Trajectory planning problem for detecting radioactive sources using underwater gliders

Ahmed Ibrahim
ahmed.ibrahim@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2023

Abstract

Radioactivity monitoring offers the opportunity to understand its impact on ocean ecosystems in various extreme locations, such as underwater volcanoes, seismic faults, or deep-ocean drilling locations. To expand radioactivity monitoring capabilities, the RAMONES project aims to develop lightweight, high-resolution, power-efficient radiation spectrometers integrated aboard autonomous underwater vehicles, namely underwater gliders, to perform in situ natural and artificial radioactivity measurements in the marine environment. The problem of detecting sources of radioactivity with one or more underwater gliders equipped with radiation spectrometers is a challenging one given the large search areas and the small detection distance of the sensors. To address this problem, in this thesis project, coverage algorithms are devised to generate a path so that gliders can determine the most efficient route to maximize the probability of detecting a radioactive source while avoiding obstacles and ensuring safe navigation through complex environments. The implemented algorithms are Travelling Salesman Problem (TSP), Minimum Spanning Tree (MST), and Optimal Control Problem (OCP). The latter considers the dynamics of the vehicle, that's why, modeling of the vehicle is considered.
Path planning simulations for each algorithm have been implemented using Matlab and an analysis has been done according to the processing time, uncovered areas, length of the traversed path, and time taken during each path. OCP is considered to be more useful if the time to traverse the path is constrained but has a much higher processing time than the other algorithms, and is less suitable when the objective is to cover the area, where a TSP algorithm fares better. Moreover, if the main goal is to obtain a solution fast, MST-based approaches yield smaller processing times.

Keywords: Radioactive sources detection, Autonomous underwater vehicles, Optimal Control, Covering Problem.

1. Introduction

Searching for such radioactive sources requires a trajectory generation algorithm to guarantee a high probability of finding radioactive sources. This can also lead us to an important concept which is motion planning or path planning. Geographic information systems, robotics, computer graphics, and other industries all encounter path-planning or motion-planning problems. The problem consists of finding an optimal route that avoids obstacles and achieves a certain goal. Normally, the path's quality is evaluated based on its Euclidean path length, smoothness, and obstacle-free status. In particular, the coverage problem is considered, which amounts to visiting every point in a defined area within a predefined distance.

Throughout this thesis, several covering algorithms are to be discussed and implemented based on the state-of-the-art planning methods and coverage problem mentioned in chapter 2. These methods can be classified into standard path planning methods e.g. TSP, MST, etc., and search Theory using Optimal Control Problems (OCP) as in the Koopman Search Theory[10]. Chapter 3 discusses the theory behind TSP, MST, and OCP. Also, it shows a framework for modeling basic search problems that were established as part of WWII antisubmarine warfare activities and have since been widely used in industrial and tactical applications [3]. Afterward, the mathematical difficulty of the Nomoto Steering Model3 is studied as well as the new numerical methods that are now available to solve it. These methods are then used to implement a high-dimensional search problem with many searchers, nonlinear searcher dynamics, and control restrictions.

Several implemented algorithms can be found in 4 associated with multiple simulations and graphs shown in 4, where a comparison between the aforementioned algorithms is done according to several key points e.g. Length of the traversed path and time taken during each path. For, the MST problem, 2 configurations were utilized which are the square configuration and the Hexagonal configuration to define the path traversed in each algorithm.

2. Literature Review

2.1. Search Theory Problem

Search theory [4] plays a crucial role in the field of robotics, offering a structured approach to solving problems related to navigation, decision-making, and resource management. One of its primary applications is path planning, particularly for autonomous robots. By employing search theory, robots can determine the most efficient route to maximize the probability of detecting a determined object while avoiding obstacles and ensuring safe navigation through complex environments.

Moreover, search theory enables the optimization of resources in robotics. Robots often have limited resources, such as energy and computational power. Through the principles of search theory, they can plan their actions to minimize energy consumption and maximize operational time. This is of paramount importance in scenarios where robots are deployed in remote or hostile environments and must operate efficiently with constrained resources.

Furthermore, search theory is useful in exploration and mapping activities. When robots are entrusted with mapping unknown or partially known settings, this theory guides their exploration, allowing them to efficiently map the surroundings and uncover new areas or places of interest. It is essential in applications such as search and rescue, environmental monitoring, and archaeological exploration.

Starting with [4], which covers recent advances in pursuit-evasion and autonomous search that apply to mobile robotics applications. Karaman & Frazzoli (2011) [9] discuss sampling-based algorithms for optimal motion planning in robotics. They present algorithms such as Probabilistic Roadmaps (PRM) and Rapidly-exploring Random Trees (RRT) that have been shown to work well in practice and have theoretical guarantees of probabilistic completeness. Masakuna and Fukuda Masakuna et al. (2019) [13] address the problem of search by a group of solitary robots. They propose a coordinated search strategy for self-interested robots without prior knowledge about each other. The results demonstrate the effectiveness of the strategy in achieving efficient search. Ismail & Hamami (2021) [8] conduct a systematic literature review on swarm robotics strategies applied to the target search problem with environment constraints. The review identifies various strategies and highlights their applicability in different research domains, including computational and swarm intelligence.

2.2. Coverage Problem

For several convincing reasons, the coverage problem is extremely important in the realm of robotics. First and foremost, it is critical to efficiently use resources, particularly when resources such as time, energy, or materials are limited. To properly complete their tasks, robots must make well-informed decisions on how to navigate and cover an area. Several researchers consider the covering problem as part of the motion planning problem.

Another critical component related to the coverage issue is time efficiency. Identifying the most time-efficient methods for area coverage is critical in search and rescue missions, inspection activities, and surveillance, especially when time is of the essence. Furthermore, in mobile robotic systems, energy is frequently a valuable and restricted resource. Efficient coverage tactics are critical for reducing energy consumption, prolonging the robot's operational time, and improving overall efficacy.

The coverage problem guarantees that sensors mounted on robots acquire information equally across the entire region, lowering the danger of missing key data points in data-collecting applications such as environmental monitoring and exploration. In other cases, such as floor cleaning robots or crop harvesting equipment, complete coverage is required because missing any part of the area is undesirable.

Furthermore, coverage methods are frequently included in path planning. They aid in the generation of pathways that not only cover the area effectively but also take obstacle avoidance and safe navigation into account, especially in dynamic contexts. The coverage problem has several practical applications, including autonomous lawnmowers, vacuum cleaners, agricultural robots, and autonomous cars, all of which require good area coverage to complete their tasks.

The coverage problem in robotics refers to the task of ensuring that one or more robots visit each point in a target area at least once. Several papers have addressed this problem and proposed different approaches and algorithms.

Batalin & Sukhatme (2004) [2] defined the cover-

age problem as the maximization of the total area covered by a robot's sensors. They discussed the exploration and deployment of a mobile robot in a communication network and proposed algorithms for efficient coverage.

Galceran & Carreras (2013) [5] conducted a survey on coverage path planning for robotics. They presented a collection of algorithms for complete coverage path planning using a team of mobile robots in unknown environments. They discussed various approaches, including boustrophedon decomposition and cellular decomposition.
Algorithms mentioned by Batalin & Sukhatme (2004) [2] and Le et al. [12] **improved** the efficiency of coverage path planning, however resource constraints and coverage optimization were not studied, which was done by Strimel & Veloso (2014) [14].

2.3. Minimum Spanning Tree

An MST is a graph tree given by a subset of edges that connects all nodes while minimizing the overall sum of edge weights. In robotics, these nodes can represent critical sites that the robot must visit, and the MST acts as a roadmap for linking these spots effectively. Robots can cross the MST's edges to visit all essential places while traveling the least amount of distance.[6].

2.4. Traveling Salesman Problem

An alternative casting for the coverage problem can implemented through the Travelling Salesman Problem (TSP) [1].The TSP belongs to the class of combinatorial optimization problems, where the objective is to find the shortest single path that, given a list of cities (or nodes) and distances between them, visits all the cities only once. Here is an example solution of the TSP in figure 1.

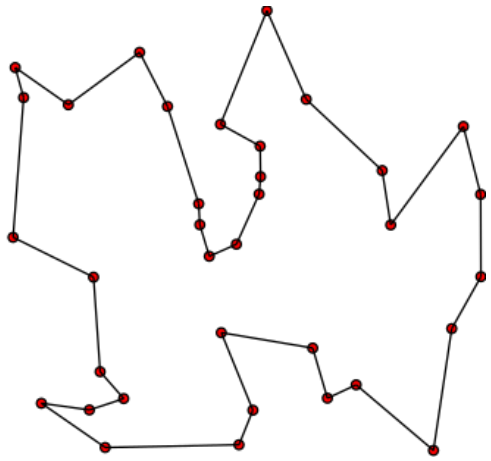


Figure 1: Solution of a traveling salesman problem: the black line shows the shortest possible loop that connects every red dot.

2.5. Optimal Control Problem

Optimal control problems present themselves in a variety of domains, such as path planning, to determine the control inputs that minimize a given cost function while fulfilling system dynamics and limitations.

There are various types of optimal control problems, depending on the performance index, the type of time domain (continuous, discrete), the presence of different types of constraints, and what variables are free to be chosen. The formulation of an optimal control problem requires the following:

1. a mathematical model of the system to be controlled,
2. a cost function,
3. a specification of all boundary conditions on states, and constraints to be satisfied by states and controls,
4. a statement of what variables are free.

There are various algorithms for solving optimal control problems, such as Evolutionary Algorithms, Model Predictive Control (MPC), shooting Method approaches, and others.

Searching for some sources in an environment can be approximated as searching for uncertain targets as expressed in [15], which has implemented the shooting approach principles. The shooting approach has been depicted in the optimal search problem, created in the 1940s, offering a fundamental framework for motion planning when looking for uncertain targets. The optimal search problem examines how to maximize the likelihood of detecting a non-evasive target with uncertain properties, given the capabilities of the detection equipment and some restrictions on searcher trajectories.

An example of the shooting method is the Koopman Search Theory algorithm [10], which was first created as part of the Second World War anti-submarine warfare efforts, and has been extensively used in tactical and industrial applications. Further details about the mathematical issue of this model are studied as well as the modern numerical techniques that can now be used to solve it. Then, applying these techniques to a multi-searcher, nonlinear searcher dynamics, and control constraint high dimensional search issue.

3. Methodology

The OCP algorithm considers the dynamics part of the vehicle, so it is discussed here in advance. As a preliminary approach, underwater autonomous

gliders will be approximated to an Unmanned Surface Vehicle (USV), which is extensively studied in the literature. For this research, it has been decided to follow the method studied in [11], which is devoted to Mine Countermeasures (MCM) search missions.

3.1. Modeling

Throughout this thesis, an underwater autonomous glider is approximated to an Unmanned Surface Vehicle (USV), which conducts search missions at a **constant** velocity, and therefore exhibits simple planar motion at the sea surface (pitch, roll, and heave motions are zero) Assuming also that no sway slipping is performed. The equations of motion (EOM) can be easily modeled by kinematics as in equations (1) to (4). Using figure 2, let the state variable pair $(x(t), y(t))$ be the vehicle's position (**vertical** and **horizontal** respectively) in meters north and east from an inertial reference frame, $\psi(t)$ be its heading angle in radians measured clockwise from north, and $r(t)$ is its turn rate in radians per second. If the vehicle travels at constant forward velocity V meters per second, the state–space Equation Of Motion (EOM) for the state vector
$\overrightarrow{x(t)} \equiv [x(t), y(t), \psi(t), r(t)]^T$ and control input u(t) are

$$\dot{x(t)} = V cos\psi(t), \tag{1}$$

$$\dot{y(t)} = V sin\psi(t), \tag{2}$$

$$\dot{\psi(t)} = r(t), \tag{3}$$

$$\dot{r(t)} = \frac{1}{T}(Ku(t) - r(t)). \tag{4}$$

Equation 4 implements a first-order approximation to the well-known Nomoto model for ship-steering equations, after applying the Laplace transform $\mathcal{L}\{f(t)\}$ to the differential equation 4 assuming zero initial conditions, a simple transfer function in **S-domain** between rudder deflection angle $u(s) = \delta_r(s)$ and turn rate r(s) that "is the most popular model used for ship autopilot design due to its simplicity and accuracy" [[11], p. 5]. Note that a transfer function is a relation between output Y(s) and input U(s) in the **S or Frequency** domain as follows: $G(s) = \frac{Y(s)}{U(s)}$. The Laplace transform $\mathcal{L}\{f(t)\}$ is a tool for analyzing Linear Time-Invariant systems (LTI), providing a way to analyze their behavior in the frequency domain.

Design Parameter	Value
Nomoto Gain Constant,K	0.5 1/s
Nomoto Time Constant,T	5.0 s
Velocity,V	2.5 m/s

Table 1: Design Parameters for a USV Model

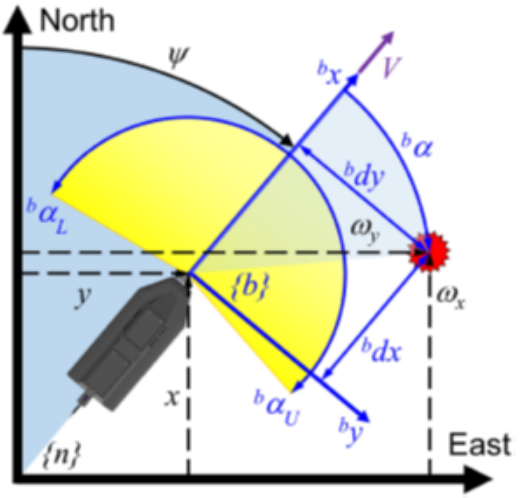


Figure 2: Model: Horizontal Plane Geometry of a USV

3.1.1 Nomoto Steering Model

Bearing in mind that sway velocity is zero, (no sideslip), a Nomoto Steering model is derived as follows:

$$\begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{r}(t) \\ \dot{\psi}(t) \end{bmatrix} + \begin{bmatrix} -N_r & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} N_{\delta_r} \\ 0 \end{bmatrix} \delta_r(t) \tag{5}$$

where z=I_{zz}-$N_{\dot{r}}$
$r(t)$yaw rate, a control variable according to equation 3;
$\psi(t)$ yaw angle;
$\delta_r(t)$ rudder angle;
I_{zz} vehicle's moment of inertia about the z-axis;
Y hydro. coefficients producing sway forces;
N hydro. coefficients producing yaw moments.

In general, control inputs and state variables (and their derivatives) produce nonlinear hydrodynamic forces and moments. It is common practice, however, to approximate these effects by multiplying each contributing variable with a linearized hydrodynamic coefficient. In equation (5), Y and N denote coefficients that produce sway forces and yaw moments, respectively, while subscripts denote their corresponding control input e.g. $\delta_r(t)$ or state variable.

3.1.2 Assumptions

1. Zero stream current
2. Vehicle traveling at a constant speed
3. The Problem is considered to be a simple planar motion at the **sea surface** (i.e. pitch, roll, and heave motions are zero).

4. yaw rate can be directly controlled (Nomoto time constant is considered to be very large)

3.2. Optimal Control Problem (OCP)

Searching for radioactive sources can be approximated as searching for uncertain targets as expressed in [15].

For the sake of constructing a performance criterion for the optimal search problem, the probability of target detection must be modeled. According to the famous Koopman search theory [10], an exponential probability of detection model is derived which has since become ubiquitous in optimal search literature.

The exponential detection model follows from the assumption that the instantaneous rate of detection of a target can be thoroughly modeled.
Given the position of a searcher at $p1(t)$ and a target at $p2(t)$, this instantaneous rate of detection is a function $\eta(p1(t); p2(t); t)$ such that the probability of detection in a quiet small interval $[t, t + \Delta t]$ is independent from previous time intervals and given by the quantity $\eta(p1(t); p2(t); t)\Delta t$. The rate function $\eta(p1(t); p2(t); t)$ is chosen to model the qualities of sensor equipment such as acoustic and sonar sensors, which have fast enough sweep rates to be modeled as continuous processes. An example of a detection rate function using the Poisson Scan Model is shown in figure 3.

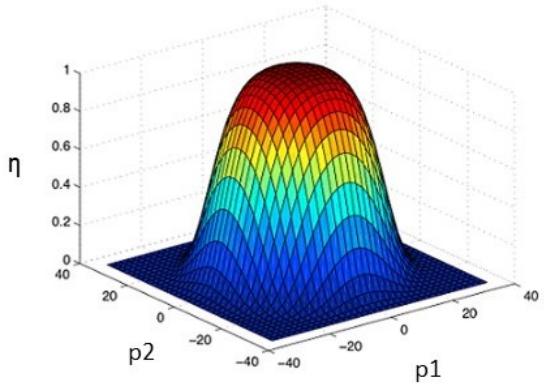


Figure 3: Example detection rate function η: Poisson Scan Model

Taking into account the previous assumptions, an explicit formula for the probability of target detection is derived. If we denote the probability of non-detection with the function P(t), the independence of the time intervals creates the following difference equation:

$$P(t + \Delta t) = P(t)[1 - \eta(p1(t), p2(t), t)\Delta t]$$

where

$\eta(p1(t), p2(t), t)\Delta$t: represents the probability of detection during the time interval Δt. It's the product of the instantaneous detection rate and the duration of the time step.
1η(p1(t),p2(t),t)Δt: the probability of non-detection during the time interval Δt. It complements the probability of detection, indicating the likelihood that the target remains undetected during the time step.
P(t+Δt): the updated probability of non-detection at the next time step t+Δt. It's determined by multiplying the current probability of non-detection P(t) by the probability of non-detection during the time step.

As $\Delta t \to 0$ this has an exponential solution:

$$P(t) = e^{-\int_0^t \eta(p1(\tau),p2(\tau),\tau)d\tau} \tag{6}$$

Using similar techniques, a range of probability for numerous searchers and targets can be obtained. These include the likelihood of finding every target and, in the worst case, the likelihood of finding none of the targets. Generally, this can be expressed as follows:

$$P(t) = G\left(\int_0^t \eta(p1(\tau), p2(\tau), \tau)d\tau\right).$$

Throughout this thesis, the model of conditionally deterministic motion is taken into consideration. That model assumes that the motion of the targets is given entirely by a function of time and a parameter ω. This parameter is an element of a bounded parameter space $\Omega \subset \mathbb{R}^n$ and has a known probability density function $p : \Omega \to \mathbb{R}$.

Considering a conditionally deterministic target motion $y(t|\omega) = h(t|\omega)$ leads to a probability quantity which is itself a random variable, i.e. P(t,ω). A natural performance metric is to minimize the expectation of this random variable over a time interval [0, T]. This will create such a cost function class:

$$J = \int_\Omega G\left(\int_0^T \eta(p1(t), u(t), p2(t, w), w)dt\right)p(w)dw \tag{7}$$

Using the derivations in the reference paper[15], Equation (7) can be rewritten in another way to perform discretization if needed but it is outside the scope of this thesis.

3.3. Minimum Spanning Tree

A Minimum Spanning Tree (MST) [6] is a fundamental concept in graph theory, particularly in the

field of optimization. Given a connected, undirected graph $G = (V, E)$ with a set of vertices V and edges E, an MST is a tree that spans all vertices in V while minimizing the total edge weight.[6]

3.3.1 Problem Formulation

Let $G = (V, E, w)$ be a weighted graph, where $w : E \to \mathbb{R}^+$ assigns a non-negative weight to each edge. The goal is to find an MST T with the minimum total weight:

$$\text{minimize} \quad \sum_{e \in T} w(e)$$

subject to the constraint that T forms a tree that spans all vertices in V.

3.3.2 Prim's Algorithm

One algorithm to find the MST is Prim's algorithm[6], which starts with an arbitrary vertex and greedily grows the tree by adding the smallest-weight edge that connects a vertex in the tree to a vertex outside the tree. The process continues until all vertices are included in the MST.

The algorithm can be described using the following steps:

1. Initialize an empty set T to represent the MST.
2. Choose an arbitrary vertex v_0 and add it to T.
3. While T does not span all vertices:
   (a) Select the smallest-weight edge (u, v) such that $u \in T$ and v is outside T.
   (b) Add vertex v and edge (u, v) to T.

The algorithm maintains a set T that grows into the MST.

3.3.3 Kruskal's Algorithm

Another algorithm for finding the MST is Kruskal's algorithm, which is based on sorting the edges by weight and greedily adding edges to the MST while avoiding cycles.

The algorithm can be summarized as follows:

1. Sort all edges E in non-decreasing order of weight.
2. Initialize an empty set T to represent the MST.
3. Iterate through the sorted edges and add each edge (u, v) to T if adding it does not form a cycle.

The algorithm builds the MST incrementally by adding edges to the growing set T.

3.3.4 Equations and Notations

$$G = (V, E, w) \quad \text{(Weighted graph)} \tag{8}$$
$$w(e) \quad \text{(Weight of edge } e\text{)} \tag{9}$$
$$T \quad \text{(Minimum Spanning Tree)} \tag{10}$$
$$\sum_{e \in T} w(e) \quad \text{(Total weight of edges in } T\text{)} \tag{11}$$

The total weight of edges in the Minimum Spanning Tree (T) is the objective function to be minimized. The solution of an MST is a collection of vertices defined at a distance ***d*** from each other, where these vertices are located outside the polygons. In figure 4, two possible configurations for the path are implemented which are the square and hexagonal configurations where the vertices are chosen to be at a ***d*** distance from each other in the form of a hexagon and the same way for a square configuration.

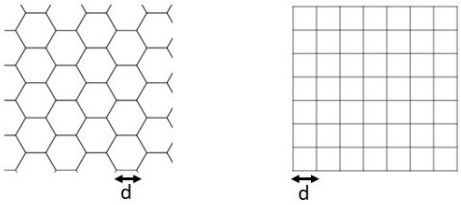


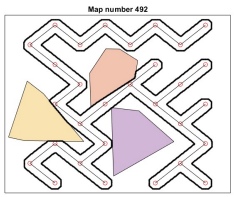
Figure 4: MST Hexagonal grid (left) and square grid(right) configurations

In the hexagonal configuration, the vertices are governed by the following equations:

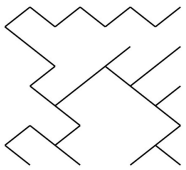
$$v_1 = (i \cdot \frac{3}{2} \cdot d, j \cdot \frac{\sqrt{3}}{2} \cdot d)$$
$$v_2 = (v_1 + (d, 0))$$
$$v_3 = (v_1 + (\frac{d}{2}, \frac{\sqrt{3}}{2} \cdot d))$$
$$v_4 = (v_1 + (-\frac{d}{2}, \frac{\sqrt{3}}{2} \cdot d))$$
$$v_5 = (v_1 + (-d, 0))$$
$$v_6 = (v_1 + (-\frac{d}{2}, -\frac{\sqrt{3}}{2} \cdot d))$$

In these equations, *i* and *j* represent the grid coordinates of the hexagon, and *d* is the spacing between the hexagons. The equations provide the coordinates of the six vertices of the hexagon.

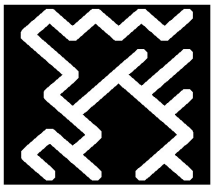
In figure 5, the randomly generated map number 492 is generated with obstacles and paths. For the sake of analysis and debugging, the MST is transformed into this Morphological dilatation in 5c. Then, computing a line that gives contour which gives a path to be traversed as in 5d.



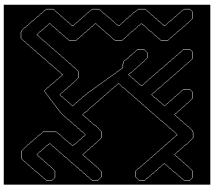
(a) Main figure: Map and Path



(b) Minimal covering Tree



(c) Morphological dilatation



(d) Generated Contour

Figure 5: Map492: MST approach

3.4. Traveling Salesman Problem

The Traveling Salesman Problem (TSP) [1] is a classic optimization problem in the field of combinatorial optimization. It involves finding the shortest possible tour that visits each city exactly once and returns to the original city. TSP can be formulated as an Integer Linear Programming (ILP) problem. In the ILP formulation, binary decision variables are introduced to represent whether an edge (the connection between two cities) is included in the tour or not.

3.4.1 Problem Statement

Consider a set of n cities $C = \{1, 2, \ldots, n\}$, and let d_{ij} represent the distance between cities i and j. The objective of the TSP is to minimize the total tour length L for a given permutation of cities.

3.4.2 Objective Function

The total tour length L can be expressed as the sum of distances traveled between consecutive cities in the tour:

$$L = \sum_{i=1}^{n} \sum_{j \neq i} d_{ij} x_{ij} \tag{12}$$

3.4.3 Constraints

To ensure a valid tour, the following constraints are imposed:

$$\sum_{j \neq i} x_{ij} = 1, \quad \forall i \in C \tag{13}$$

$$\sum_{i \in C} x_{ij} = 1, \quad \forall j \in C \tag{14}$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1$$

for all non-empty proper subsets $S \subseteq \{1, 2, \ldots, n\}$ (15)

Equation (13) ensures that each city is visited exactly once, and Equation (14) ensures that the salesman leaves each city exactly once. Equation (15) ensures Subtours elimination which prevents cycles that do not include all cities.

Similar to MST, vertices are chosen at a specified distance but here it is ***0.5*d*** instead of d. The TSP problem is formulated as follows:

1. Set up the integer linear programming problem to solve the TSP.
2. Defining constraints to ensure that each node is visited exactly once.

The objective function in TSP seeks to minimize the total distance traveled while visiting each node exactly once and returning to the starting node. This can be represented as follows:

$$\text{Minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} \cdot x_{ij} \tag{16}$$

Where:
N represents the number of nodes (locations to visit).
c_{ij} represents the distance or cost between nodes i and j
x_{ij} is a binary decision variable, where $x_{ij} = 1$ if traveling from node i to node j occurs, and 0 otherwise.

4. Results & discussion

Simulations are being shown in this section for each algorithm with 2 different generated maps via the Matlab function $rng()$.

4.1. Traveling Salesman Problem

The solutions and eliminated subtours are visualized on a graph. Additionally, obstacles (polygons) are also plotted on the same graph. To have performance criteria, path computation is performed simply by computing the time it takes to traverse the optimal path, considering a specific speed. Moreover, computing the final positions (first coordinate and second coordinate) and the associated time.

Here is the generated trajectory for the randomly generated maps 493 and 292.

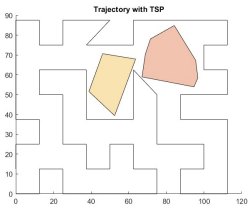


Figure 6: TSP map 493

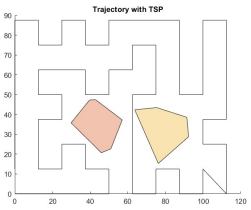


Figure 7: TSP map 292

4.2. Minimum Spanning Tree

Similarly, here is the generated trajectory for the randomly generated map 493 & 292.

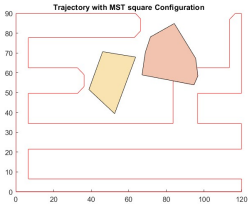


Figure 8: MST with square configuration map 493

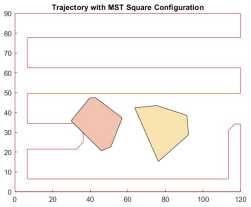


Figure 9: MST with square configuration map 292

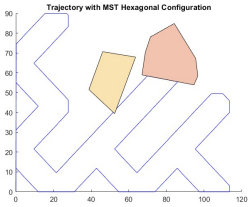


Figure 10: MST with hexagonal configuration map 493

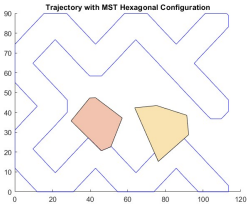
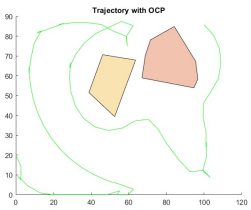


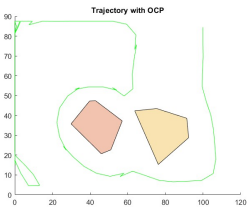
Figure 11: MST with hexagonal configuration map 292

4.3. Optimal Control Problem

From figure 12b, it can be observed that there are a lot of uncovered areas due to a large radius of detection.



(a) OCP map 493



(b) OCP map 292

Figure 12: OCP Trajectories

Note: OCP failed to give a proper solution in some maps e.g. 101 due to a failure of the solver.

4.4. Discussion

In this chapter, the results of map 493 are discussed concerning several performance criteria taken into consideration e.g. processing time of each algorithm, uncovered area, traversed path, and traversed distance. First, let's start with plotting all paths together as shown in figure 13, then traversed distance and time in figure 14.

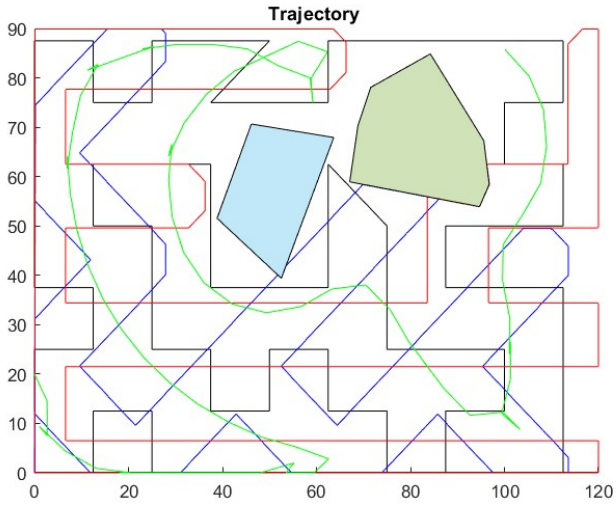


Figure 13: Map493: Paths of all algorithms are plotted

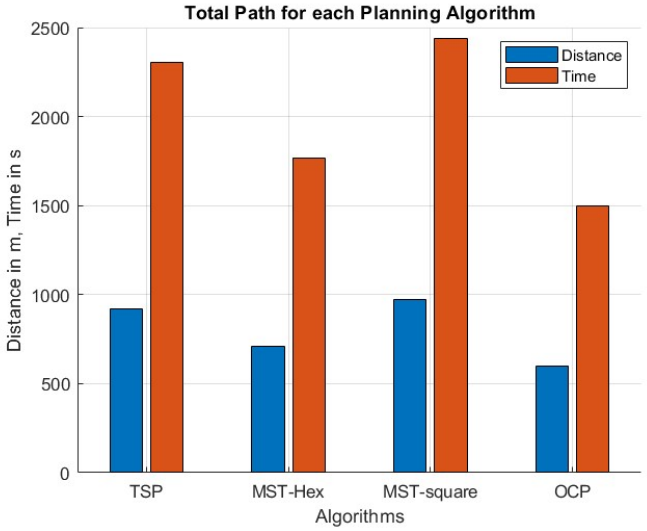


Figure 14: Map 493: Traversed distance, and time taken for each algorithm

The processing time graphs are as follows:

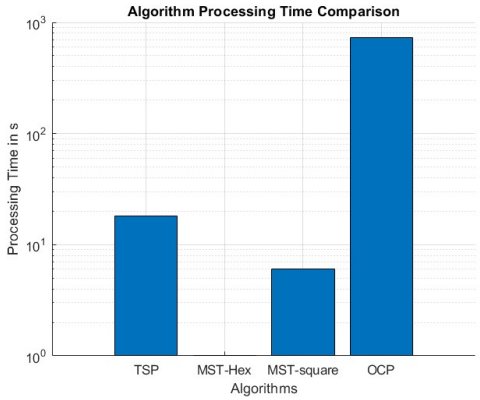


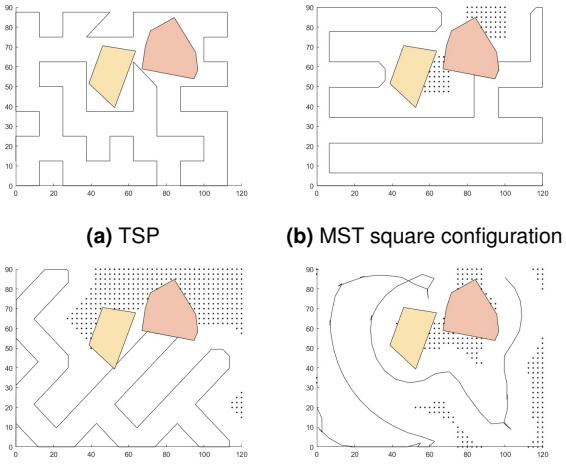
Figure 15: Map 493: Processing time taken for each algorithm

It is observed that the MST square algorithm takes the longest time to traverse the path unlike OCP, which is controlled to be set to 1500s, so one can not compare it with the others. OCP has the least traversed distance, unlike MST square with the largest traversed path.

From figure 15, one can see that there is a prominent drawback of OCP which is the processing time with an average time of about 2550s, which is the maximum processing time compared to the other algorithms while the MST Hexagonal holds the record for the least processing time with about 1s.

4.4.1 Uncovered Areas

The uncovered areas are shown in the environment for each algorithm using map 493. The uncovered areas are represented with dots.



(a) TSP **(b)** MST square configuration

(c) MST Hex configuration **(d)** OCP

Figure 16: Map493: Uncovered Areas

Using the uncovered areas graphs and 17, one can deduce that there is a trade-off between uncovered area and traversed path. The larger the traversed path means less uncovered area. From figure 16, it is observed there are some intersections between the path and some obstacles which means failure to compute a suitable path.

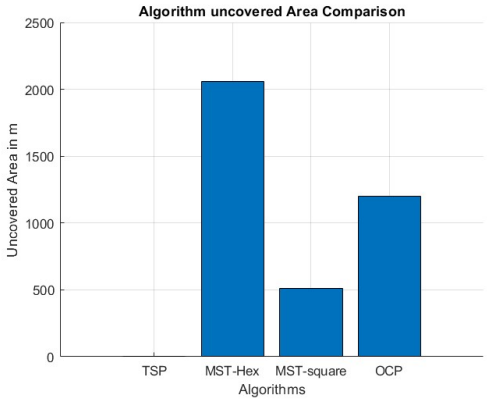


Figure 17: uncovered Areas for each algorithm for map 493

It is clear that the uncovered areas for the OCP are more in map 292 than 493 and this is clear from graph 16d.

In a nutshell, OCP is considered to be more useful if the time to traverse the path is constrained but has a much higher processing time than the other algorithms, and is less suitable when the objective is to cover the area, where a TSP algorithm fares better. Moreover, if the main goal is to obtain a solution fast, MST-based approaches yield smaller processing times.

OCP is more flexible to adapt to other situations where different detections could be done for diversified applications with the help of different sensors being fitted on the vehicle. It has been observed that there are fewer missing spots as the radius of detection is reduced but that comes with a disadvantage with more computational power since it increases the coverage area. OCP has a fixed traversed time with a drawback of more uncovered area.

The OCP approach considers the dynamics of the vehicle other algorithms do not. However, there is a drawback for the OCP which is the missing spots depending also on the radius of detection. As the radius increases, this will lead to more processing time but more covered area to solve the problem of high covered areas in graph 16d.

5. Conclusions

Trajectory generation is an important problem as part of the RAMONES project for which underwater

gliders are used to monitor underwater radioactivity, which requires significant covering and traversing. Coverage algorithms are devised to generate a trajectory that maximizes the probability of detecting a radioactive source and covers the working environment as much as possible in particular through the solution of a generalized optimal control problem (OCP) with an exponential detection model. To have a fair comparison with another algorithm found in the literature review, the Travelling Salesman Problem (TSP) and Minimum Spanning Tree (MST) were implemented to analyze the trajectory generated and compare it with the OCP concerning some performance criteria e.g processing time, uncovered areas, traversed time and distance of the trajectory. OCP is considered to be more useful if the time to traverse the path is constrained but has a much higher processing time than the other algorithms, and is less suitable when the objective is to cover the area, where a TSP algorithm fares better. Moreover, if the main goal is to obtain a solution fast, MST-based approaches yield smaller processing times.

5.1. Future Work

Due to time constraints, an image processing algorithm could not be continued but it is crucial to identify and label obstacles from a Google Maps image.

Applying the Voronoi-based coverage path planning approach can lead to good results. Huang et al. [7] propose a method, for planning the paths of mowers. The approach utilizes an algorithm based on Voronoi diagrams to navigate through irregular environments. By dividing the environment into regions and generating paths that cover each region the robot can effectively cover the area while avoiding obstacles.

References

[1] Travelling salesman problem, Dec. 2022. Page Version ID: 1129661278.

[2] M. Batalin and G. Sukhatme. Coverage, exploration and deployment by a mobile robot and communication network. *Telecommunication Systems*, 26:181–196, 2004.

[3] D. V. Chudnovsky and G. V. Chudnovsky. *Search theory: some recent developments*. Crc Press, 2023.

[4] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics: A survey. *Autonomous robots*, 31:299–316, 2011.

[5] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61:1258–1276, 2013.

[6] R. Graham and P. Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.

[7] K.-C. Huang, F.-L. Lian, C.-T. Chen, C.-H. Wu, and C.-C. Chen. A novel solution with rapid voronoi-based coverage path planning in irregular environment for robotic mowing systems. *International Journal of Intelligent Robotics and Applications*, 5(4):558–575, 2021.

[8] Z. Ismail and M. Hamami. Systematic literature review of swarm robotics strategies applied to target search problem with environment constraints. *Applied Sciences*, 11:2383, 2021.

[9] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30:846–894, 2011.

[10] B. O. Koopman. The Theory of Search. II. Target Detection. *Operations Research*, 4(5):503–531, 1956. Publisher: INFORMS.

[11] S. Kragelund, C. Walton, I. Kaminer, and V. Dobrokhodov. Generalized Optimal Control for Autonomous Mine Countermeasures Missions. *IEEE Journal of Oceanic Engineering*, 46(2):466–496, Apr. 2021. Conference Name: IEEE Journal of Oceanic Engineering.

[12] A. Le, P. Veerajagadheswar, V. Sivanantham, and R. Mohan. Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor. *Sensors*, 18:2585, 2018.

[13] J. Masakuna, S. Utete, and S. Kroon. A coordinated search strategy for solitary robots. 2019.

[14] G. Strimel and M. Veloso. Coverage planning with finite resources. 2014.

[15] C. L. Walton, Q. Gong, I. Kaminer, and J. O. Royset. Optimal Motion Planning for Searching for Uncertain Targets. *IFAC Proceedings Volumes*, 47(3):8977–8982, Jan. 2014.