of simulated annealing to only 2 minutes of search. As the search space expands with more goal points and their corresponding homotopy classes, simulated annealing is able to explore proportionally less of that space, and so is less likely to find a short path.

B. Computational Performance

Computing the optimal solution to the Travelling Salesperson Problem, and, by extension, the NE-TSP is NP-Hard, meaning that no polynomial time algorithm to compute the exact solution exists (unless P = NP). The NE-TSP is made even more difficult, since the number of variables in our MIP model scales not only with the number of goal points, but also with the length of the tether and the number of

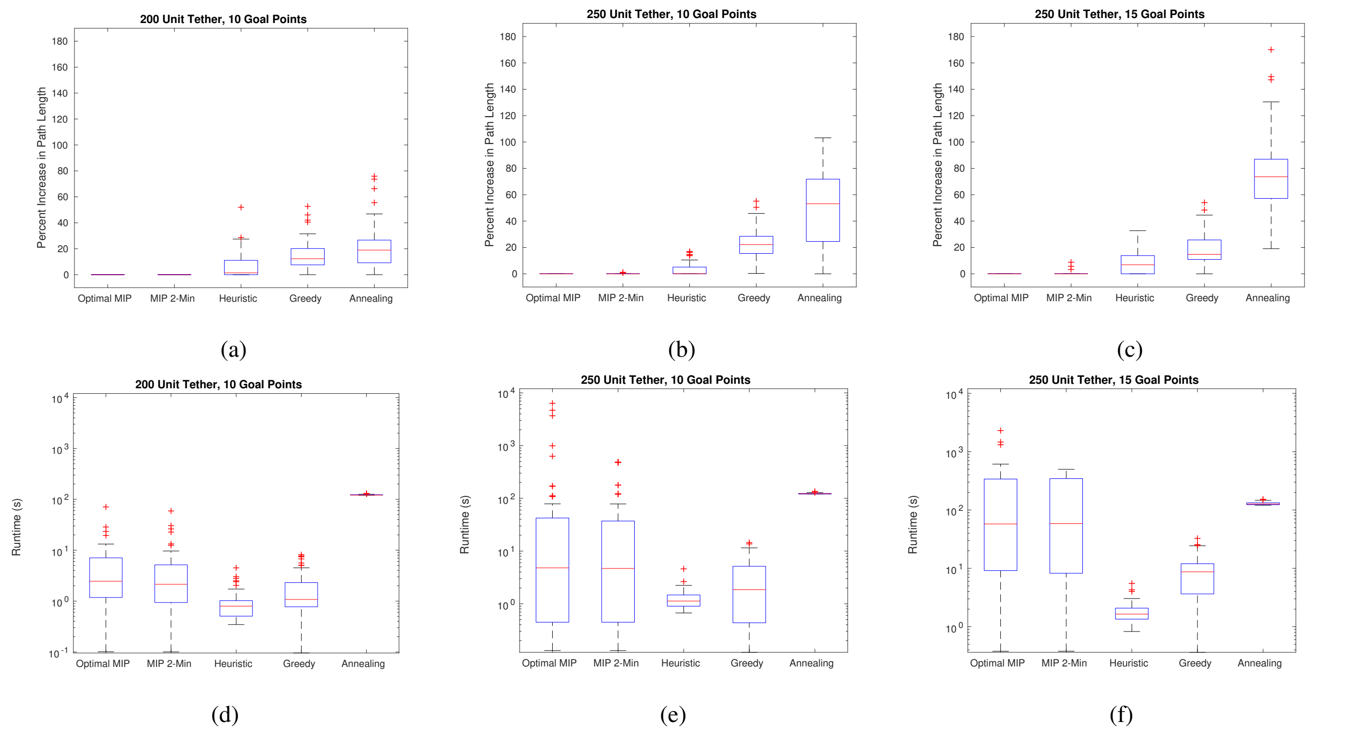


Fig. 5: Comparison of path length and computation time. Fig. 5a to Fig. 5c show the percent increase in path length over the optimal MIP length. Fig. 5d to Fig. 5f compare the computation time required to compute each solution. Note the logarithmic axes on Fig. 5d to Fig. 5f. The runtime shown here includes the time taken to compute a distance matrix between all potential goal points, and so can sometimes appear to slightly exceed 2 minutes of computation time.

obstacles in the environment. A longer tether and more obstacles allows the robot to reach the same goal point in more homotopy classes. The effects of this can be seen in Fig. 5d to Fig. 5f, where as the number of goal points and the length of the tether increase, the computation time for the optimal MIP solution increases exponentially. This is especially clear in Fig. 5f, where the longest computation time for the optimal MIP solution took over 12,000 seconds.

To evaluate the viability of our method in a practical application, we evaluated the computation time for problems with increasing numbers of goal points and with increasingly longer tethers. The results of this are shown in Table I. Each element in the table contains the average time to convergence for the optimal solution and the number (out of 10) of paths for which the MIP solver was able to compute the optimal solution (shown in parentheses). Computing the optimal solution remains feasible for shorter tethers (200 units) or smaller numbers of goal points (5-10). However, once the problem expands beyond this, an approximation should be used.

C. Pool and Field Tests

To show that the tether behaved as expected, we conducted a set of pool tests to ensure that the paths we generated were non-entangling when executed on a tethered vehicle. We implemented the non-entangling planner on a Seabotix vLBV-300 underwater vehicle [17] equipped with the Greensea INSpect GS3 Inertial Navigation System, a Teledyne Explorer DVL, and a Tritech Gemini multibeam sonar. The SeaBotix vehicle can be controlled via a series of waypoints provided through a Robotic Operating System (ROS) interface with a command station [18]. Using the non-entangling planner, the vehicle planned paths around a set of buoys, shown in Fig 6, and was able to successfully execute them without becoming entangled.

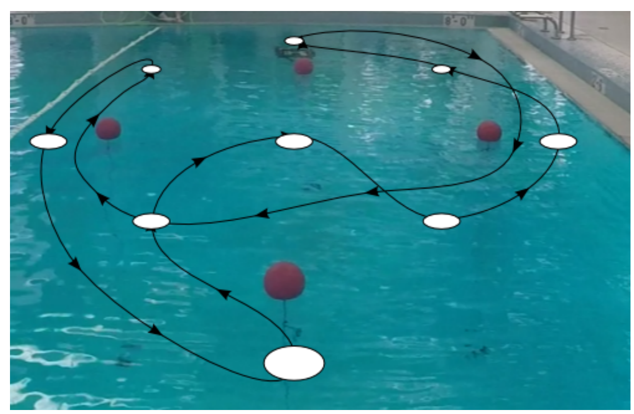
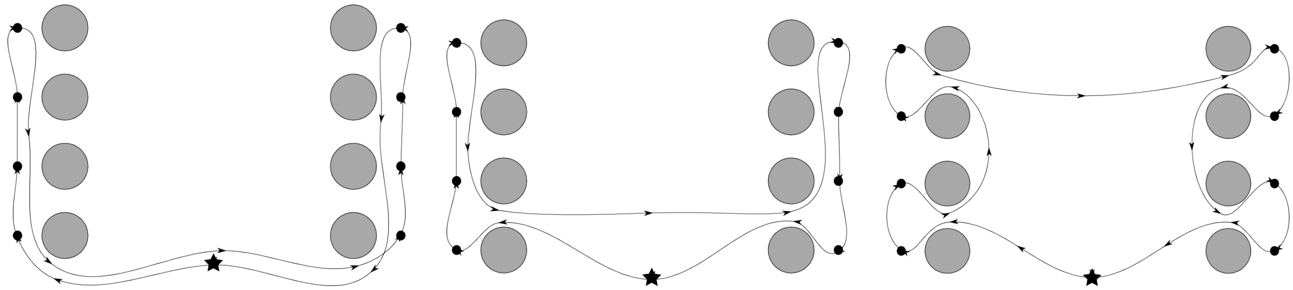


Fig. 6: An example obstacle and goal layout for a tethered vehicle. The white circles indicate goal locations (all of which lie on the water's surface). The red buoys act as obstacles and indicators of entanglement. The black line shows the planned path for the AUV, and the direction of travel along that path.

To represent an offshore inspection task, we deployed the vehicle from the Hatfield Marine Science Center wharf in Newport, Oregon and conducted an inspection of the wharf's pilings. The vehicle can be seen inspecting a piling in Fig. 1. While beneath the wharf, the vehicle was subject



(a) Systematic Path 1 (b) Systematic Path 2 (c) Optimal MIP Path

Fig. 7: Visualization of paths for dock inspection task. The robot is deployed from a base station located at the star, and must inspect the eight wharf pilings (gray circles) at the inspection locations (black circles). The endpoint of the tether is located at the base station.

TABLE II: Comparison of path lengths for hand-generated systematic paths and optimal MIP path for a wharf inspection task.

	Systematic Path 1	Systematic Path 2	MIP Path
Path Length	65.8 m	58.8 m	56.6 m

to a current of up to about 2 knots. The inspection task involved maneuvering to each of 8 pilings and pausing to inspect them. Two systematic paths were also hand-generated for non-entangling inspection tours of the same points. A comparison of the path lengths of each inspection tour are shown in Table II, and a visualization of the paths that the robot took is shown in Figs. 7a - 7c.

VI. CONCLUSION AND FUTURE WORK

In this work, we have introduced the Non-Entangling Travelling Salesperson Problem and have presented a Mixed Integer Programming model which can compute the optimal solution for a tethered robot. Leveraging homotopy augmented graphs, we can maintain a non-entangling guarantee on all paths generated. To improve computation time we developed a heuristic for selecting good homotopy classes for each goal point, reducing the search space needed by the MIP model to compute a near-optimal solution. We compare the optimal MIP solution with the anytime solution generated after 2 minutes with the MIP solver and our heuristic solution, as well as a simulated annealing and a greedy approach. To avoid the significant computational expense required to compute the optimal solution to the MIP Model, the 2-minute anytime solution and the heuristic solution were found to be close approximations of the shortest path. In field trials, we were able to plan short, non-entangling paths both in a pool environment and during a wharf inspection.

Future directions for this work include incorporating uncertainties into both the tether behavior and the model of the world. We plan to expand this work by examining dynamic obstacles, unknown obstacles, and disturbances to the tether.

REFERENCES

[1] C. Beels, P. Troch, J. Kofoed, P. Frigaard, J. Kringelum, P. Kromann, M. Donovan, J. Rouck, and G. Backer, "A methodology for production and cost assessment of a farm of wave energy converters," *Renewable Energy*, vol. 36, no. 12, pp. 3402 – 3416, 2011.

[2] J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, 2000.

[3] A. Kim and R. Eustice, "Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.*, pp. 1559–1565.

[4] S. Kim, S. Bhattacharya, and V. Kumar, "Path planning for a tethered mobile robot," in *Proc. IEEE International Conference on Robotics and Automation, 2014*, pp. 1132–1139.

[5] I. Shnaps and E. Rimon, "Online coverage by a tethered autonomous mobile robot in planar unknown environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 966–974, 2014.

[6] S. Bhattacharya, V. Kumar, and M. Likhachev, "Search-based path planning with homotopy class constraints," in *Proc. AAAI Conference on Artificial Intelligence*, 2010.

[7] F. T. Pokorny, M. Hawasly, and S. Ramamoorthy, "Topological trajectory classification with filtrations of simplicial complexes and persistent homology," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 204–223, 2016.

[8] J. Yu, J. Aslam, S. Karaman, and D. Rus, "Anytime planning of optimal schedules for a mobile sensing robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015*, pp. 5279–5286.

[9] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522–537, 2008.

[10] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. IEEE American Control Conference*, vol. 3, 2002, pp. 1936–1941.

[11] S. McCammon and G. Hollinger, "Planning non-entangling paths for tethered underwater robots using simulated annealing," in *Proc. Robot Learning and Planning Workshop, Robotics Science and Systems 2016*.

[12] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[14] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[15] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] *vLBV Operator's Manual*, Revision l ed., SeaBotix Inc., 2877 Historic Decatur Road STE 100, San Diego, CA 92106 USA, jun 2014.

[18] N. Lawrance, T. Somers, D. Jones, S. McCammon, and G. Hollinger, "Ocean deployment and testing of a semi-autonomous underwater vehicle," in *Proc. IEEE/MTS OCEANS Conference, Monterey, CA*, 2016, to appear.