Coverage Path Planning for a Moving Vehicle

by

Barry J. Gilhuly

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A simple coverage plan called a Conformal Lawn Mower plan is demonstrated. This plan enables a UAV to fully cover the route ahead of a moving ground vehicle. The plan requires only limited knowledge of the ground vehicle's future path. For a class of curvature-constrained ground vehicle paths, the proposed plan requires a UAV velocity that is no more than twice the velocity required to cover the optimal plan. Necessary and sufficient UAV velocities, relative to the ground vehicle velocity, required to successfully cover any path in the curvature restricted set are established. In simulation, the proposed plan is validated, showing that the required velocity to provide coverage is strongly related to the curvature of the ground vehicle's path. The results also illustrate the relationship between mapping requirements and the relative velocities of the UAV and ground vehicle. Next, I investigate the challenges involved in providing timely mapping information to a moving ground vehicle where the path of that vehicle is not known in advance. I establish necessary and sufficient UAV velocities, relative to the ground vehicle velocity, required to successfully cover any path the ground vehicle may follow. Finally, I consider a reduced problem for sensor coverage ahead of a moving ground vehicle. Given the ground vehicle route, the UAV planner calculates the regions that must be covered and the time by which each must be covered. The UAV planning problem takes the form of an Orienteering Problem with Time Windows (OPTW). The problem is cast the problem as a Mixed Integer Linear Program (MILP) to find a UAV path that maximizes the area covered within the time constraints dictated by the moving ground vehicle. To improve scalability of the proposed solution, I prove that the optimization can be partitioned into a set of smaller problems, each of which may be solved independently without loss of overall solution optimality. This divide and conquer strategy allows faster solution times, and also provides higher-quality solutions when given a fixed time budget for solving the MILP. We also demonstrate a method of limited loss partitioning, which can perform a trade-off between improved solution time and a bounded objective loss.

Acknowledgements

I would like to thank my supervisor, Stephen L. Smith, for all of his help and support.

I would also like to thank my fellow graduate students for being there when I needed them.

Dedication

To Mircah, Love Stoney

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cooperative applications that combine a ground vehicle and an unmanned aerial vehicle (UAV) are an extremely active area of research [75], with many open questions to be resolved. The capabilities of ground and aerial vehicles are quite different, and collaborative applications take advantage of the strengths of both.

Collaboration can lead to better path planning for the ground vehicle. UAVs are highly mobile and can readily explore large areas of terrain and negotiate around or over most obstacles. They are frequently used in applications such as search and rescue, where their greater mobility and unobstructed viewpoint help to cover terrain and locate targets quickly [33]. UAVs are also more likely to have a clear view of GPS satellites [13] for navigation purposes. Ground vehicles, on the other hand, are confined to two dimensions and must contend with every tree, hole, and creek they encounter. They can be further slowed by the type of terrain they are crossing – mud and sand are much harder to traverse than pavement. A UAV, with its greater speed, mobility and ability to fly above the area [10], can stay ahead of the ground vehicle and provide real-time imagery for route planning and mapping. With real-time information from the UAV, the ground vehicle can better adapt its path as the situation demands.

In an urban environment, a larger heterogeneous team of autonomous vehicles (fixed wing aircraft, a blimp, and ground vehicles) is used to provide an operator with situational awareness over an wide area [13]. The height and maneuverability of the aerial vehicles allows them to see over obstacles, have unobstructed access to GPS and provide localization and mapping information to the ground vehicles. This approach has been extended to surveillance and tracking [77] where, using a probabilistic algorithm, the team plans observation paths that attempt to take into account occlusions in the environment and

keep the target in view at all times.

The UAV's mobility advantages also apply to mapping and exploration applications. Taking advantage of its ability to fly high overhead, the UAV is able to generate a point cloud representation of the environment around the ground vehicle [23] using the data it collects. The point cloud is subsequently used to generate feasible routes through the local terrain for the companion ground vehicle. Even though UAVs excel at travelling quickly over terrain [47, 61], the information they collect generates a relatively low resolution picture of features in the environment. Yet, despite the relatively low resolution and accuracy, the ground vehicle can process this raw information to identify candidate areas for inspection [29]. Such cooperative actions reduce the overall distance the ground vehicle must travel by eliminating areas of the environment unlikely to be useful.

Collaborative operations can significantly increase the flight range of the UAV. One of the biggest challenges for UAVs is their limited flight duration, with many having flight times of thirty minutes or less. As a result, they require frequent stops for refuelling. Further, the duration of flight can be affected by the weather conditions (shortened due to extreme heat or cold), the requested flight speed, and by the complexity of the route flown – a turn manoeuvre is more energy intensive than straight flight [25]. Ground vehicles operate for much longer on a single load of fuel, have more carrying capacity, and can even function as a mobile fuel depot [50], significantly extending the range of the UAV. The same approach applies in agricultural applications such as [68], where the superior carrying capacity of the ground vehicle can maximize the range of the UAV.

Collaborative applications improve inspection studies and make planning actions easier. UAVs are commonly used to provide a ground vehicle with an additional perspective of the environment. While navigating a minefield [12], a ground vehicle can launch a UAV when more visual information is required, and the UAV's sensors will feed data to the ground vehicle for processing and inspection. Similarly, a UAV and ground vehicle team can work together to explore and map in the aftermath of a disaster [4]. Finally, in agriculture [42], the imagery from the UAV identifies potential weed locations over the entire field, and the ground vehicle travels only to those locations to provide detailed inspection and identification.

In this thesis, I consider the problem of providing current terrain information to a ground vehicle moving through an uncertain environment, perhaps in the aftermath of a natural disaster. In an environment where road or safety concerns are in flux, it is important for the driver of the vehicle to know the current conditions of the terrain on the route ahead. Maps produced even the day before may be rendered useless if a storm has washed out a key bridge or blown trees onto the road, a forest fire burn line has shifted in

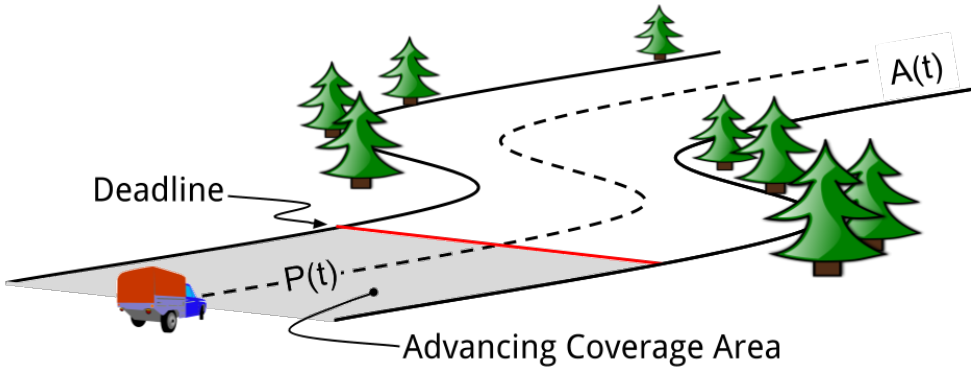


Figure 1.1: An illustration of the increasing coverage area ahead of the ground vehicle.

the night, or a building has collapsed after an earthquake.

In such situations, it is critical that the driver has access to the most current information. A UAV, working with the ground vehicle, can fly a coverage plan some distance ahead, transmitting the real-time state of the terrain along the ground vehicle's planned route. This coverage data from the UAV may be viewed as a live feed, or possibly input into another system for 3D reconstruction to aid in vehicle route planning. Since the ground vehicle is continuously moving forward, the coverage area is also continually updating, requiring the UAV to maintain a leading edge position just ahead of the coverage area (Figure 1.1). This leading edge is referred to as the deadline – a line perpendicular to the ground vehicle's path at the coverage distance. The coverage plan that the UAV follows must ensure that any terrain behind the deadline has already been covered by the UAV. This results in an observed corridor that straddles the ground vehicle's route, with two variables that determine the area to be mapped: the width of the coverage corridor, and the distance ahead along the route. The corridor through the environment is centered on the ground vehicle's route as illustrated in Figure 1.1.

I explore the coverage planning problem for a moving ground vehicle in three scenarios: a UAV covering a known ground vehicle path, a UAV required to anticipate and cover the ground vehicle's path, and a UAV selectively covering only targeted areas of the ground vehicle's path.

The known path method, discussed in Chapter 3, allows the UAV access to a limited window of the ground vehicle's upcoming path. Based on that knowledge, the UAV must build and execute an appropriate coverage plan. Due to the short-term nature of the path information, the UAV must continuously update its mapping plan as the ground vehicle advances and supplies updated directions. For this purpose, we adapt the lawn mower

plan to take advantage of the limited information, and then characterize the efficiency of our approach by establishing lower and upper bounds on the UAV's velocity.

The predictive problem, explored in Chapter 4, considers the case where, for security or other reasons, the planned route is unavailable to the UAV. As a result, the UAV must use alternative data to predict the ground vehicle's future path. Initial predictions are made based only the ground vehicle's current state (position and velocity) as well as the previous path.

I conclude my research with an implementation of a selective approach in Chapter 5. Using the idea that covering the complete ground vehicle corridor is resource intensive, I instead investigate a method of limiting the area that UAV must cover to only the terrain which is expected to be obstructed from the ground vehicle's view. The information about the ground vehicle's intended path, when combined with general knowledge of the terrain, allows for the prioritization of areas for mapping. A coverage plan is then found by casting a Mixed Integer Linear Program (MILP), developing in the process, a method of reducing the problem size to achieve faster solution times. Even with the reduction in coverage area, finding an optimal coverage plan is still difficult. I investigate using a divide-and-conquer method to reduce the problem size by making exact optimality preserving cuts in the graph and solving the sub-problems. Finally, I develop a limited loss version of the exact cuts, allowing broader application.

1.1 Literature Review

My research touches on two primary applications in the literature: coverage path planning and collaborative mapping.

1.1.1 Coverage Path Planning

Coverage Path Planning (CPP) is the process of constructing a route through an environment that allows an autonomous robot to collect sensor data from all areas of that environment. The robot is assumed to have some sort of sensor footprint that informs the spacing of the path, with a planned route minimizing unnecessary sensor overlap (or restricting it to within a certain range) – surface reconstruction applications may require a high percentage overlap in order to ensure there are enough detectable features shared between images to reconstruct depth [62]. Once the size of the sensor footprint is determined,

a path is constructed that minimizes some aspect of coverage. Typically applications minimize one or more of time, distance or overall energy consumption.

Decomposition The style of coverage planning used in this paper is based on exact cellular decomposition – a process of dividing the environment into many small cells, and then planning a path that visits all of the cells, thereby completely covering the area as required. Early implementations of this method used trapezoidal decomposition [15] to first divide the area into convex areas before planning coverage for each. This strategy, however, could result in sub-optimal coverage paths, due to unnecessary divisions and increased transition travel between the subdivided areas.

The term boustrophedon decomposition is introduced in [14], and describes a method of covering an area with a back and forth, lawn mower style pattern. The method minimizes overlap and transition movements between cells. Non-convex areas may be covered completely if the orientation of the lines defining the robot path can be drawn continuously without creating an orphan section or extending outside of the area. The problem of finding a shortest length coverage plan is subsequently shown in [3] to have NP-hard complexity.

Minimizing Turns Turns are the most expensive component of flight, as the UAV must apply additional force to halt travel in one direction and start in another. The author of [37] argues that, by minimizing the number of turns in a coverage path, a time optimal solution can be generated. A method of decomposing concave polygons and finding the optimal coverage sweep pattern is presented; however, the algorithm is described as exponential in nature and therefore impractical for complex environments.

Building on [37], the authors of [48] prove that turns are more costly in terms of energy requirements, time, and distance travelled, and that the way to minimize coverage costs is to minimize the number of turns in the coverage pattern. In [25], the cost due to turns is quantified in increased energy cost. Further improvements are made in [69] to find an optimal coverage path that takes into account the approach and exit paths the UAV must take when leaving and returning to the initial launch location.

Optimizing by Sweep Line Complex concave regions may be decomposed into smaller convex or concave areas that are optimally coverable a single sweep direction [7]. Coverage planning is then carried out using a Generalized Large Neighbourhood Search (GLNS) on the individual sweep lines to find the minimized path. In Chapter 5, this method forms the basis of placing the individual sweep lines for complete coverage.

In [41], an optimal plan is found using a sweep method to cover multiple disjoint areas, minimizing the overall coverage path. However, the areas of interest in this case are limited to rectangular shapes. The authors of [72] present an alternative method for planning coverage when the areas to be covered are disjoint. Instead of first determining the sweep direction by finding the minimal altitude of each area, the authors instead first calculate a coverage order by solving the Travelling Salesperson Problem (TSP) that minimizes the distance required to visit the center of each area. Once the global order has been established, the optimal coverage path is found by minimizing the cost of the coverage path, assuming the UAV starts at the previous area and ends at the next.

1.1.2 Applications of Coverage Path Planning

Surveying and Inspection Surveying and Inspection tasks are some of the most straightforward applications of CPP – a path must be planned that allows the robot to visit every part of the area of interest. In agriculture, for example, a farmer may use a UAV to inspect crops for insect damage, or plan the route for the combine to harvest [43]. Inspection applications involve recording the outside of buildings, high and inaccessible towers (for example, radio antennae) and bridges [60, 35, 51].

Surveillance In [74], an application of multiple UAV surveillance planning presents a time-based coverage problem in a fixed frame of reference. The algorithm plans for visits to disjoint areas using a fleet of UAVs. Each area is decomposed into individual cells, and a plan for visiting each is created. There is no notion of using sweep lines or otherwise minimizing the energy consumption required to cover each area. As it is a surveillance problem, visits to areas of the map must be scheduled such that information on a particular cell never becomes stale.

Persistent Monitoring Persistent monitoring is the application of CPP to surveillance problems, allowing robot coverage of locations whose contents may change over time. Depending on the formulation, different areas of interest in the region may have an associated timer or staleness factor that requires the agent (the UAV) to revisit. The simplest example may be that of a robot vacuum cleaner that has to maintain a floor to a certain level of dust. As dust is expected to build up over time, the vacuum must plan a path that visits every location before the threshold where the dust becomes obvious [2, 64].

The Art Gallery Problem The Art Gallery Problem and its closely related variant, the Watchman's Route Problem, both involve actively monitoring an area with a number of guards (sensors). In the Art Gallery problem, the guards are stationary, and the planning problem requires locating them such that all locations of the facility are in view at all times. The Watchman's Route problem allows the guards to move, and the challenge becomes one of finding an optimal route such that all parts of the area are in view of at least one of the guards from some location along the path [49]. Further variants add scheduling, meaning locations must be visited within a certain window of time (e.g., every location must have been seen within the last hour)[70].

1.2 Collaborative Mapping

In the mapping domain, several studies investigate using a collaborative team of UAVs and ground vehicles to explore an unknown terrain [44, 12, 36]. However these studies generally employ a UAV in a stationary eye in the sky position above the ground vehicle [36], creating a high vantage point, but providing only a limited view of the area that lies ahead. Others have investigated using the UAV's faster velocity and easier navigation to map a region quickly, allowing a ground vehicle to plan a safe route through difficult terrain while visiting locations of interest [46, 12]; in these studies though, the region to be mapped is fixed, and not restricted by the motion or capabilities of the ground vehicle. In another example, machine learning techniques [16] are used to plan the ground vehicle's route based on information collected by the UAV. Still other mapping studies use UAVs to explore points of interest while the ground vehicle acts simply as a mobile supply depot, providing support and resources to keep the UAV flying [59, 50].

In [44] a pair of UAVs are used to provide a stereo image of the path ahead of the ground vehicle. Using the information the UAVs provide, the ground vehicle is able to detect upcoming obstacles. In this case, the UAVs are limited to the immediate proximity of the ground vehicle, using it as a reference point to maintain position and establish the visual stereo base line for object recognition. A forward-facing camera on a UAV can also be used to provide a ground vehicle with a better view of the surrounding terrain [36]. The UAV maintains a position directly above the ground vehicle.

As an example of cooperative exploration with a unmanned ground vehicle (UGV) and a team of UAVs [31], a UGV dynamically modifies the potential field representing the environment based on obstacle location information from a team of UAVs flying directly overhead. Without the input of the UAVs, the UGV can become trapped in a local minimum. In a somewhat more extreme example, a UAV may even serve as the ground

vehicle's sensors [57], allowing the ground vehicle to avoid obstacles in the terrain that it is unable to sense.

1.3 Contributions

In Chapter 3 I introduce the problem of providing continuous coverage of the path ahead of a moving ground vehicle. Next, I present a plan capable of providing coverage with only limited knowledge of the ground vehicle's path. I further establish upper and lower bounds on the length of this plan and, based on that distance, estimate the required UAV velocities. Then, I prove that when the curvature of the ground vehicle path is limited, the proposed plan provides a UAV path that requires no more than twice the velocity of that required to cover the optimal path.

In Chapter 4 I investigate a worst case scenario, where the future path of the ground vehicle is completely unknown to the UAV planner. I establish the necessary UAV speed for complete terrain coverage, and illustrate one possible implementation that demonstrates a sufficient speed. I show that both the necessary and sufficient speeds increase dramatically when little or no information on the ground vehicle's future path is available.

In Chapter 5 I present the problem of providing UAV coverage path planning to a moving ground vehicle and solve it as an Orienteering Problem with Time Windows. A Mixed Integer Linear Program is formulated to maximize the coverage area while minimizing UAV path length. Next, I prove this coverage problem, under certain conditions, can be exactly partitioned without loss of optimality, resulting in shorter and more predictable solution times. Finally, I present a method of limited loss partitioning with a bounded loss in solution quality.

Chapter 2

Background

2.1 Path Planning

In the broad space of path planning, the goal is to find an optimal path through the environment to some destination, possibly collecting some reward along the way. The measure of optimality depends on the problem being considered, but may include path length, resources required, reward collected, etc.

2.1.1 Goal Oriented Path Planning

In graph based, goal oriented path planning, the task is focused on moving the agent from a starting position to a final position with as direct a path as possible while avoiding collisions in the environment. These algorithms can be grouped into two broad categories: deterministic methods and stochastic methods. In both cases, the first step is to build a graph that represents the planning space. Deterministic methods, such as a Visibility Graph [38], use a heuristic to place and connect vertices. Stochastic methods, such as Probablistic Road Maps (PRM) [26], use randomly sampled locations in the environment to build a graph. PRMs build a graph of the entire environment by first randomly sampling many locations as vertices, then adding a connecting edge between any two locations in direct line of sight of each other.

Once the graph has been constructed, it may be searched by one of many possible algorithms to find the shortest path through the environment. Two commonly used algorithms are Dijkstra's algorithm [20] which finds the shortest path to all vertices in a graph from one source, and A* [32], which finds the shortest path between two vertices.

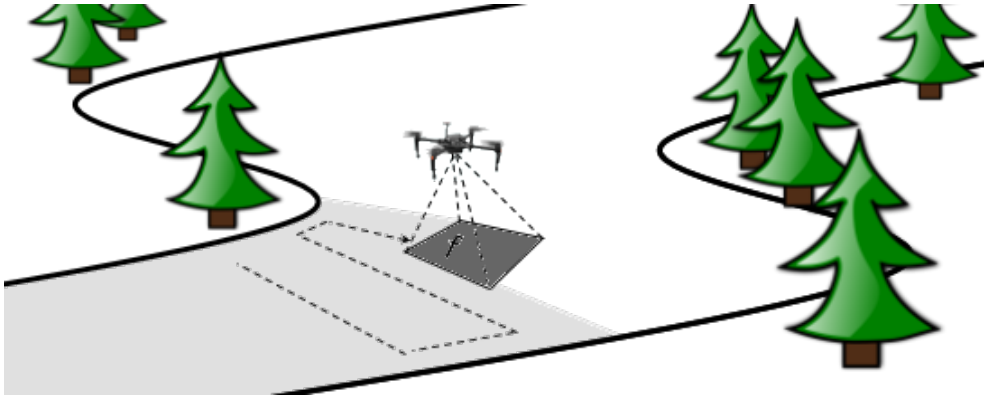


Figure 2.1: The optical footprint of the UAV, projected on the ground

2.1.2 Coverage Path Planning

In coverage path planning, the objective is to find a path that allows the agent to capture information from the entire environment. The agent may carry a sensor package that enables it to capture a footprint in the environment (see Figure 2.1). Coverage path planning is process of guiding the agent through the environment such that all points in the environment fall within the agent's footprint at some time. Consider an agent moving through an environment E in $\mathbb{R}^2$. If the area of the agent's sensing footprint can be expressed as $A(t)$ for some time t in $[0, T]$, then the environment can be considered completely covered if

$$\bigcup_{t \in [0,T]} A(t) = E.$$

2.1.3 Path Planning on a Graph

Coverage Path Planning (CPP) can be expressed as a graph, with the vertices taking on the coordinates in $\mathbb{R}^2$ of each of the locations the robot must visit, and the edges representing the cost for the robot to travel between the individual vertices. Additional costs and constraints may be added for the time required to visit a particular vertex, how much an agent can carry, resource limits, and so on. Once put into graph form, the planning problem can usually be solved using one of the following techniques.

Travelling Salesperson Problems

The Travelling Salesperson Problem (TSP) [24] describes the plight of the salesperson looking for the shortest route to visit all of her customers before returning home. Represented as a graph, the customers and the salesperson's home are the vertices and the edges represent the connections between vertices, typically with an associated travel cost. The goal is to find the shortest simple path that visits all vertices and returns to the start without repetition.

Vehicle Routing Problems

The Vehicle Routing Problem (VRP) is a generalization of the TSP [11], where the agent now has a carrying capacity. First described in [18] as the Truck Dispatching Problem, each agent (or truck) has a fixed maximum capacity. The task is to find the shortest set of routes which allows the agent to visit each of the customers without repetition, only returning to the depot when necessary. If the agent has a sufficient capacity to visit all of the customers in one trip, then the problem is equivalent to the TSP. Like the TSP, there are many variants: multiple agents, dynamic vehicle routing problems (DVRP) with new customers that appear over time and must be serviced, and with time windows (DVRTW), to name a few. The vehicle routing problem with time windows enforces an interval at which the agent is allowed to visit the customer. If the agent arrives before the interval starts, the agent is forced to wait.

Orienteering Problems

The Orienteering Problem is another variation of the TSP where the agent has limited resources and is only able to visit a subset of the vertices in the graph. The planner must decide the optimal vertices to visit, and the visiting order, to solve the problem. These problems are generally viewed as being NP-hard [71].

2.2 Dubins Vehicle Model

The Dubins Vehicle Model [21] was developed to describe the possible forward motion dynamics of a vehicle using a bicycle model: a front wheel (or pair of wheels) that provides steering, and a following rear wheel(s) on a axle rigidly attached to the front wheel. The Dubins model describes the forward motion of a typical car.

The Dubins model characterizes the shortest path between two poses as a series of three possible motions: a turn at the minimum turning radius, a straight section or possibly an opposite turn, followed by another minimum radius turn. Using this description, there are six possible motion patterns: Right-Straight-Right, Right-Straight-Left, Right-Left-Right, Left-Straight-Right, Left-Straight-Left, and Left-Right-Left.

The Dubins model has also been used to represent the flight characteristics of fixed wing aircraft [53], appropriate since fixed-wing aircraft typically have a fixed minimum radius which is related to the aircraft's forward velocity – assuming a constant velocity, then the curves can be calculated. In Chapter 5, the Dubins model is also used to estimate an appropriate cost for turns, accounting for the increased time, energy and distance involved as the UAV slows and accelerates to execute each manoeuvre.

2.3 Computational Geometry

The research in Chapter 5 uses aspects of computational geometry to solve the reduced coverage problem.

2.3.1 Convex Hulls

Convex hulls are used in this study to determine the necessary coverage area, given a collection of discrete cells, each cell representing a small area of the environment. Representing each cell as point in $\mathbb{R}^2$, the enclosing convex hull is the polygon formed if a string is wound tightly around the set, ensuring no concavities in the boundary.

2.3.2 Minimum Altitude

The minimum altitude of a polygon is the smallest distance between a pair of parallel lines that contact a polygon on two sides. It is referred to as the rotating calipers method (imagine the polygon is turned within the jaws of a pair of calipers and the minimum space between the jaws measured) and is introduced in [63] and [58].

2.3.3 Visibility Polygon

A visibility polygon represents the line of sight viewable space from a single position in an environment. The polygon is constructed by sweeping a ray from the view point in a circle

and connecting where the ray intersects with obstacles.

2.4 Linear Programming

Linear programming is a method [6] of solving optimization problems expressed with a goal function and constraints of the general form:

$$\begin{aligned} \textbf{minimize:} \quad & c^T x \\ \textbf{subject to:} \quad & Ax \leq b \\ & x_i \geq 0 \end{aligned}$$

The goal function and the constraints must all be linear. If a solution to a Linear Programming problem exists, that solution is exact and can be calculated in polynomial time.

2.4.1 Mixed Integer Linear Programming

The MILP is an extension of Linear Programming; that is, it is a method of minimizing (or maximizing) a linear function subject to a set of constraints. However, in MILP problems, some subset of the constraints is limited to integer values. MILP problems are known to be NP-hard [54] and, as a result, solutions are difficult to find, generally requiring exponentially increasing time and resources as the size of the problem grows. MILP and TSP solutions were found using the Gurobi solver [30].

2.5 Simulation Libraries

All simulations are implemented in C++ using the following libraries and tools: OMPL [66], View polygons [45], Boost [8], Minimum polygon heights [58], Concave polygons [1], Quick-Hull [52], Polygon decomposition [56], and Dubins curves [73].

Chapter 3

Covering a Known Path

3.1 Introduction

In this chapter, I consider a path planning problem where a moving ground vehicle has provided a UAV with access to a limited window of the ground vehicle's upcoming path. The UAV must build and execute an appropriate coverage plan for the ground vehicle's travel corridor based on the provided information. Due to the short-term nature of the path information, the UAV must continuously update its mapping plan as the ground vehicle advances and supplies updated directions. I adapt the common lawn mower plan to take advantage of the limited information, and then characterize the efficiency of my approach by establishing lower and upper bounds on the UAV's velocity.

3.2 Problem Statement

Consider a ground vehicle moving through an environment in $\mathbb{R}^2$ following a smooth path $P(t)$ at a constant velocity v_{gv} for time $t \in [0, t_{\max}]$.

A distance d_{map} ahead of the ground vehicle defines the length of the coverage area. The width of the coverage area w is specified by the operator as a path parameter. As the ground vehicle moves along the path, this $d_{\text{map}} \times w$ coverage area moves ahead of it, generating a mapping demand for the UAV. I generally expect the total length of the path to be much greater than w.

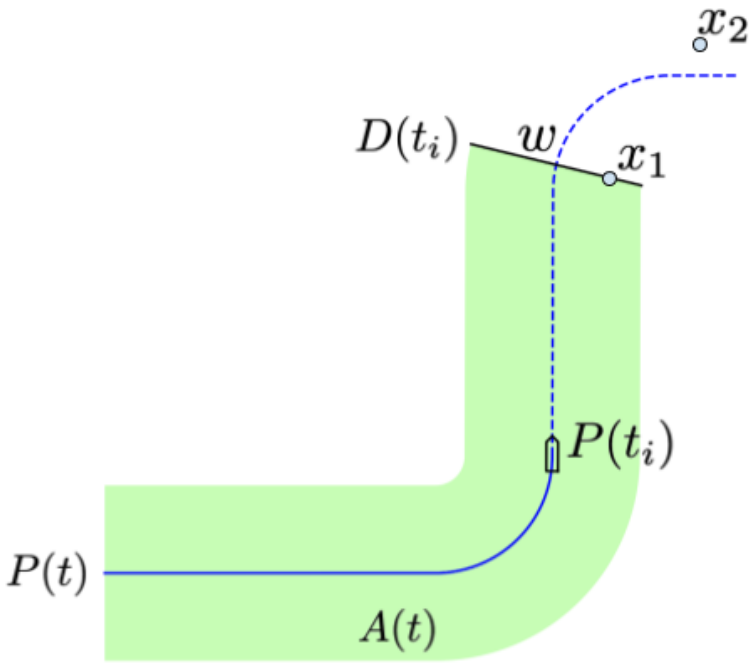


Figure 3.1: Ground Vehicle Path and the Coverage Corridor.

The leading edge of this coverage area is called the deadline. If I define t_{map} as the time required for the ground vehicle to traverse the map distance, $\frac{d_{\text{map}}}{v_{\text{gv}}}$, then given a unit normal $\overrightarrow{n}$ to $\dot{P}(t + t_{\text{map}})$, define the deadline of the coverage area, $D(t)$, as

$$D(t) = \left\{ x \in \mathbb{R}^2 | x = P(t + t_{\text{map}}) + \alpha \vec{n},\ \alpha \in \left[-\frac{w}{2}, \frac{w}{2} \right] \right\}.$$

All ground vehicle paths, $P(t)$, are members of the set

$$\mathbb{P} = \left[P(t) \middle| c(t) \leq \frac{2}{w}, t \in [0, t_{\text{max}}] \right],$$

where $c(t)$ is the curvature of $P(t)$ at time t. This curvature constraint ensures that, as the vehicle progresses along the path P(t), the endpoints of the deadline D(t) always make non-negative progress along the boundary of the coverage area.

From this the coverage area $A(t)$ can be more formally defined to be the union of points found by sweeping $D(t_{\text{dl}})$ along $P(t_{\text{dl}})$ for $t_{\text{dl}} \in [0, t + t_{\text{map}}]$, expressed as

$$A(t) = \cup_{t_{\text{dl}}=0}^{t+t_{\text{map}}} D(t_{\text{dl}}).$$

A UAV is deployed to provide mapping imagery, using a monocular vision system to capture terrain data. Similar to [76], I model the UAV motion using single integrator dynamics and focus on the high level planning problem. The UAV's camera has a fixed-size square optical footprint with sides of length f, where $f < w$. If $f \geq w$, then the solution is to simply fly the UAV along the ground vehicle path at the same velocity, $v_{\text{uav}} = v_{\text{gv}}$. The total area of the map covered by the UAV over the interval $[0, t]$ is denoted $M(t) \subset \mathbb{R}^2$.

The UAV is unable to create an optimal mapping plan, as it only has a limited window of the ground vehicle's upcoming path.

Figure 3.1 shows an example path with the ground vehicle located at $P(t_i)$. The coverage area starts at $P(0)$, is centered on $P(t)$, and continues to a point d_{map} units ahead of the ground vehicle at $P(t + t_{\text{map}})$. The environment is assumed to be free of obstacles that affect the UAV. There may be obstacles that limit the possible trajectories of the ground vehicle; however, I assume that the UAV does not have access to this information.

For all points in $x \in A(t)$, the expiry time $t_{\text{exp}}(x)$ is defined as the time at which x intersects with the deadline $D(t')$ for the first time $t' \leq t$. If a point is not in $M(t)$ before expiring, then it is considered a coverage failure. The expiry of a point x is

$$t_{\text{exp}}(x) = \arg\inf_t \left\{ x \in A(t) \right\}.$$

For example, in Figure 3.1, the point x_1, seen on the line $D(t_1)$ has just expired. The point x_2 is still outside the coverage area.

At time $t = 0$, I assume the UAV is positioned at the beginning of its first pass on one side of the path, ready to start mapping. The deadline is located at $P(0)$, with the ground vehicle not yet on the path. After a delay of $\Delta t = \frac{f}{v_{\text{gv}}}$, enough time for the UAV to map the first pass of the path, the ground vehicle and the deadline begin to move forward.

Given this background information, the problem may be formally stated.

Problem 3.2.1 (Complete Coverage)**.** Consider a ground vehicle traveling through an environment following a path, $P(t) \in \mathbb{P}$, creating a coverage demand of $A(t)$. A UAV travels ahead of the ground vehicle producing a coverage area of $M(t)$. Assume the UAV has knowledge of an upcoming window of the ground vehicle's path, $P(\bar{t}), \bar{t} \in [t, t + \Delta t]$. Determine a plan for the UAV that guarantees

$$A(t) \subseteq M(t_{\max}), \forall t \in [0, t_{\max}]. \tag{3.1}$$

I seek to characterize this plan's efficiency as follows.

Problem 3.2.2 (Proof Of Efficiency)**.** Given the plan determined by (3.1), what is the efficiency relative to the optimal coverage plan for the same path, $P(t)$?

3.3 The Conformal Lawn Mower Path

It is well established that a simple, non-overlapping lawn mower path is an optimal method for covering a rectangular area [14]. I propose that for the ground vehicle path, $P(t)$, I can define a *Conformal lawn mower* path such that the lines defining the back and forth motion of a regular lawn mower may no longer be parallel. Instead, the angle between any two adjacent lines is allowed to range from parallel up to a maximum value defined by the curvature of the path and the UAV's optical footprint. Refer to Figure 3.2 where the UAV coverage plan (red dashed line) is overlaid on the ground vehicle path.

Definition 3.3.1 (Traversal)**.** A *Traversal* is a line segment of length w normal to $\dot{P}$. To guarantee complete coverage, the distance between any two traversal lines has an upper bound of f.

Definition 3.3.2 (Transit)**.** A *Transit* is defined as the section of the UAV's coverage plan that connects the ends of two adjacent traversals. Transits are assumed to follow the profile of the path edge (i.e., an arc when the path is curved).

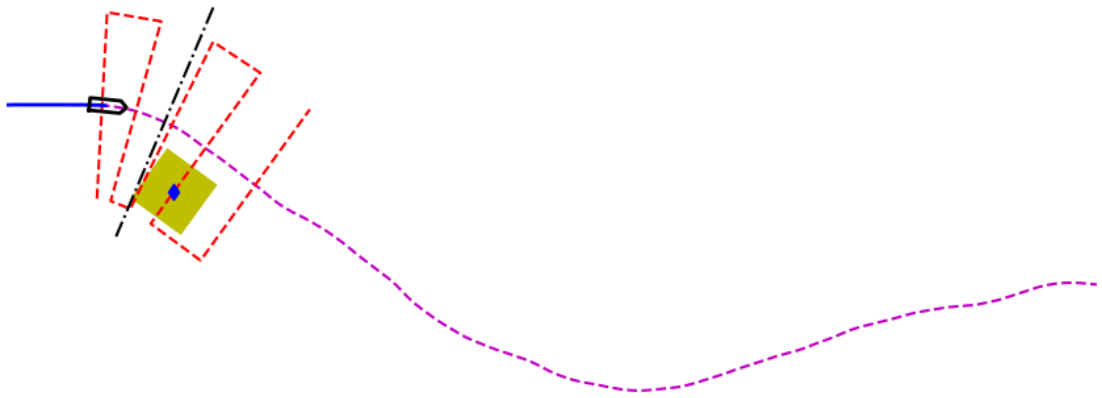


Figure 3.2: Conformal lawn mower plan with a limited window.

Definition 3.3.3 (Period)**.** A *Period* for a Conformal lawn mower plan is a grouping of the movements required to cover a section of the path and return to the same position, but shifted forward along the path. A period consists of the following movements: traversal, transit, traversal, transit.

The Conformal plan is a sequence of alternating traversals and transits that allow the UAV to completely map $A(t)$. The procedure for constructing a Conformal lawn mower path is shown in Algorithm 1.

Algorithm 1 Conformal Lawn Mower Plan

1. Add an initial traversal at $P(0)$ to the plan.
2. Find the first point on the path $P(t)$ such that a traversal centred at $P(t)$ has an endpoint at distance f from the corresponding endpoint on the same side of the previous traversal.
3. Add a transit to this traversal at $P(t)$ and the traversal to the plan, where successive transits alternate sides.
4. If the ground vehicle has stopped, add a final transit and traversal to the plan and exit.
5. Otherwise, when there is new path information, repeat from step 2.

The UAV uses the provided path information to map the initially known $A(t)$ following

the Conformal plan. As the ground vehicle moves forward and additional path information comes available, the UAV plan is extended, allowing the UAV to map the new territory.

Theorem 3.3.1 (Complete Coverage)**.** The Conformal lawn mower plan in Algorithm 1 provides complete coverage of path $P(t)$.

Proof. (By Construction.) From Algorithm 1, the ground vehicle path is sampled, placing a new traversal where necessary to maintain the maximum separation. The algorithm starts by placing a traversal at $t = 0$. The distance between the endpoints of the last traversal added to the path, and a prospective one at the current location of P, is calculated at each sampled location of $P(t)$. When the distance of either endpoint from the previous traversal is greater than f, the algorithm places a transit, locating it on the opposite side from the previous one, then places the prospective traversal at the location of the previous sampled location. By enforcing the distance between traversals to be less than or equal to f, the algorithm ensures complete coverage. This process repeats until the end of the path has been reached.

No two traversals are ever separated by more than f, so that two sequential passes of the UAV, one on each traversal, captures all of the area of $A(t)$ between those traversals in $M(t)$. Since all of $P(t)$ is sampled by traversals, and $A(t)$ is defined by $P(t)$, then

$$A(t) \subseteq M(t). \tag{3.2}$$

Therefore, the Conformal lawn mower completely covers the swept area, $A(t)$, defined by the path, $P(t)$. □

3.4 Coverage Efficiency

I begin by proving the performance of the Conformal lawn mower plan. Using these results, I present a solution to Problem 3.2.1. Finally, I demonstrate the suboptimality of the Conformal plan, by presenting a hand crafted alternative.

3.4.1 Proof of Efficiency

The UAV has perfect knowledge of the ground vehicle's intended path for a limited window – the UAV is given the ground vehicle's path for the range $[t_i, t_i + \Delta t]$. The UAV must create a coverage plan that ensures all of $P(t), t \in [t_i, t_i + \Delta t]$ is covered prior to expiry.

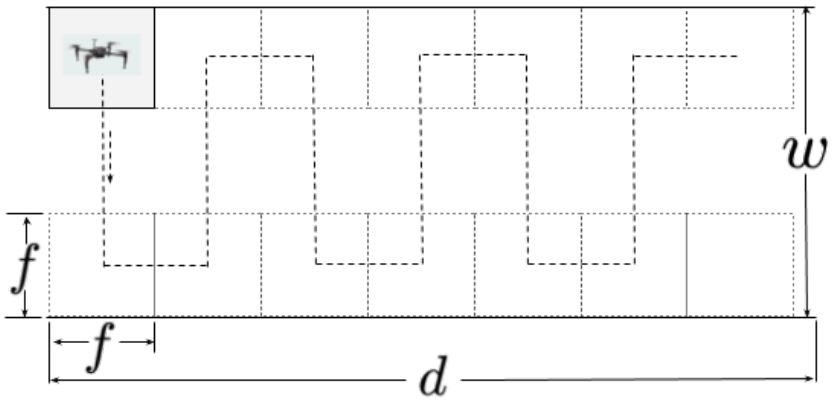


Figure 3.3: An optimal coverage plan for a straight path.

I demonstrate a worst case scenario that minimizes the distance the ground vehicle travels relative to the UAV. Based on this, I establish a sufficient relative velocity for the UAV to successfully cover any ground vehicle path within the curvature constrained set $\mathbb{P}$.

Theorem 3.4.1 (Efficiency)**.** For any path P(t) in the set $\mathbb{P}$, the Conformal lawn mower plan has a length that is no more than two times the optimal coverage plan.

To prove this result I require two preliminary lemmas.

Lemma 3.4.2 (The Optimal Straight Path Ratio)**.** For a straight path $P(t)$ (Figure 3.3), the ratio of the distance travelled by the UAV to that of the ground vehicle is $\frac{w}{f}$.

Proof. (By Construction.) The ground vehicle travels down the centre of the path, moving a distance of d. Therefore the total coverage demand is wd. The ratio of the ground vehicle velocity to the UAV is determined by

$$\frac{d}{v_{\text{gv}}} \geq \frac{wd}{fv_{\text{uav}}}.$$

Since the velocity of both vehicles is fixed, I can eliminate the time component on both sides and state this in terms of distance. Therefore, the ratio of the vehicle distances is

$$\frac{d_{\text{uav}}}{d_{\text{gv}}} \geq \frac{v_{\text{uav}}}{v_{\text{gv}}} \geq \frac{w}{f}. \tag{3.3}$$

□

Remark (Optimality of the lawn mower coverage plan)**.** The lawn mower coverage plan, illustrated in Figure 3.3, is an optimal plan for the straight path. Each traversal is $w - f$ in length and spaced f apart, meaning that for one complete period of two traversals and two movements of f, the ratio of UAV distance to ground vehicle distances is

$$\frac{d_{\text{uav}}}{d_{\text{gv}}} \geq \frac{2(w-f)+2f}{2f} = \frac{w}{f}.$$

•

The length of the UAV coverage plan for any arbitrary path cannot be any shorter than the optimal coverage plan for the straight path.

Lemma 3.4.3 (Arbitrary Paths have the Same Length)**.** An arbitrary corridor centred on a path P in $\mathbb{P}$ of length d has an optimal coverage plan at least as long as the optimal coverage plan for a straight path of the same length.

Proof. (By Construction.) I first prove an arbitrary corridor centred on P in $\mathbb{P}$ has the same area as an equivalent straight corridor with the same centre-line length. The comparison of optimal coverage path lengths flows directly from this fact.

Let S be an arbitrary corridor centred on P in $\mathbb{P}$ of length d and width w. The arbitrary path can be decomposed into a set of n curve sections, $\{s_1, s_2, \ldots, s_n\}$, where each section has a centre-line length Δd such that

$$d = \sum_{i=1}^{n} \Delta d.$$

Each segment s_i can then be approximated by a segment s_i', which has length Δd and a constant curvature equal to the maximum curvature of s_i. The concatenation of these segments $s_1', \ldots, s_n'$ creates a curve S_n. Notice that by the smoothness of paths in $\mathbb{P}$, we have $S_n \to S$ as $n \to \infty$ and thus $\Delta d \to 0$.

The total area of S_n is the sum of the areas of all n of its sections,

$$\text{area}(S_n) = \sum_{i=1}^{n} \text{area}(s_i').$$

For each section, s'_i, the area is calculated in one of two ways. If the section s_i is straight, its area is Δdw. Otherwise, letting r be one over the curvature of the section, the area of curved section s'_i is

$$\text{area}(s'_i) = \frac{\theta}{2\pi}\left((\pi(r+\frac{w}{2})^2 - (\pi(r-\frac{w}{2})^2\right) = \theta\,(rw)\,. \tag{3.4}$$

From the geometric equation for the length of an arc, $\theta = \frac{\Delta d}{r}$. Substituting into (3.4) gives

$$\text{area}(s'_i) = \frac{\Delta d}{r} rw = \Delta dw.$$

Therefore the area of S is

$$\text{area}(S) = \lim_{\Delta d \to 0} \sum_{i=1}^{n} \Delta dw = dw.$$

This is exactly the area of a straight path of length d and width w.

From Lemma 3.4.2, the UAV must travel at least w/f times as far as the ground vehicle when covering a straight path. Since the arbitrary path has exactly the same area as the straight path, it must generate exactly the same coverage demand. The UAV's ability to satisfy the coverage demand remains the same, governed by the size of its optical footprint, f. Therefore, the optimal coverage plan for the arbitrary path in $\mathbb{P}$ must be at least as long as the optimal coverage plan for the equivalent straight path. □

Remark (The need for a curvature constraint)**.** The analysis is restricted to paths in $\mathbb{P}$ whose curvature is at most $2/w$. If a path contains a curve with curvature greater than $2/w$, the deadline endpoint on the inside of the curve moves in the opposite direction of the ground vehicle motion, resulting in a reduced swept area $A(t)$. In this scenario Lemma 3.4.2 no longer holds, and thus the analysis does not follow through. •

Based on Lemma 3.4.2 and Lemma 3.4.3, Theorem 3.4.1 can now be proven.

Proof of Theorem 3.4.1. Consider a straight ground vehicle path of width w with a UAV providing mapping coverage using an optical footprint of size f. When a UAV path is constructed as a series of alternating traversals and transits, that path is maximized if the transits are all of length f, as illustrated in Figure 3.4. Since the maximum separation between traversals is less than or equal to f, to find the worst case distance ratio between the ground vehicle and the UAV, the ground vehicle distance is minimized.

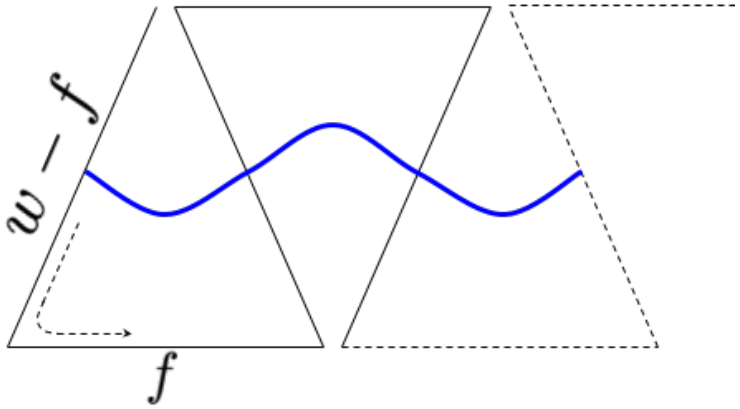


Figure 3.4: A path minimizing d_{gv} with respect to d_{uav}.

Starting with parallel traversals, incrementally increase the angle between them. As the angle is increased, the curvature of the path increases, and the length of the ground vehicle path segment between the traversals decreases. Since each traversal must cross the path at right angles, the path must be a series of alternating circular arcs, with a curvature directly dictated by the angle between the traversals. The length of the ground vehicle's path segment between two traversals can be expressed as

$$d_{\text{gv}} = r\theta = r\frac{f}{\frac{w}{2}+r}, \quad r \geq \frac{w}{2}. \tag{3.5}$$

Note that the distance in (3.5) is minimized when $r = \frac{w}{2}$.

The UAV travels the length of one traversal, followed by a transit to the next traversal. Therefore, the distance that the UAV must travel is

$$\begin{aligned} d_{\text{uav}} &= (w-f) + (\frac{w-f}{2} + r)\theta \\ &= (w-f) + (\frac{w-f}{2} + r)\frac{f}{\frac{w}{2}+r} \\ &\leq (w-f) + f = w, \end{aligned} \tag{3.6}$$

since the traversals are separated by not more than f. Therefore, the ratio of the UAV distance (3.6) to the ground vehicle distance (3.5) can be calculated

$$\frac{d_{\text{uav}}}{d_{\text{gv}}} \leq \frac{w}{r\frac{f}{\frac{w}{2}+r}} \leq 2\frac{w}{f}, \quad \text{if } r = \frac{w}{2}. \tag{3.7}$$

The upper bound on the ratio of the velocity the UAV requires, relative to the ground vehicle velocity on an arbitrary path, is then given by

$$\frac{v_{\text{uav}}}{v_{\text{gv}}} = \frac{d_{\text{uav}}}{d_{\text{gv}}} \leq 2\frac{w}{f}. \tag{3.8}$$

Applying Lemma 3.4.2 and Lemma 3.4.3 shows that the ratio of the length of the coverage plan to the ground vehicle distance for any arbitrary path must be at least $\frac{w}{f}$. Therefore for any arbitrary path in $\mathbb{P}$, following a Conformal lawn mower plan requires no more than twice the velocity necessary for the optimal coverage plan on the same path. □

Remark (Another optimal path)**.** Note that a small modification to the worst case UAV plan shown in Figure 3.4 results in the optimal plan. In particular, by traveling each traversal in the opposite direction, the UAV travels in a 'W' motion on traversals, and the transit between each traversal is 0 instead of f. The resulting path has length $\frac{w}{f}d_{\text{gv}}$, which is exactly the optimal solution. If UAV had global knowledge of the path, it could select the appropriate direction to perform the traversals. However, given its limited knowledge of the future path, this is not possible, and the resulting path can be a factor of two times longer. This illustrates the impact of limited path information on the coverage efficiency.•

Remark (A near worst case path)**.** A single curve of maximum curvature $2/w$ is in $\mathbb{P}$ and provides a ratio of distances traveled that is nearly as large as for the path in Figure 3.4. In particular, given the UAV and ground vehicle distances

$$d_{\text{gv}} = 2r\frac{f}{r+\frac{w}{2}}, \quad d_{\text{uav}} = 2r\frac{f}{r+\frac{w}{2}} + 2(w-f),$$

it is straightforward to calculate their ratio as

$$\begin{aligned}\frac{d_{\text{uav}}}{d_{\text{gv}}} &= 1 + \frac{w-f}{f}\left(\frac{r+\frac{w}{2}}{r}\right) \\ &= 1 + \frac{2(w-f)}{f} \\ &\leq 2\frac{w}{f}.\end{aligned} \tag{3.9}$$

The ratio of (3.9) is at maximum when $r = \frac{w}{2}$ and therefore reduces to

$$\frac{d_{\text{uav}}}{d_{\text{gv}}} = 1 + \frac{2(w-f)}{f} = 2\frac{w}{f} - 1.$$

•

3.4.2 The Correctness of the Conformal Lawn Mower Plan

With Theorem 3.3.1 and Theorem 3.4.1 I have shown both complete coverage of the path $P(t)$ and the sufficient velocity the UAV requires for any path in $\mathbb{P}$. I can now state the main result.

Theorem 3.4.4 (Correctness of Conformal Lawn Mower plan)**.** Consider a ground vehicle, travelling at velocity v_{gv}, with initial condition at the start of path $P(t), t = 0$. The path $P(t)$ has a coverage width of w and a maximum curvature $\frac{2}{w}$. Then, a UAV with velocity $\geq 2\frac{w}{f}v_{\text{gv}}$ and following the Conformal lawn mower path solves problem 3.2.1.

Proof of Theorem 3.4.4. For complete coverage of $P(t)$, the region $A(t)$ swept out by $D(t), t \in [0, t_{\max}]$, must be entirely within the mapped area $M(t)$ before expiry. Based on Theorem 3.3.1, $M(t)$ contains all of $A(t)$. It remains to prove that no elements of $A(t)$ expired before they were included within $M(t)$.

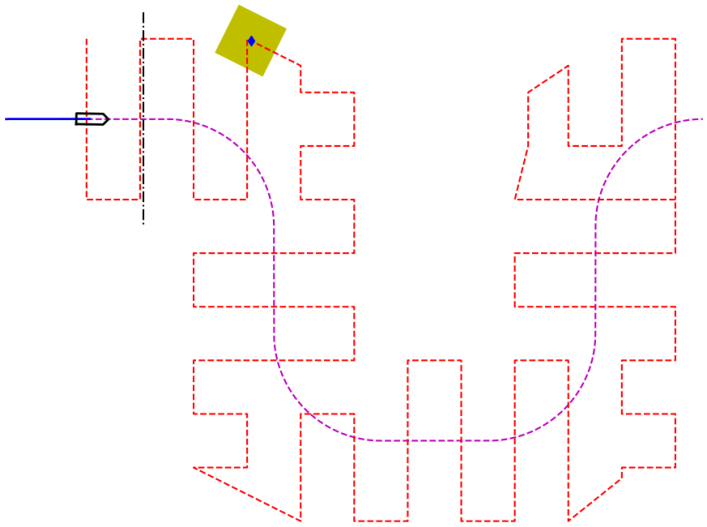
From the initial conditions, the UAV starts ahead of the deadline with at least one completed traversal already mapped before the ground vehicle starts moving. For all remaining elements of $A(t)$ to be mapped correctly, I only need to show that the UAV maintains or extends its position ahead of the ground vehicle. If the UAV uses a velocity that is at least $\frac{2w}{f}$, where w is the width of the path and f the UAV's optical footprint, then Theorem (3.4.1) asserts this is true.

Therefore, all of $A(t)$ is successfully mapped, and Problem 3.2.1 is solved. □

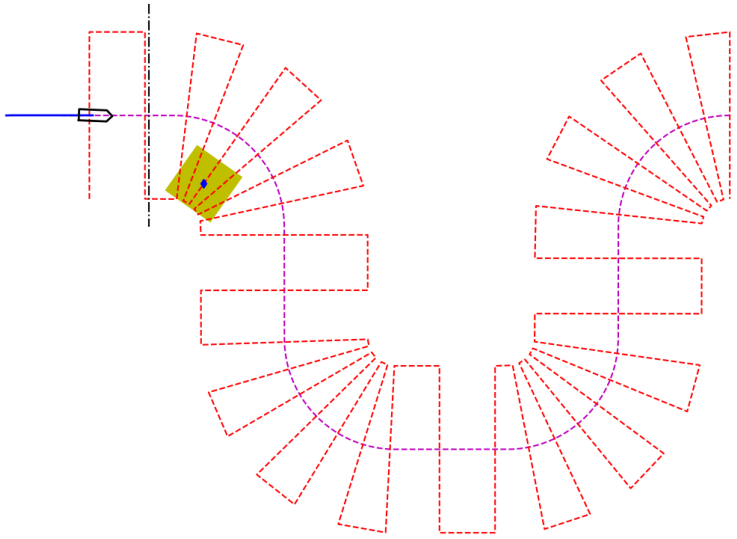
3.4.3 Suboptimality of the Conformal Lawn Mower Plan

In section 3.4, the Conformal lawn mower plan is shown to be within a factor of two of the optimal plan. For some paths in $\mathbb{P}$ there may be more efficient coverage solution that minimizes the ratio $\frac{v_{\text{uav}}}{v_{\text{gv}}}$. With full path knowledge, a coverage plan may be proposed that reduces the scanning overlap, and requires a lower sufficient velocity from the UAV as a result. Consider a path $P(t)$ with coverage width $w = 400$ and a maximum curvature of $\frac{1}{200}$. A hand crafted coverage plan for $P(t)$ is presented in Figure 3.5a, with the equivalent Conformal plan in Figure 3.5b. Simulation results of both plans are presented in Table 3.1. From these results, the hand crafted path reduces the minimum required by 7 m/s. The hand crafted plan, while not necessarily optimal, is clearly an improvement.

Note that finding the optimal path appears to be an NP-hard problem. According to [3], the lawn mowing problems are NP-hard in general; however, whether this formulation of the problem with the additional constraints is NP-hard is the subject of further investigation.



(a) Hand crafted



(b) Conformal

Figure 3.5: Two Coverage plans over a path with width 400m, curvature $\frac{1}{200}$.

Table 3.1: A comparison of Conformal vs. Hand crafted plans.

UAV	Conformal		Hand crafted	
Velocity	Distance	%Coverage	Distance	%Coverage
20	8000	22	8000	65
21	8400	23	8400	89
22	8800	28	8800	100
23	9200	35	9027	100
24	9600	43	9038	100
25	10000	53	9040	100
26	10400	70	9036	100
27	10800	89	9042	100
28	11200	99	9044	100
29	11588	100	9038	100
30	11781	100	9039	100

3.5 Simulation Results

Simulations were run for two scenarios, varying the UAV velocity on different types of ground vehicle paths (straight, decreasing curvature, and randomly generated), and using a single velocity while progressively decreasing the curvature. For all simulations, the fixed parameters are: $v_{\text{gv}} = 5$ m/s, $w = 400$ m, $f = 100$ m. In Table 3.2 the results of several simulations are shown. The control case, a straight path, reaches full coverage between 20 and 21 m/s, as expected if allowances are made for slight rounding errors in the simulation. The simulations used a random ground vehicle path with minimum curvature of 1/200, as well as curvatures ranging from 1/200 to 1/1000. The full simulation results are displayed in Figure 3.6, while illustrations of some of the test paths can be seen in Figure 3.7.

In all cases, the UAV and the ground vehicle start on the left side of the path, with the dark grey areas indicating successful mapping. Areas that are light grey expired before the UAV was able to cover them. As expected, all of the paths show increasing degrees of success as the UAV velocity is increased. The results are summarized in Table 3.2, showing the first velocity where full coverage was achieved. These tests also illustrate the relation between the curvature of the path and the success rate. For a given velocity, as the curvature of the path is increased, the success rate at mapping decreases, matching expectations.

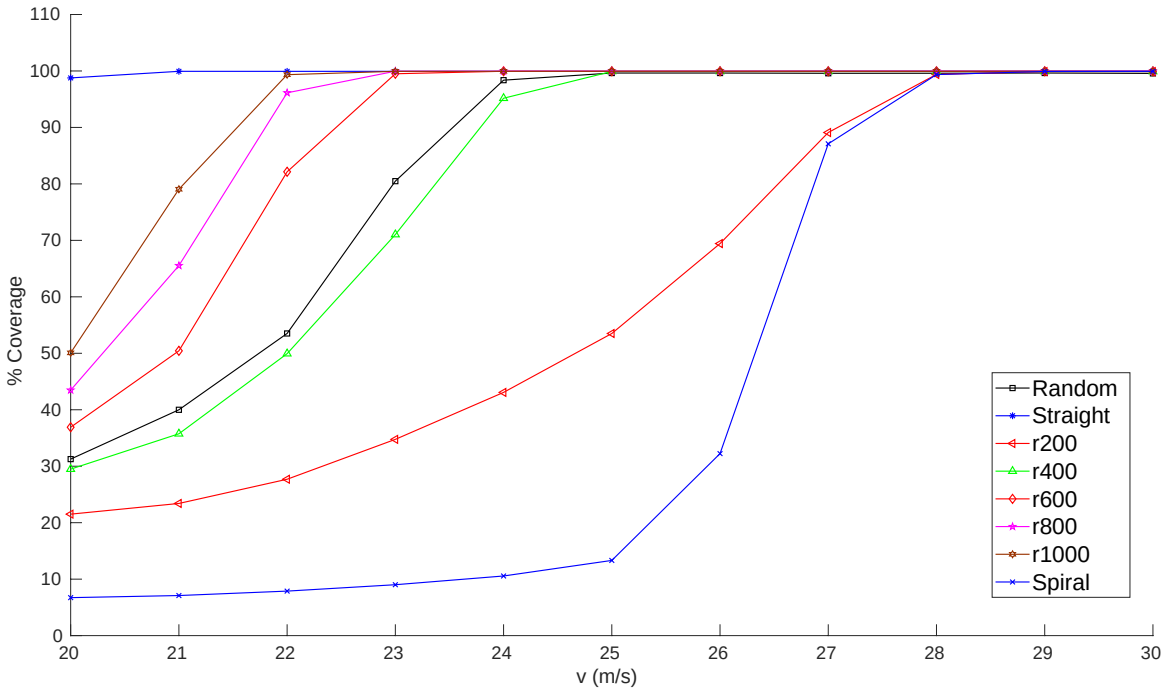
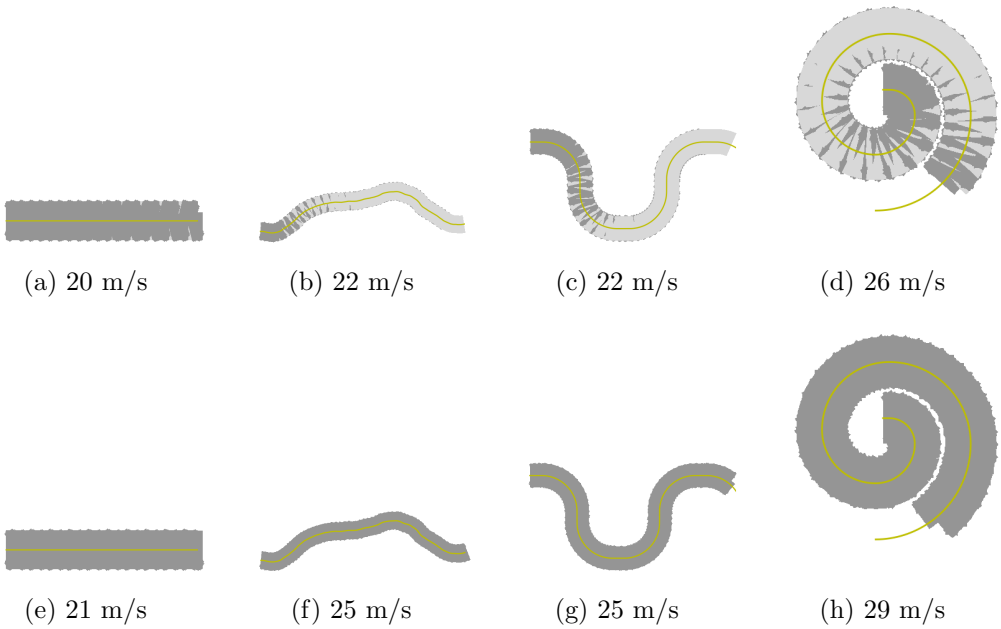


Figure 3.6: Percent coverage as a function of UAV velocity and path type.



(a) 20 m/s (b) 22 m/s (c) 22 m/s (d) 26 m/s

(e) 21 m/s (f) 25 m/s (g) 25 m/s (h) 29 m/s

Figure 3.7: Increasing UAV velocity over various path configurations.

Simulations also looked at the effect on the UAV as the curvature of the path decreases, using the path modeled in Figures 3.8a-3.8b. As the curvature of the path decreases (i.e., the radius is increasing), the UAV becomes progressively more successful in mapping the path for a given UAV velocity. This increasing rate of success can be directly attributed to the reduction in overlap of the UAV sensor footprint while following the Conformal plan as the ground vehicle path straightens out.

3.6 Conclusions

In this chapter, the problem of providing path planning coverage for a moving ground vehicle was defined. Minimum performance requirements were developed for the UAV to provide timely coverage of the path immediately ahead of ground vehicle as it travels through the environment. The plan that was developed, a variation of the classic lawn mower plan, ensures complete path coverage without prior knowledge of the entire plan. I have shown that the Conformal lawn mower path can be no longer than twice the length of the optimal plan, and therefore require no more than twice the UAV velocity, for any path with maximum curvature constrained less than $2/w$.

Table 3.2: A comparison of velocity vs. path type, showing the minimum velocity for complete coverage.

Path	V (m/s)	% Coverage
straight	21	99.96
r200	29	99.89
r400	25	99.91
r600	25	99.95
r800	23	99.96
r1000	23	99.94
random	25	99.63
spiral	29	99.92

(a) r200 m (b) r400 m

Figure 3.8: Coverage Results at 25 m/s – dark grey is successful coverage, light grey expired. As the minimum path radius is increased, the UAV successfully covers a larger fraction of the total area.

One observation from this work is that high UAV velocities are needed to successfully map a region, given a reasonable ground vehicle velocity. In the simulations, the velocity of the ground vehicle was limited to 5 m/s (or about 20 km/h) over a corridor width of 400 m, and a UAV footprint of 100 m $\times$ 100 m. With those parameters, the minimum required velocity for the UAV to be successful was 20 m/s, well in excess of the capability of most rotor-based UAVs, particularly for sustained flight.

Now that the problem has been defined, what happens in the extreme case where the ground vehicle shares no information about its planned route? This question is the focus of the work on predictive mapping in the next chapter.

Chapter 4

Predictive Mapping for a Moving Vehicle

The path planning problem for predictive coverage is challenging, given that the UAV has no knowledge of the ground vehicle's intended path. The UAV must predict the future path by calculating the reachable set of possible paths and endpoints that the ground vehicle could potentially move to from its current location, limited by the required map distance d_{map}. This chapter considers how the set of possible routes shifts over time as the ground vehicle moves forward and takes new orientations in the environment. Based on the ground vehicle's reachable set, the lower and upper bounds on the possible territory that must be mapped for the UAV to be successful are derived. Finally, based on the bounds, both necessary and sufficient relative velocities for the UAV are calculated as a ratio of the ground vehicle velocity.

4.1 Problem Statement

The problem setup is similar to that of Chapter 3, with one exception: the UAV is only provided with the ground vehicle's historical position data (physical location in the environment and orientation), and from that data must predict the ground vehicle's future path.

The problem may be summarized as follows.

Problem 4.1.1 (Necessary UAV Velocity)**.** A ground vehicle travels through an environment in $\mathbb{R}^2$, following a path $P(t)$, and creates a coverage demand of $A(t)$. A UAV travels

ahead of the ground vehicle producing a coverage area of $M(t)$. Determine the necessary UAV velocity that guarantees

$$A(t) \subseteq M(t), \forall t \in [0, t_{\max}]. \tag{4.1}$$

Sub Problem 4.1.2 (Sufficient UAV Velocity and Candidate Policy)**.** Assume the UAV only has access to the ground vehicle's historical route data. Determine a coverage policy that satisfies (4.1).

4.2 The Boundary of the Ground Vehicle's Reachable Set

The reachable set encompasses the end points of all the possible trajectories a ground vehicle may follow, starting at the origin and travelling at a constant velocity for a defined interval. For a Dubins vehicle, the outer boundary of the reachable set has a closed form expression [17], assuming that the vehicle is initially located at $(x, y, \theta) = (0, 0, \frac{\pi}{2})$:

$$x(\theta) = \rho(1 - \cos(\theta)) - \sin(\theta)(v_{\text{gv}}t - \rho\theta), \tag{4.2}$$
$$y(\theta) = \rho\sin(\theta) - \cos(\theta)(v_{\text{gv}}t - \rho\theta), \tag{4.3}$$

where

- ρ is the minimum turning radius of the ground vehicle,
- θ is the final angle of the ground vehicle trajectory, $0 \leq \theta \leq \frac{v_{\text{gv}}t}{\rho}$
- v_{gv} is the velocity of the ground vehicle, and
- t is the interval over which the movement occurs.

These equations describe the boundary for $x > 0$. This boundary and its mirror in the y-axis are the ground vehicle's reachable set.

Given that the UAV does not have prior knowledge, any point in the reachable set could be on the ground vehicle's actual path. Therefore, in order to determine the set of possible deadlines, the entire territory within the reachable set must be covered by the UAV. Let t_{map} be the time the ground vehicle requires to travel the map distance, $t_{\text{map}} = \frac{d_{map}}{v_{gv}}$.

Let $R(t + t_{map})$ be the set of all possible end points (at some (x, y, θ)) for a trajectory of duration t_{map}. This R is the coverage reachable set, defined by the map distance, the coverage width, and the ground vehicle's movement parameters and expressed as

$$R(t + t_{map}) \subseteq \mathbb{R}^2 \times S^1.$$

At each point in $R(t+t_{map})$, there is a deadline perpendicular to the orientation θ. The set of all possible deadlines is then

$$\mathscr{D}(t) = \{x \mp l \sin\theta, y \pm l \cos\theta | (x, y, \theta) \in R(t + t_{map}), \\ l \in [-\frac{w}{2}, \frac{w}{2}]\}.$$

The full derivation of the boundary for deadlines tangent to the reachable set can be found in Appendix A.1.

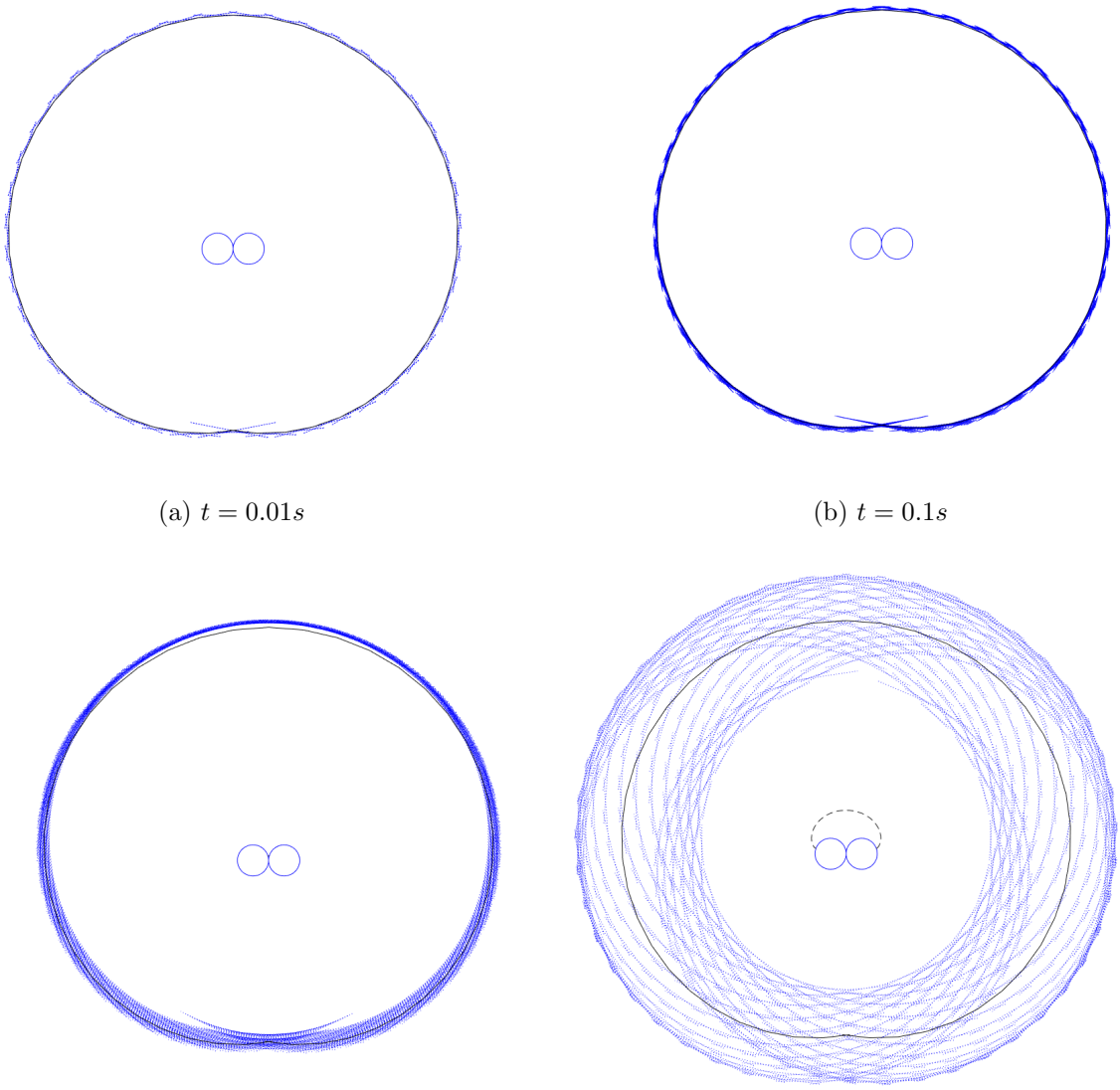
Calculating the exact outer boundary that contains all of the possible deadlines is computationally difficult. Alternatively, lower and upper bounds can be estimated, such that

$$\underline{D}(t) \subseteq \mathscr{D}(t) \subset \overline{D}(t),$$

where $\underline{D}(t)$ and $\overline{D}(t)$ are the lower and upper bounds respectively.

4.2.1 Deadline Lower Bound

The deadline lower bound, $\underline{D}(t)$ is based on the boundary of the reachable set. Consider the case where the deadline is located tangent to the boundary at t_i. At the next time step, t_{i+1}, as the ground vehicle moves forward, the deadline likewise moves forward into unmapped territory. Since the UAV doesn't know the deadline's actual position, it must assume the deadline could be anywhere on the boundary. Consequently, the mapping demand must be the entire leading edge of the boundary. The images in Figure 4.1 show how the mapping demand generated by the ground vehicle's reachable set (defined by v_{gv}, ρ, and t_{map}) is affected by translation and rotation over time intervals of 0.01s, 0.1s, 1s, and 10s. Notice that the outer bound of the mapping demand expands outward along the entire boundary with increasing time.



(a) $t = 0.01s$

(b) $t = 0.1s$

(c) $t = 1s$

(d) $t = 10s$

Figure 4.1: Possible deadlines shifted by increasing values of t_{mov}. The small (blue) circles at the centre represent the minimum turning radius of the ground vehicle.

Define the lower bound, $\underline{D}$, as an arc of maximum radius that fits entirely within the set of tangent deadlines. The minimum radius is calculated[1] as

$$r_{\underline{D}} = r_{\text{gv}} + \sqrt{r_{\text{gv}}^2 + \left(\frac{L}{2} + \frac{\sqrt{L^2 - 4r_{\text{gv}}^2}}{2}\right)^2}, \tag{4.4}$$

where L is the distance from either end of the deadline to the ground vehicle, when the deadline is located at top dead centre. L is calculated

$$L = \sqrt{d_{map}^2 + \frac{w^2}{2}} - \frac{\pi}{2} r_{\text{gv}}.$$

Since an area is only mapped once, and since the initial reachable set is already mapped, only the new area on the leading edge needs to be considered ($y > 0$ if the ground vehicle is at $(0, 0, \frac{\pi}{2})$).

With the lower bound defined, the necessary conditions for the UAV's velocity relative to the ground vehicle can be established.

Lemma 4.2.1 (Necessary Velocity)**.** The UAV must cover territory as fast or faster than the movement of the boundary of ground vehicle's reachable set can generate new coverage demand. The smallest amount of demand that the ground vehicle can generate is defined by the length of the lower deadline bound, $\underline{D}$, and therefore the necessary velocity of the UAV relative to the ground vehicle must be at least

$$v_{\text{uav}} = \frac{\pi r_{\underline{D},est}}{f} v_{gv}. \tag{4.5}$$

Proof. (By Construction.) To establish the necessary conditions, the argument from Chapter 3, Theorem 3.4.1 is used: the UAV must consume (map) new territory at the same rate (or faster) than the ground vehicle can create it. Therefore, the ratio of the speeds of the two vehicles can be established by comparing their rates of generation/consumption. The length of the arc is

$$d_{gv\ boundary} = \pi r_{\underline{D},est},$$

and the ground vehicle generates mapping demand at $\pi r_{\underline{D}(t)} v_{gv}$. The UAV has an optical footprint of f and consumes mapping demand at a rate of $f v_{\text{uav}}$. Therefore, equating these two functions and rearranging,

$$v_{\text{uav}} = \frac{d_{gv\ boundary}}{d_{\text{uav}}} v_{gv} = \frac{\pi r_{\underline{D},est}}{f} v_{gv} \tag{4.6}$$

[1] (See Appendix A.2 for a detailed derivation).

□

Note that for increasing values of v_{gv} this ratio escalates quickly.

4.2.2 Deadlines Upper Bound

To calculate the upper bound of deadlines, $\overline{D}(t)$, start by considering the maximum projection of a deadline beyond the boundary of the ground vehicle's reachable set.

Lemma 4.2.2 (Deadline Upper Bound)**.** The set of all deadlines, $\mathscr{D}(t)$, is bounded by an arc with a radius defined by the coverage distance d_{map} and the width of the deadline w. The radius is defined

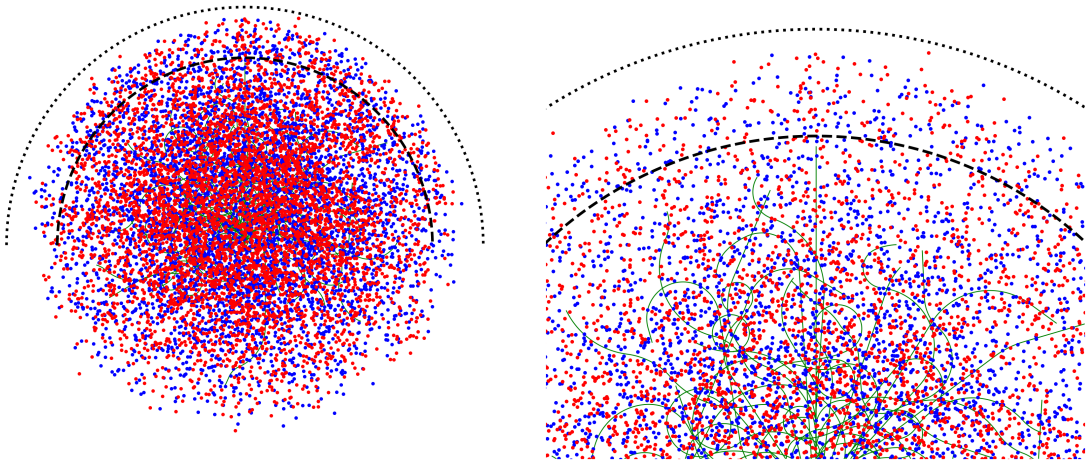
$$r_{\overline{D}} = d_{map} + \frac{w}{2}. \tag{4.7}$$

Proof. (By Construction.) A ground vehicle moves through its reachable set, following a trajectory ending at x. There are two cases: x is on the boundary of the reachable set, or x is a point inside the boundary. If x is on the boundary, then the deadline associated with x is tangent to the boundary. The perpendicular distance from the endpoints of this deadline to the boundary is less than $\frac{w}{2}$. If x is inside the boundary, no matter what the ground vehicle's final orientation, the maximum extent of one of the endpoints occurs when the deadline is perpendicular to the boundary. If the extreme is assumed – that somehow the ground vehicle path terminated at the boundary, with an orientation tangent to the boundary – then the maximum extent of the deadline is a perpendicular distance no more than $\frac{w}{2}$ from the boundary. However, it is not possible for a vehicle with an orientation parallel to the boundary to be on the boundary – any such vehicle must be inside the reachable set. Therefore, the furthest possible extension of any deadline past the boundary is $\leq \frac{w}{2}$.

From the calculation of the boundary, the furthest straight line distance of the boundary from the ground vehicle occurs when $\theta = 0$. Therefore, the ground vehicle reachable set is fully contained within an an arc of radius d_{map}. If the radius of this arc is extended by $\frac{w}{2}$, it is guaranteed to fully enclose $\mathscr{D}(t)$. Therefore, define

$$r_{\overline{D}} = d_{map} + \frac{w}{2}.$$

□



(a) Deadline upper and lower bounds

(b) Enlarged View

Figure 4.2: Sampling the Possible Deadlines. For reference, arcs representing the upper and lower bounds are added in black.

4.2.3 Deadlines Illustration

To illustrate the upper and lower bounds, I sampled the deadlines of 5000 random paths as shown in Figure 4.2. Each deadline has a left and right endpoint, drawn in red and blue respectively. Some sample paths in green are included for reference.

4.3 Predictive Path Planning

In order to successfully cover the ground vehicle's intended path, all of $\mathscr{D}(t)$ must be covered before those points expire. Based on Lemma 4.2.2, the upper bound $\overline{D}(t)$ is used to determine a sufficient velocity. Taking the UAV's camera footprint as a practical interval, the UAV must have completed at least one pass ahead of $\overline{D}(t)$, and therefore must follow an arc of radius $\overline{D}(t) + \frac{3f}{2}$.

The predictive plan maintains the UAV's position by monitoring the current heading and velocity of the ground vehicle. Any changes in trajectory are applied to the UAV's path such that the arc is rotated to stay in front of the ground vehicle's projected path. The resulting coverage policy is illustrated in Figure 4.3. Let $r_{\text{uav}} = (r_{\overline{D}} + \frac{3f}{2})\pi$ be the radius of the UAV's path. The UAV's velocity requirements can be divided into two components:

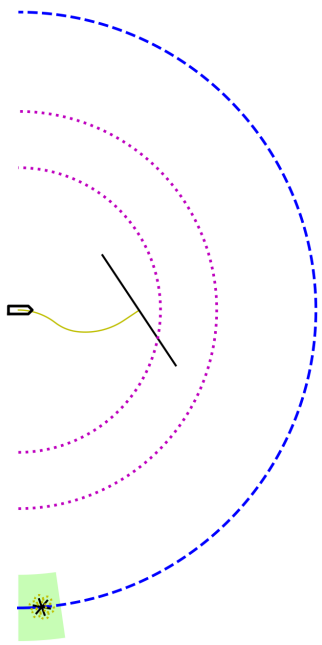


Figure 4.3: Coverage using an Arc Path. The UAV traces an arc one footprint, f, away from the set of possible deadlines, $\overline{D}$.

velocity required to cover the arc, and velocity required to maintain position. For the arc, the UAV must complete a pass in the same time that the ground vehicle moves forward f distance. The resulting minimum velocity of the UAV can be calculated as

$$v_{\text{uav}} = \frac{r_{\text{uav}}\pi + f}{f} v_{\text{gv}}. \tag{4.8}$$

To account for rotation, the UAV must add additional velocity as the orientation of the ground vehicle changes. This allows the UAV to maintain its position relative to the ground vehicle. Assuming the ground vehicle rotates by $\theta_{\text{gv}} = \frac{v_{gv}\text{dt}}{\rho}$, where ρ is the minimum turning radius of the ground vehicle. Ignoring the complication of overlapping map coverage[2], then in the same interval, the UAV must cover the same angle. Assuming the ground vehicle turns at constant rate, the angle of rotation of the ground vehicle is $\theta_{\text{gv}} = \frac{v_{gv}dt}{\rho}$. Similarly for the UAV, the angle of rotation is calculated $\theta_{\text{uav}} = \frac{v_{\text{uav}}dt}{r_{\text{uav}}}$. Setting the angles equal, the UAV velocity must increase by $\frac{r_{\text{uav}}}{\rho} v_{gv}$ in order to maintain its position as the ground vehicle turns.

Therefore, the full ratio of UAV velocity to that of the ground vehicle can be stated as

$$v_{\text{uav}} = (\frac{r_{\text{uav}}\pi + f}{f} + \frac{r_{\text{uav}}}{\rho}) v_{\text{gv}}.$$

For a map distance of $d_{map} = 500, w = 200$, this translates to a velocity ratio of

$$v_{\text{uav}} \approx 60 v_{\text{gv}}.$$

At a ground vehicle velocity of 5 m/s, the UAV would have to travel $\sim 300 m/s$ to map all possible paths successfully.

4.4 Simulation Results

Simulations were performed on several different paths. For each path, the simulation is first run using a naive lawn mower path (see Figure 4.4), using a velocity of 300 m/s. The remaining columns are the same paths, covered with increasing UAV velocities until coverage is successful. The high velocity for the naive coverage simulations was selected

[2]This is assuming that the UAV is not taking advantage of the fact that it may have already mapped this territory on an earlier pass.

to demonstrate that there is no velocity at which the naive solution can be completely successful when there is no path information available from the ground vehicle.

The simulation parameters are:

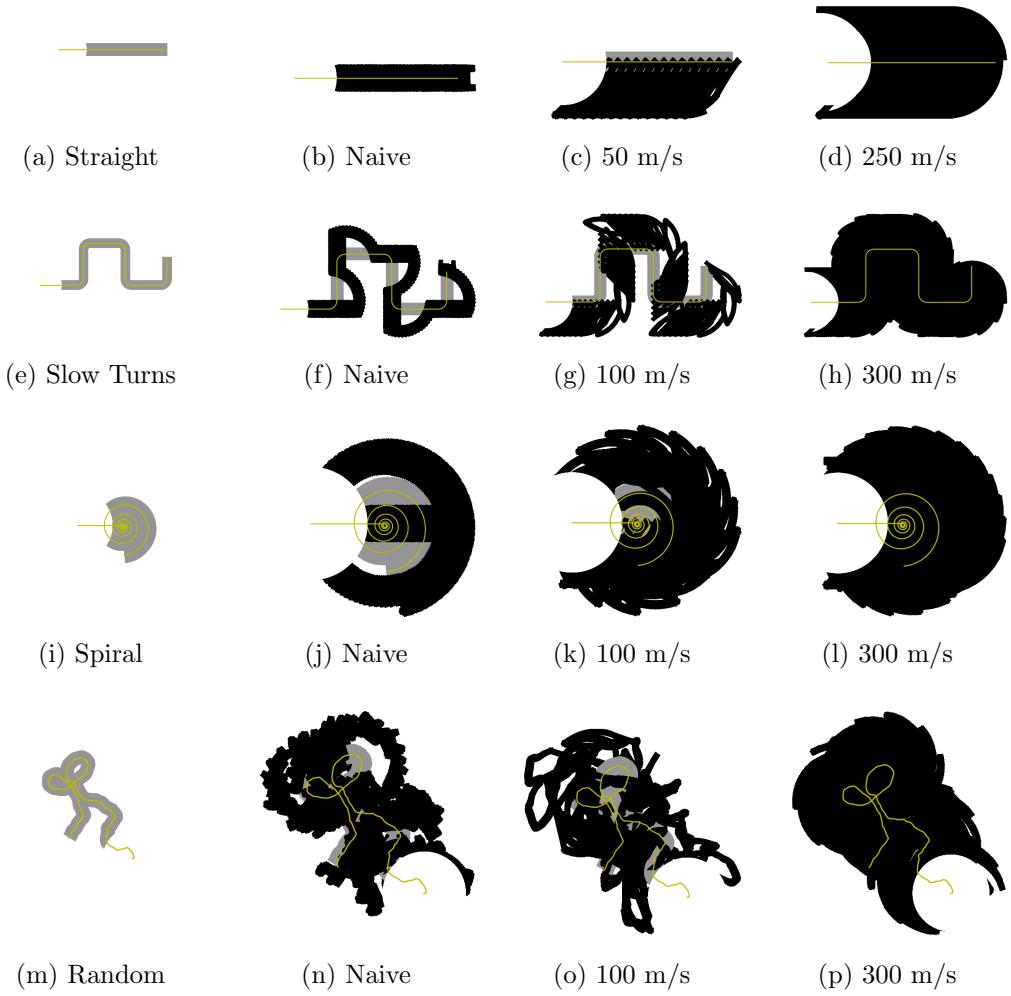
$$v_{\text{gv}} = 5.66m/s, \quad d_{\text{map}} = 500m, \quad w = 300m, \quad f = 100m$$

Note that with the exception of the straight path, the naive planner is unable to provide complete coverage, despite the relatively high velocity used.

4.5 Conclusions

In this chapter, minimum performance requirements were developed for successful coverage when only limited information with respect to the ground vehicle is available. With no prior knowledge of the ground vehicle's path, the UAV must cover a much larger area than necessary, and subsequently requires a much higher UAV velocity to be successful.

Clearly the velocities required for successful coverage are extremely high relative to typical UAV capabilities. One possible solution to this problem is to limit coverage to only those areas that may be more important, as determined by evaluating obstacles, ground conditions, and vehicle parameters. I consider this approach in Chapter 5.



(a) Straight (b) Naive (c) 50 m/s (d) 250 m/s

(e) Slow Turns (f) Naive (g) 100 m/s (h) 300 m/s

(i) Spiral (j) Naive (k) 100 m/s (l) 300 m/s

(m) Random (n) Naive (o) 100 m/s (p) 300 m/s

Figure 4.4: Simulation of Predictive Planning. The first column illustrates the ground vehicle path. The second column shows the results of using a naive approach of covering only the expected corridor at any given time. The third and fourth columns illustrate the coverage improvement as the UAV velocity is increased. Note the excessive coverage required for complete success.

Chapter 5

Selectively Covering the Ground Vehicle Path

Chapters 3 and 4 studied the problem of providing online mapping of the terrain ahead of moving vehicle, but found the mapping demands generated by the ground vehicle can quickly grow beyond the capabilities of many UAVs. Outside of some limited circumstances, the problem is physically difficult, either restricting the ground vehicle to extremely slow speeds, or tightly limiting the scope of the mapped area, or requiring a UAV capable of high velocities for extended periods of time. As an alternative to the previous method of covering the entire path, in this chapter I propose an alternative approach: selectively mapping only those portions of the environment that are deemed necessary.

In applications such as search and rescue, disaster recovery, or transportation of goods and personnel in hostile environments, a ground vehicle may have to traverse uncertain and potentially dangerous terrain. The operators of the ground vehicle have a map, but it may differ from the current conditions, and any obstructions such as buildings may conceal those differences until it is too late to react. A tree may have fallen, a bridge collapsed, or there might even be adversaries hidden behind a building. A UAV, with its greater speed, mobility and flight capability [10], can stay ahead of the ground vehicle and provide real-time imagery for route planning and mapping. With real-time information from the UAV, the ground vehicle can better adapt its route as the situation demands.

In Chapter 3, I considered a precursor problem where the operators of a ground vehicle defined a view corridor along and to either side of the upcoming route. The UAV provided complete coverage of the corridor, maintaining a position ahead of the moving ground vehicle. However, the UAV speeds required to stay ahead of the ground vehicle and cover

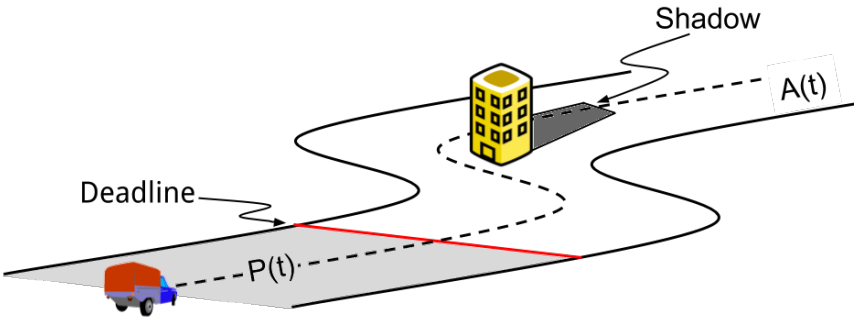


Figure 5.1: The coverage corridor projected ahead of the ground vehicle with the observation area highlighted in light grey. The observation area moves forward with the ground vehicle.

the corridor completely are very large. As a result, the ground vehicle must either be restricted to very slow speeds, or the UAV can cover only a narrow corridor around the planned route. Motivated by this drawback, in this chapter I consider the problem in which the UAV is tasked with covering only the parts of the corridor that cannot be observed directly from the ground vehicle due to occlusions.

As the ground vehicle moves through the terrain, any hidden regions, those occluded parts of the corridor, must be visited in advance by the UAV. See Figure 5.1. Hidden regions are approximated by finding areas that are never within the visibility polygon when viewed from locations along the ground vehicle's route [34, 27]. The planner must find a UAV path that covers as much of the hidden areas as possible within the resource and time constraints available: this is an Orienteering Problem with Time Windows (OPTW) [71]. Similar to the approaches of [65, 39], I cast a Mixed Integer Linear Program (MILP) to find a solution. Given that OPTW problems are proven to be NP-hard [40], I expect this formulation is likely to be NP-hard as well.

In this paper I focus on the problem of providing coverage for a fixed ground vehicle route. I envision this being used as part of a receding horizon planner that replans the coverage path when the ground vehicle's route is updated due to newly acquired information on hidden regions. However, I leave the development of a receding horizon planner for future work.

5.1 Related Work

In orienteering problems, an agent must collect some optimal subset of rewards given constraints on limited resources. The problem may be one of exploration with a UAV [55], path planning for optimal views [60], or even team oriented planning [67].

The Vehicle Routing Problem with Time Windows (VRPTW) is a related problem where the agent has a capacity that limits the number of vertices that can be visited before returning to the start, resulting in a route that takes the form of multiple loops [39]. However, the focus is on optimal scheduling and minimal travel distances, not maximizing a potential reward.

The problem of vehicle routing with time constraints also appears in [5, 9]. In each case, an efficient heuristic solution is developed after the time constraints are discretized and the problem is reduced to a directed acyclic graph (DAG). In this case, reframing the problem in as a DAG isn't possible as the resulting graph would have cycles.

5.2 Problem Statement

Let $\mathscr{E}$ be a planar environment in $\mathbb{R}^2$. The environment contains a set of m polygonal obstacles $O_1, \ldots, O_m$ that could be buildings, or other objects. Each obstacle is assumed to be tall enough that it blocks the ground vehicle's line of sight. The ground vehicle follows a collision free route $P(t)$ through $\mathscr{E}$ with velocity V_{gv} for $t \in [0, T]$ as shown in Figure 5.1. The area of the environment immediately ahead of the ground vehicle and centred on the route is defined as the observation area. The observation area has width w and extends distance d in front of the ground vehicle. As the ground vehicle traverses P, the observation area moves forward as well.

The leading edge of the observation area is denoted as the deadline. As the ground vehicle moves forward along P, the deadline remains a distance d ahead. If the time required for the ground vehicle to traverse d is defined as $\Delta t = \frac{d}{v_{\text{gv}}}$, then the deadline is at $P(t_d)$ where $t_d = t + \Delta t$.

Define the region of $\mathscr{E}$ that is swept by the deadline from time 0 to a time $t \in [0, T]$ as the coverage corridor $A(t)$. The close time t_{close} for any point in $A(t)$ is defined as the first time the deadline sweeps over that point. An open time is also defined for each point, which captures the earliest time that an observation of the point is useful to the ground vehicle. The open time is relative to the velocity of the ground vehicle and the close time: $t_{\text{open}} = t_{\text{close}} - d/v_{\text{gv}}$.

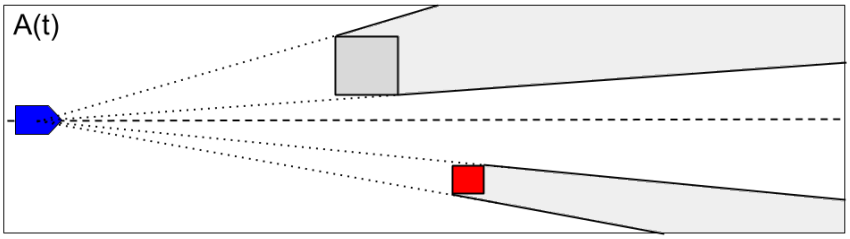


Figure 5.2: Illustration of the visibility region for the ground vehicle, used to calculate the hidden regions.

Let $V(t)$ be the set of points in $\mathscr{E}$ visible by the ground vehicle when located at the point $P(t)$ on the path, as illustrated in Figure 5.2. A point $p \in A(T)$ is said to be *hidden* if it is has not been observed by the ground vehicle prior to the deadline passing it. Let R_{hidden} denote the *hidden region*, which is the set of all hidden points in the coverage corridor $A(T)$. More formally, consider a point $p \in A(T)$ and let t_d be the time when the deadline passes p. Then, p is hidden if

$$p \notin V(t) \quad \text{for all } t \in [0, t_d - \Delta t].$$

The UAV flies over $\mathscr{E}$ with velocity V_{uav}. I assume that the UAV flies at sufficient height that its environment is obstacle-free. Thus, it can travel over the obstacles $O_1, \ldots, O_m$. I also assume that the UAV model is such that there is a steering function available that generates a path between any two points in $\mathscr{E}$. The UAV has a downward focused camera providing coverage information to the ground vehicle. The functional size of the sensor footprint on the ground is $f \times f$. Finally, I assume the velocity of the UAV is greater than that of the ground vehicle.

With this background, the primary problem can be stated.

Problem 5.2.1 (Selective Coverage Path)**.** Given a ground vehicle moving through the environment along the route P, plan a path for a UAV to maximize the area of the points within R_{hidden} that are covered within their time windows.

Solution Approach First the continuous environment $A(T)$ is converted into discrete cells and the hidden regions R_{hidden} are computed. Next I utilize sweep lines to cover each connected component of the hidden region. Then a graph with vertices representing sweep lines and edges representing transitions is constructed. And finally, a MILP is cast to solve for a path that maximizes the total coverage.

I also develop a divide and conquer approach of partitioning the graph into subgraphs while provably preserving the overall optimal solution. After which I describe a method of lossy partitioning that allows for even smaller subgraphs, while providing a guarantee on the maximum amount of lost coverage. The full UAV path is then constructed by concatenating the paths found in each subgraph in temporal order, starting with the earliest.

5.3 Development of MILP Solution

I begin by describing the method for finding the hidden regions and placing the coverage sweep lines. Then, given the set of coverage lines, I derive a graph representation and subsequently solve for optimal coverage using a MILP.

5.3.1 Placing Coverage Lines

The environment $\mathscr{E}$ is first discretized into cells. The size of each cell is determined by the accuracy desired and the computational time available. Each cell uses t_{open} and t_{close} of its centre. Next the hidden regions, R_{hidden}, are calculated by sampling the views along the ground vehicle route P and converted into a cellular representation. A cell is considered to be within a hidden region if its centre is within the boundary of the region. Finally, the enclosing hulls are calculated for the cells in each hidden area. The result is a set of polygonal approximations of the hidden regions and the cells they contain.

Parallel line segments, separated by f, are placed to provide complete coverage of each polygon found for the hidden regions. To minimize resource consumption [19], the segments are placed with an orientation that is perpendicular to the minimum altitude line of each polygon [58, 7]. The resulting coverage lines, when followed by the UAV, completely cover the polygon. A brief greedy optimization is then applied to reduce the total coverage distance required. Any two adjacent convex polygons are merged if the overall coverage path distance is reduced. This is performed by comparing the Travelling Salesperson Problem (TSP) tour of the coverage lines in the individual polygons with a TSP of a set of coverage lines that covers the convex hull that encloses them both. The merging process continues until no new improvements are found.

The closing time for a sweep line is the latest time before which the UAV will be able to successfully cover that line. Closing times must take into account both the earliest closing time of all the cells that fall within the the UAV sensor footprint as the UAV transits the

line, and the time required to transit the entire line. The closing time for the ith sweep line is defined as

$$c_i = \min \left\{ t_{\text{close}} \text{ of cells within } \frac{f}{2} \right\} - \frac{|\text{length of sweep}_i|}{v_{\text{uav}}}.$$

The opening time for a sweep line is the earliest closing time minus the offset d/v_{gv}, written

$$o_i = \min \left\{ t_{\text{close}} \text{ of cells within } \frac{f}{2} \right\} - \frac{d}{v_{\text{gv}}}.$$

If the UAV arrives at a sweep line prior to this interval, the UAV must wait, adding idle time to the solution. Note that, by definition, the length of any sweep line is limited such that $o_i < c_i$.

The reward R_i for sweep line i is the number of cells that are covered when the UAV transits that line.

5.3.2 Graph Representation

The coverage problem is represented as a directional graph with time constraints, parameterized as $G = (V, E, s, t)$ with time windows (o, c) and rewards R. There is a pair of vertices in V for each of the two directions that a line may be traversed (See Figure 5.3). There is also one vertex for each of the start and finish positions. The open and close times for each vertex are o and c respectively, and when referring to a vertex v the times are written as v_{open} and v_{close}.

The edges E of G are not fully connected. Given two vertices a and b, then edge $\{a, b\}$ is added to the graph if vertex a and vertex b do not represent the same sweep line, and if $a_{\text{open}} + s_a + t_{a,b} \leq b_{\text{close}}$, where s_a is the service cost of a and $t_{a,b}$ is the transit cost for edge $\{a, b\}$. The service cost s_a is the time required to traverse the sweep line represented by a. Since visiting a vertex necessitates a physical transition in space, the travel costs between vertices are asymmetric. The last parameter, R, is the reward acquired for visiting a vertex.

NP-hardness This formulation of the OPTW problem can be seen to be NP-hard as it contains the Euclidean TSP as a special case when the time windows for each vertex are $[0, \infty)$ and the segment lengths are zero.

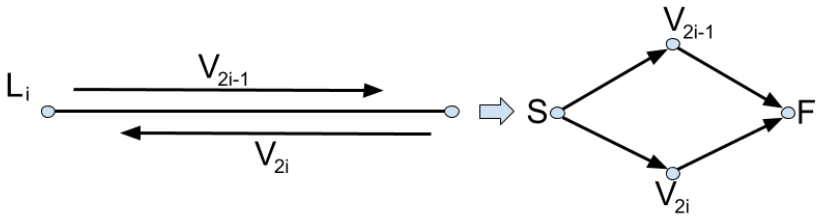


Figure 5.3: Every coverage line L_i has two associated directional vertices in the graph representation, v_{2i-1} and v_{2i}. The UAV only visits one of these when moving from S to F.

5.3.3 Formulation of the Mixed Integer Linear Program

To solve the Selective Coverage Path Problem on a given graph G, or a subgraph of the graph, I formulate a Mixed Integer Linear Program (MILP). Each pair of vertices representing a sweep line in G is identified sequentially v_{2i-1}, v_{2i} for $i \in \{1, \ldots, \frac{|V|}{2}\}$. The start and finish vertices are v_0 and v_{N+1} where $N = |V|$. The variables of the MILP are defined as follows:

- v_i is set to 1 if vertex i is in the solution, 0 otherwise, $i \in \{0, \ldots, N+1\}$.
- x_{ij} is set to 1 if there is a path from vertex i to vertex j, and 0 otherwise.
- u_i is the service start time at vertex i. The service start time for s_0 is generally 0 but may be adjusted when solving lossy subgraphs.
- R_i is the reward collected from vertex i.
- s_i is the cost to visit vertex i, taken from graph parameter s.
- t_{ij} is the travel cost required for edge x_{ij}, from graph parameter t.
- $T_{\max}$ is the time budget available. For the full problem, it is when the ground vehicle arrives at the finish. For lossy subgraphs, the $\max\{v_{\text{close}}\}$ of G is used.
- O_i, C_i are the open (v_{open}) and close (v_{close}) times for vertex v_i.
- M is a suitably large number. In the implementation, $2T_{\max}$ is used.

The MILP has three objectives, which are solved in a hierarchical manner. Initially, I optimize for the collected reward (5.1). This produces a plan that visits as many of the coverage lines as possible within the time constraint imposed by the UAV's capabilities.

$$\max \sum_{i=1}^{N} R_i v_i \tag{5.1}$$

Subject to: (5.2)

$$u_{N+1} \leq T_{\max} \tag{5.3}$$

$$\sum_{i=1}^{N} x_{0i} = \sum_{i=1}^{N} x_{i(N+1)} = 1 \tag{5.4}$$

$$\sum_{i=0}^{N} x_{ik} = \sum_{i=1}^{N+1} x_{ki} = v_k, \forall k \in \{1, \ldots, N\} \tag{5.5}$$

$$v_{2k-1} + v_{2k} \leq 1 \quad \forall k \in \{1, \ldots, N/2\} \tag{5.6}$$

$$u_i + s_i + t_{ij} - u_j \leq M(1 - x_{ij}), \quad \forall i \in \{0, \ldots, N\}, j \in \{1, \ldots, N+1\} \tag{5.7}$$

$$O_i \leq u_i \leq C_i, \forall k \in \{1, \ldots, N\} \tag{5.8}$$

The amount of time for travel is limited and the plan must return to the finish before resources are exhausted (5.3). The solution must start at the initial vertex and end at the final vertex (5.4). For each vertex, there can be at most one connection in and one out (5.5). Only one of the vertices corresponding to a coverage line can be part of the solution. I enforce a strict ordering of the visited vertices with (5.7) which also calculates the service times and disallows subtours. Finally, visit times are confined to the specified time windows (5.8). The values of x_{ij} and v_i are binary variables, and the service times are positive real:

$$x_{ij}, v_i \in \{0, 1\}, \qquad u_i \in \mathbb{R}^+, \forall i \in \{1, \ldots, N\}.$$

The value of the maximum reward found by the first optimization is added as a constraint to the optimization problem:

$$\sum_{i=1}^{N} R_i v_i \geq R_{\text{opt}} \tag{5.9}$$

The optimization is then solved a second time, minimizing for the overall path length:

$$\min \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} t_{ij} x_{ij} + \sum_{i=1}^{N} s_i v_i. \tag{5.10}$$

If time is available, a third round of optimization is used to minimize the total service time u_{N+1}. This final optimization removes any unnecessary idle time.

5.3.4 Multiple Finish Locations

In some instances, it may be better for the UAV to return early to the ground vehicle instead of proceeding to the end of $P(\cdot)$. To enable early returns the ground vehicle's route P is discretized and, at each discrete point, the ground vehicle's expected arrival time is calculated. This creates a list of potential finish locations and associated closing times that are added to the MILP. All of the finish locations have unconstrained open times.

For F possible finish locations introduce:

- penalty $P_f, \quad P_i < P_j, \forall i, j \in \{1, \ldots, F\}, i < j,$
- variable $z_f \in 0, 1, \forall f \in 1, \ldots, F$, and
- service time $s_f \in \mathbb{R}, \forall f \in 1, \ldots, F$.

The variables z_f replace the previous variable for the finish location v_{N+1}.

The objective of the first phase (maximizing reward) is modified (5.11) to include a penalty, causing the MILP to prioritize an early return without impacting the overall collected reward.

$$\max \sum_{i=1}^{N} R_i v_i - \sum_{f=1}^{F} P_f z_f. \tag{5.11}$$

Additional constraints are added to ensure that only one of the finish locations is valid and included in the solution (5.12) and that they close when the ground vehicle passes (5.13).

$$\sum_{f=1}^{F} z_f = 1, \quad \sum_{i=1}^{N} x_{if} = z_f, \quad \forall f \in \{1, \ldots, F\} \tag{5.12}$$

$$0 \leq s_f \leq T_{\mathrm{gv}_f} \tag{5.13}$$

5.4 Partitioning to Improve Scalability

The exponential solution times of the MILP means that for many problem instances, an optimal solution is not found before the solver exhausts its time budget. However, in this section I describe how the structure of the time windows can be exploited to partition G into subgraphs that can be solved independently.

5.4.1 Exact Partitioning

At any time $t \in [0, T]$, the vertices V in G may be divided into two sets

$$L_t = \{u \in V | u_{\text{close}} \leq t\}, \quad U_t = V \backslash L_t. \tag{5.14}$$

By convention, the start vertex is in L_t and the finish vertex is in U_t. The sets L_t and U_t can be said to partition V,

$$L_t \cap U_t = \emptyset, L_t \cup U_t = V, L_t \neq \emptyset, U_t \neq \emptyset.$$

Lemma 5.4.1 (A Single Edge)**.** Given a time t, let

$$L_t = \{u \in V | u_{\text{close}} \leq t\}, \quad U_t = V \backslash L_t.$$

If for every $u \in L_t$, and $v \in U_t$, we have

$$v_{\text{open}} \geq u_{\text{close}} + s_u + t_{u,v}, \tag{5.15}$$

then in any optimal solution S^* there is exactly one edge between L_t and U_t, and it is directed from L_t to U_t.

Proof. We will prove by contradiction. Assume the route P is cut at t and the inequalities in (5.15) hold. The vertices of V are partitioned into L_t and U_t. Consider any optimal solution S^* – there are two possible cases:

No vertices in U_t appear in S^* If the optimal solution includes no vertices in U_t, then there are no edges from L_t to U_t. After visiting the last vertex $u \in L$, the UAV returns. However, by definition U_t is non-empty, and by (5.15), any vertex $v \in U_t$ can be visited from the latest closing time of any vertex in L_t. Therefore U_t is either empty or the proposed solution is not optimal, both of which are contradictions. Therefore, there must be at least one vertex from U_t in S^*.

There is at least one edge from U_t to L_t For there to be more than one edge in S^* from L_t to U_t, there must be at least one edge from U_t to L_t. However, by (5.15), every vertex in L_t must have closed before any vertex in U_t opens. This case is not possible and there can be no edges from U_t to L_t. Therefore, there can be only one edge between L_t to U_t in S^*, and it is directed from L_t to U_t. □

From this Lemma I obtain a simple procedure for partition the graph into independent subgraphs. This method is detailed in the following.

Divide and Conquer Method:

1. Sort the vertices of G in order of v_{close}.
2. For each unique close time t, partition G into disjoint subgraphs L_t and U_t.
   (a) If the conditions of Lemma 5.4.1 are not satisfied, continue from Step 2 at the next t.
   (b) Otherwise, add a dummy finish vertex to L_t with a zero cost from any vertex in L_t. Add a dummy start vertex to U_t with zero cost to any vertex in U_t.
   (c) Store subgraph L_t as G_i, incrementing i.
   (d) Continue from Step 2 with $G \leftarrow U_L$ until all possible partitions are tested. Store the remaining U_t subgraph as G_n.
3. Independently solve each stored subgraph, $G_1, G_2, \ldots G_n$. Construct the solution S by removing the dummy vertices of G_i, G_{i+1} and connecting the last vertex in the solution of G_i to the first vertex in the solution to $G_{i+1} \forall i = 1, \ldots, n-1$.

Proposition 5.4.2 (Divide and Conquer Solution is Optimal)**.** Any optimal solution of the Divide and Conquer method in Section 5.4.1 is an optimal solution to the Selective Coverage Path Problem 5.2.1.

Proof. If the conditions of Lemma 5.4.1 are satisfied, then the graph has the following properties at t:

- The vertices of $u \in L_t$, $v \in U_t$ are strictly ordered:

$$u_{\text{open}} < u_{\text{close}} < v_{\text{open}} < v_{\text{close}}. \tag{5.16}$$

- On any path from L_t to U_t, the UAV always arrives at $v \in U_t$ before or just as the time window for v opens.

- There are no edges from U_t to L_t.

Irrespective of the path selected in L_t, the UAV must always wait at the first vertex in U_t in the solution of S^*. Further, from (5.16) it can be seen that there are no edges (and therefore no paths) from U_t to L_t, no solution for U_t can affect the solution of L_t. Therefore, the two sections of the path are independent and may be solved separately. □

5.4.2 Limited Loss Partitioning

The conditions of Lemma 5.4.1 are restrictive and frequently do not allow many partitions. This is particularly true when the environment contains a dense set of obstacles close to one another, and the resulting vertices in the graph have overlapping time windows. However, these are also regions in which partitioning would be most useful. Looking at the structure of the open and close times, there are locations in the graph where, if some vertices are "temporarily" removed, the conditions of (5.15) can be met. The temporarily removed vertices may not be included in the final result and so I define their associated rewards as the *potential loss*. Thus, I generate as many partitions as possible while bounding the sum of the potential losses as a function of the total path reward R_T. This observation leads to a limited loss method of applying Lemma 5.4.1.

Limited Loss Divide and Conquer:

1. Sort the vertices of G in order of v_{close}.
2. For unique close times t_j, $j \in \{1, \ldots, n\}$ in G, find the number ℓ of vertices v with $t_{j-1} < v_{\text{close}} \leq t_j$, where $t_0 = 0$. Apply Lemma 5.4.1 at t_j.
   (a) If (5.15) is satisfied, then store $(\ell_j = \ell, c_j = 0)$ and continue from Step 2.
   (b) Otherwise, temporarily remove the next vertex with close time $> t_j$ and mark it. Re-test (5.15). If the partition is not exact, repeat for the next vertex until (5.15) is satisfied or no further vertices remain.
   (c) Evaluate the potential loss c as the sum of the rewards of the marked vertices at t_j. Store the vertex count and loss $(\ell_j = \ell, c_j = c)$ and continue from Step 2.
3. Find the subset of cuts using (ℓ_j, c_j) for $j = 1, 2, \ldots, n$ that minimizes the number of vertices in a subgraph, with a potential loss less than predefined fraction $\rho \in [0, 1)$ of the total reward R_T in G.

4. Solve the subgraphs sequentially and in temporal order.

Remark (Temporal Order)**.** The subgraphs must now be solved in increasing temporal order as limited loss partitioning no longer results in independent subgraphs. The last vertex and visit time from the solution of the prior subgraph act as the starting vertex and time for the following subgraph, preserving the overall graph timing.

Remark (On Removed Vertices)**.** The marked vertices are not actually removed when the graph is partitioned in Step 3. Those vertices are included in the subgraph immediately following the boundary where their potential loss was identified.

5.4.3 A Dynamic Programming for Min-Max Partitioning

To optimize the limited loss partition method, I select the partition of G that is the most advantageous. Since the number of variables in the MILP grows by the square of the number of vertices in the graph, the best choice is the partition that minimizes the size of the largest subgraph while respecting the specified maximum loss. Selecting a subset of cuts from a list is similar to the dynamic programming 0-1 knapsack problem [22].

We start with a graph G and a list of n possible cuts (ℓ_i, c_i), each cut i having a length ℓ_i and a potential loss c_i. For any two cuts $i, j \in \{0, \ldots, n\}, i < j$, I define the length between cuts as the number of vertices from i to j, written

$$\ell_{i,j} = \ell_{i+1} + \cdots + \ell_j.$$

Given a loss budget $B = \rho R_T$, we seek the list of cuts $I = (i_1, i_2, \ldots, i_k)$, $i_1 < i_2 < \ldots < i_k$ where $\sum_{i \in I} c_i \leq B$, that minimizes the maximum length between cuts

$$\max\{\ell_{0,i_1}, \ell_{i_1,i_2}, \ldots, \ell_{i_k,n}\}.$$

We define the subproblem

$$L(j, b) = \{\text{the min-max length of cuts up to cut } j, \text{ with budget } b\},$$

with the constraint that cut j must be included in the solution of the subproblem or the length is defined as $+\infty$. The smallest subproblems which start the recursion are

$$L(0, b) = \begin{cases} 0 & \text{if } 0 \leq b \leq B, \\ +\infty & \text{if } b < 0. \end{cases} \tag{5.17}$$

All other subproblems are solved with the recursion

$$L(j, b) = \min_{i \in \{0,\ldots,j-1\}} \max\{L(i, b - c_i), \ell_{i,j}\},$$

and the final answer given by

$$\min_{j \in \{0,\ldots,n\}} \max\{L(j, b - c_j), \ell_{j,n}\}.$$

From this it is apparent that $L(1, b)$ is

$$\begin{aligned} L(1, b) &= \max\{L(0, b - c_1), \ell_{0,1}\} \\ &= \begin{cases} \ell_1 & \text{if } c_1 \leq b \\ +\infty & \text{if } c_1 > b. \end{cases} \end{aligned}$$

The set of cuts that form the boundaries of the optimal partition of G is extracted from the solution to the subproblems.

Time Complexity One complication in the application of dynamic programming to this problem is that all of the potential losses are real values, not integers. As a result, the size of the table required to look up repeated values is unbounded. In order to provide computational guarantees the potential losses are scaled to an integer value B' and rounded up such that $c'_i = \left\lceil \frac{c_i}{B} B' \right\rceil$. For a graph with $|V|$ vertices, scaling and rounding guarantees a solution with at most $|V| \cdot B'$ values and eliminates floating point errors. The value of B' is selected to be large enough that the results remain accurate, while limiting the cost of computation. After scaling, the final dynamic programming solution has time complexity of $O(|V|^2 \cdot B')$.

5.4.4 Proof of Limited Loss

Based on the discussion of Sections 5.4.2 and 5.4.3 I arrive at the following proposition. In this proposition $R(S)$ denotes the reward collected by a path S and R_T denotes the sum of all rewards in the graph:

$$R_T = \sum_{i-1}^{|V|} R_i.$$

Proposition 5.4.3 (Limited Loss Partitioning)**.** Given a value $\rho \in [0, 1)$, the Limited Loss Divide and Conquer method of Section 5.4.2 produces a solution S with reward

$$R(S) \geq R(S^*) - \rho R_T,$$

where $R(S^*)$ is the reward collected by the optimal solution.

Proof. Given a $\rho \in [0, 1)$, the Limited Loss Divide and Conquer method produces a partition of G into k subgraphs $G_1, \ldots G_k$. Let $G'_1, \ldots, G'_k$ be the subgraphs after removing the marked vertices (as described in 2b) of the algorithm) from $G_1, \ldots G_k$. These marked vertices have a total reward of at most ρR_T. The graphs $G'_1, \ldots, G'_k$ satisfy Proposition 5.4.2 and solution $S' = \{S'_1, \ldots, S'_k\}$ can be found, obtained by concatenating the optimal solutions on each subgraph. Let $R(S'_1), \ldots, R(S'_k)$ be the rewards collected by the S' in each of these subgraphs. The optimal solution S^* on G must contain some vertices in each G'_i, call them S^*_i, such that

$$R(S'_i) \geq R(S^*_i).$$

That is, the optimal solution S^* may contain exactly the same vertices as S'_i or fewer vertices. Thus,

$$R(S') = \sum_{i=1}^{k} R(S'_i) \geq \sum_{i=1}^{k} R(S^*_i). \tag{5.18}$$

The total reward for the optimal solution is

$$R(S^*) \leq \sum_{i=1}^{k} R(S^*_i) + \rho R_T \tag{5.19}$$

Therefore, after combining (5.18) and (5.19), the reward from partitioning G' is

$$R(S') \geq R(S^*) - \rho R_T.$$

Since each solution S'_i for the subgraph G'_i is a feasible solution in G_i (the graph G_i with the marked vertices included), we immediately have that the optimal solution on $G_1, \ldots, G_k$ is at least that of the optimal solution obtained from $G'_1, \ldots, G'_k$. □

Remark (Conservativeness of Bound)**.** As ρ becomes larger, the bound in Proposition 5.4.3 becomes very loose. To obtain the bound, I assume that S^* collects the entire ρR_T reward of the removed vertices. In practice this will not be possible. In addition, when implementing the Limited Loss Partition, because vertices are not actually removed, the solution S can potentially collect rewards in addition to $R(S)$.

5.5 Simulations and Results

All simulations[1] are implemented in C++ using the following libraries and tools: Visibility polygons [45], Computational geometry functions from Boost [8], Minimum polygon heights [58], Concave polygons [1], QuickHull [52], Polygon decomposition [56], and Dubins curves [73]. The Dubins curve library is the steering function for the UAV path, adding a radius of $1m$ to all turns. I solved the MILP using Gurobi [30], allowing up to $t_{\text{total}} = 1000s$ on a Ubuntu 18.04 desktop PC with an Intel(R) i7-7700K CPU and 32GB RAM.

As shown in Figure 5.4, in each simulation the environment $\mathscr{E}$ is generated with 20 to 100 randomly placed obstacles. The start and finish positions are a constant $3000m$ apart. The parameters for the ground and aerial vehicles are V_{gv}=$1m/s$, V_{uav}=$5m/s$, w=$200m$, d=$300m$, and f=$25m$. The ground vehicle path is found using the OMPL [66] implementation of RRT*, resulting in some slight variation between simulations. Figure 5.5 is taken from an active simulation and shows the planning results. A video (see footnote 1) provides a demonstration of the simulation environment.

In the results, I present data from four different solution methods: Optimal, ExactDnC, DnC30 and DnC50. Optimal uses the unmodified graph G. ExactDnC allows partitions according to Lemma 5.4.1. DnC30 and DnC50 allow lossy partitioning limited to 30 per cent and 50 per cent of the total coverage reward R_T, respectively. The results are grouped and plotted as a function of the number of vertices in G, which are directly proportional to the number of sweep lines in $\mathscr{E}$. As the number of obstacles in the coverage corridor increases, so too does the number of sweep lines, indicating a greater difficulty in providing an optimal solution given a fixed time budget for the MILP solver.

Subdivision of Available Solution Time The MILP solver is given a fixed time budget t_{total}, to be divided between the subgraphs. Since the number of variables in the MILP grows as the square of $|V|$ (see Section 5.3.3), I allot time proportional to the square of the number of subgraph vertices. Unused solution time is passed to the next subgraph.

5.5.1 Comparison with Complete Coverage

Figure 5.6 illustrates the advantage of covering only the hidden areas when compared to covering the entire corridor using the Conformal lawn mower method [28]. In this figure,

[1] A video demonstration of the simulation environment is available: https://ece.uwaterloo.ca/~sl2smith/papers/2020CASE-Coverage.m4v

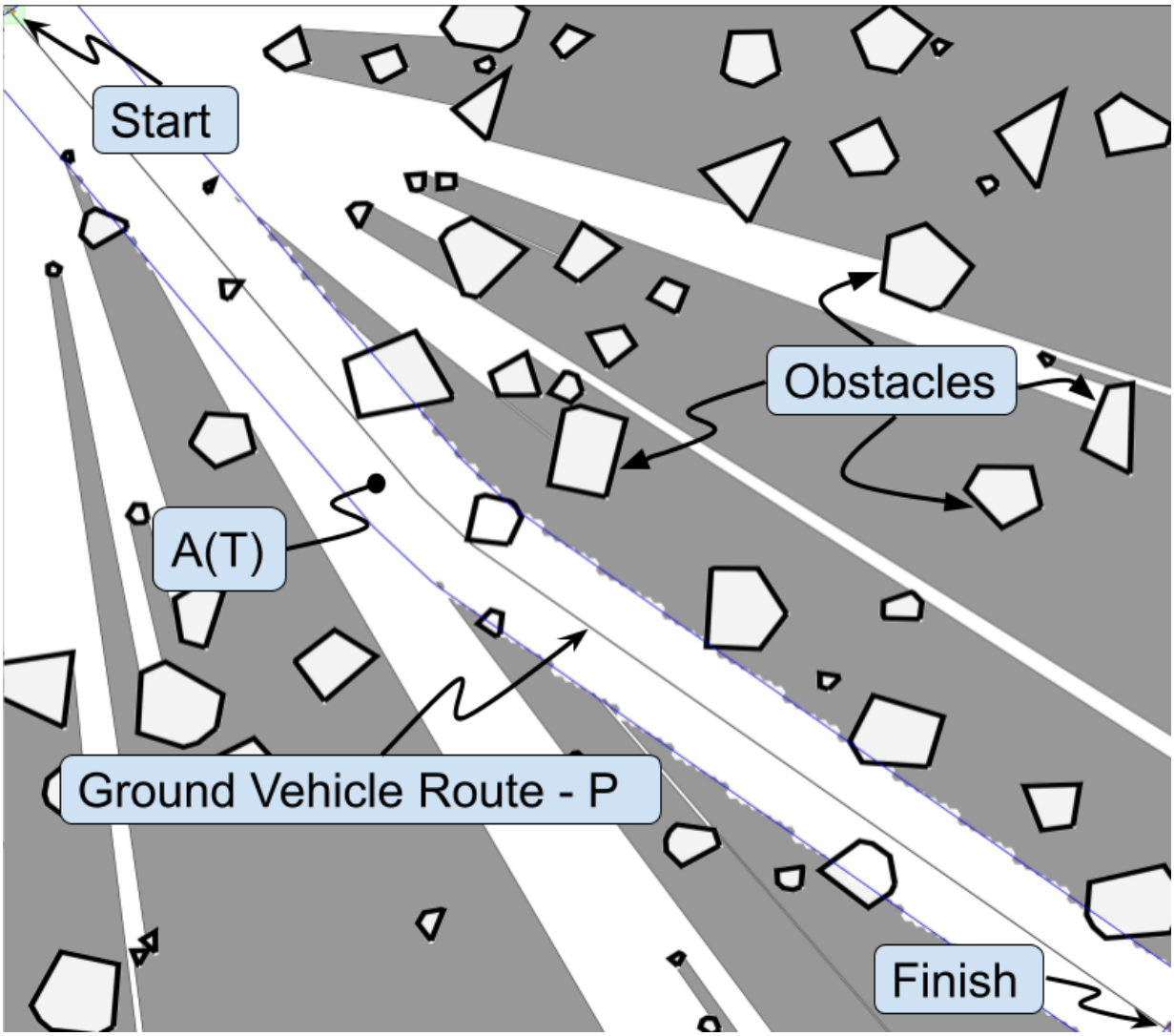


Figure 5.4: The simulation environment – the ground vehicle route runs from top left to bottom right.

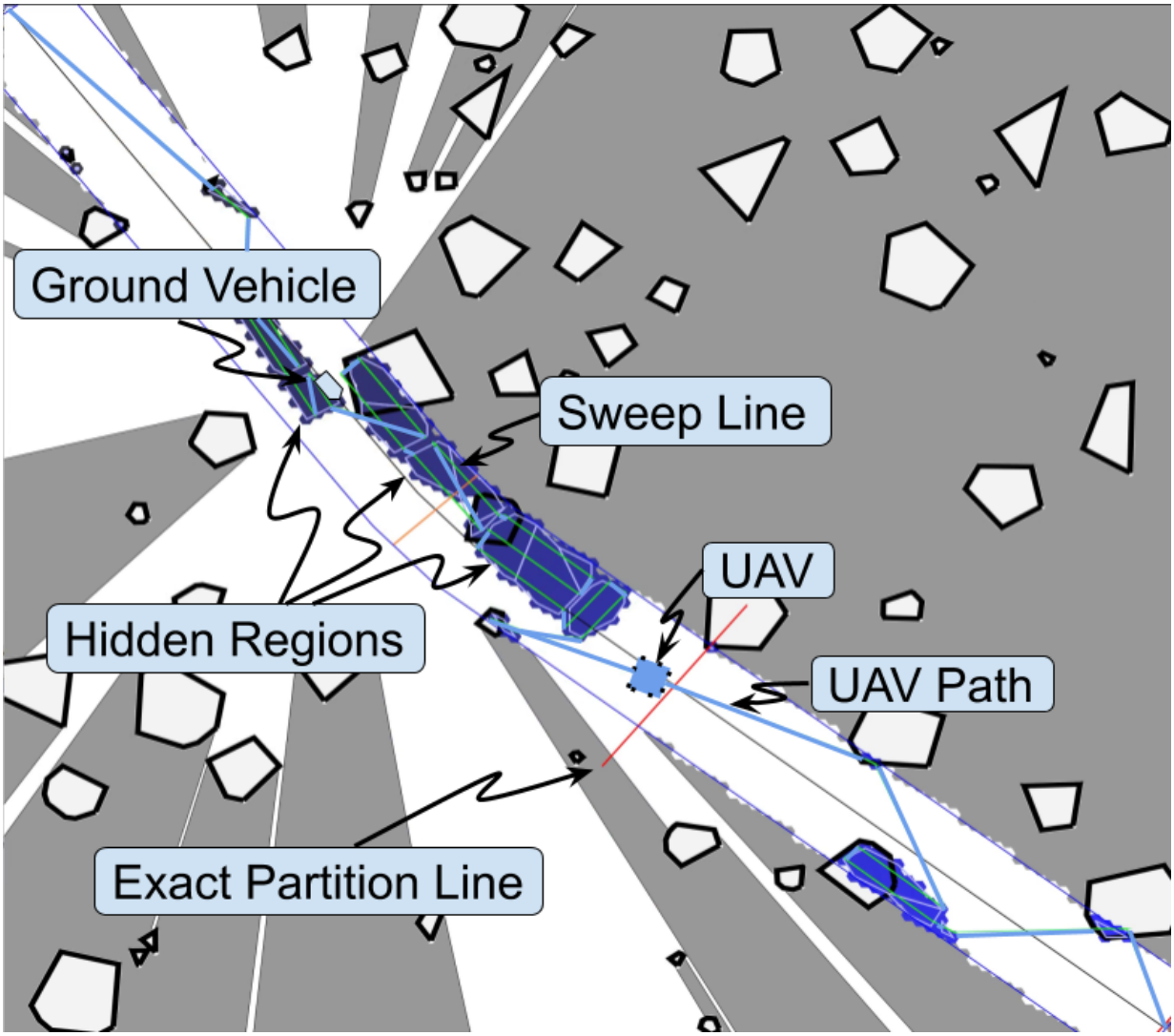


Figure 5.5: An active simulation: the hidden regions and the UAV's flight path can be seen.

the vertical axis is the total distance traveled by the UAV to perform coverage. The results show that when the number of vertices is low, covering only the hidden areas results in paths that are two to five times shorter. As the number of vertices increases, the benefit of covering only the hidden areas diminishes.

5.5.2 Exact and Limited Loss Partitioning

Figure 5.7 compares the time required to find a solution to the MILP using the four different solution methods as a function of the number of vertices in the full graph G. After one hundred simulations, the results are collected and plotted by vertex counts. The plots show the mean value and 95% confidence interval for a median of six trials for each vertex value. Initially, all methods terminate prior to their time budget. However, as the number of vertices increases, the time required to solve either of the Optimal or ExactDnC cases quickly reaches the maximum limit. The lossy methods, DnC30 and DnC50, solve within the allotted time for much larger problems. The effect on solution times is shown in finer detail in Table 5.1, comparing the maximum number of variables in the partitioned subgraphs and the solution times against the size of the initial graph. Figure 5.8 plots the number of covered cells in simulation, when following the UAV path generated by the planner. The simulations are consistently within 20% of the MILP reward shown in Figure 5.9.

Figure 5.10 illustrates the quadratic growth in the number of program variables as the number of vertices in the graph increases. Divide and conquer is clearly a viable strategy as both of the lossy implementations are smaller by an order of magnitude. This graph also provides evidence that shows that ExactDnC partitions are only possible when the graph is small, limiting the usefulness of an ExactDnC only strategy.

5.5.3 Path Lengths

In this experiment I ran 1000 trials, increasing V_{uav} to 20 m/s and limiting the MILP to $t_{\text{total}} = 120s$. The results can be seen in Figure 5.11. Under these conditions, as the scale of the problem increases, the Optimal and ExactDnC solutions are clearly not capable of minimizing the length of the path in the available time.

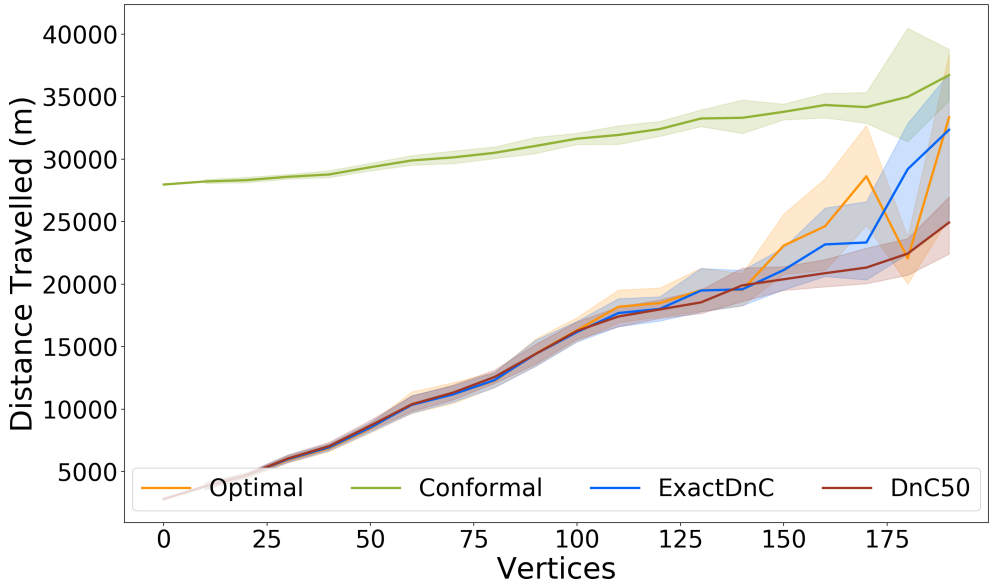


Figure 5.6: Comparing the relative distance flown by the UAV for complete coverage (Conformal) and covering only the hidden regions.

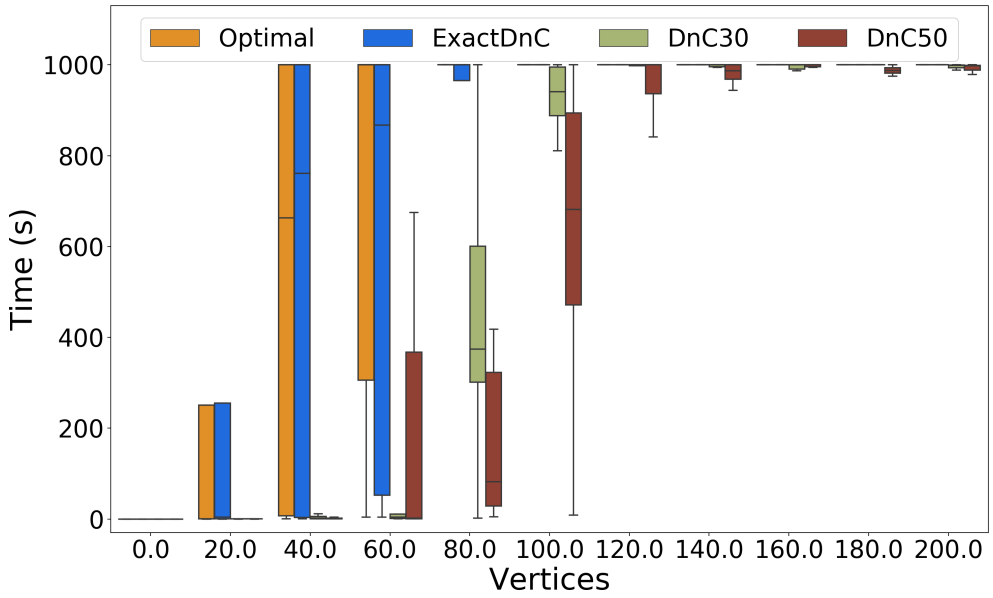


Figure 5.7: MILP solution time. The solver is limited to 1000s.

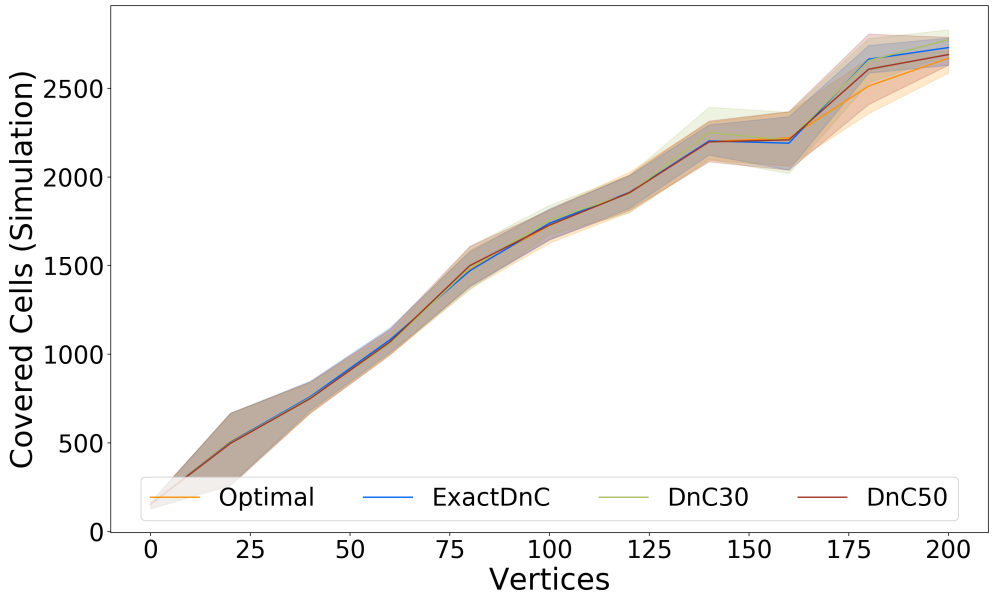


Figure 5.8: Simulated Coverage - the count of cells covered through simulation, validating MILP reward.

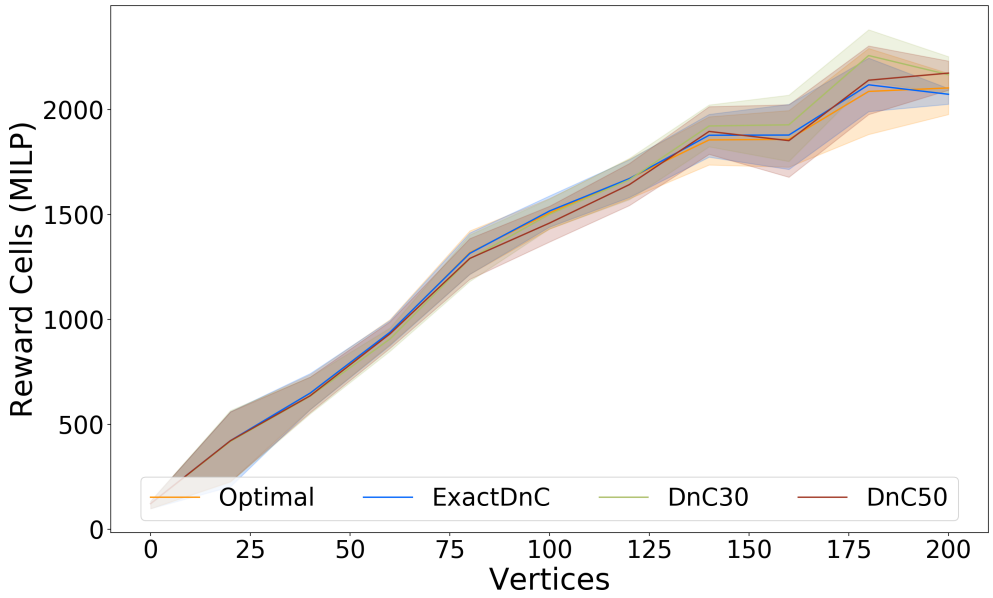


Figure 5.9: The reward from MILP solver as measured by the number of cells covered.

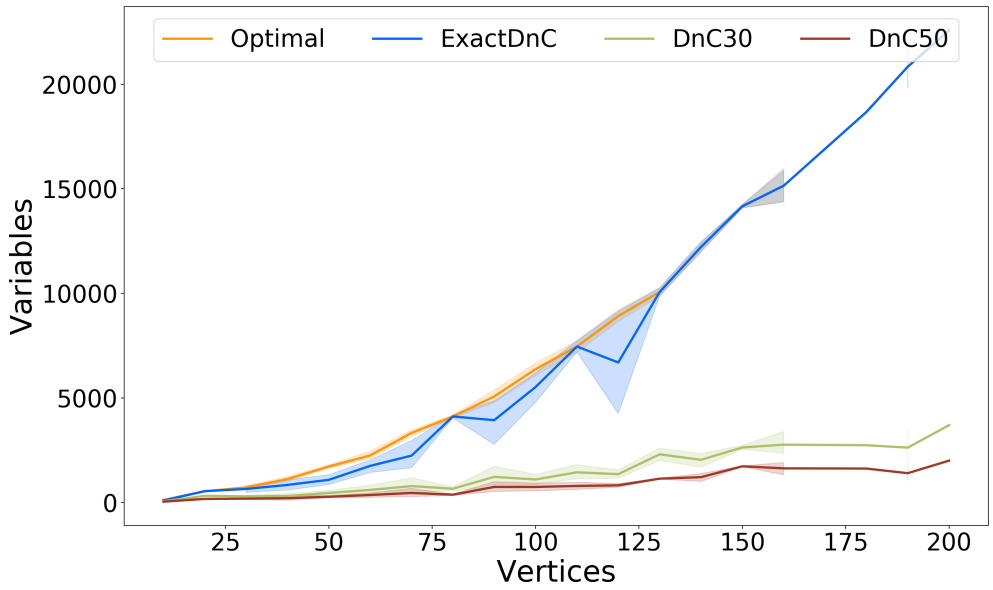


Figure 5.10: Maximum Number of Variables in subgraphs.

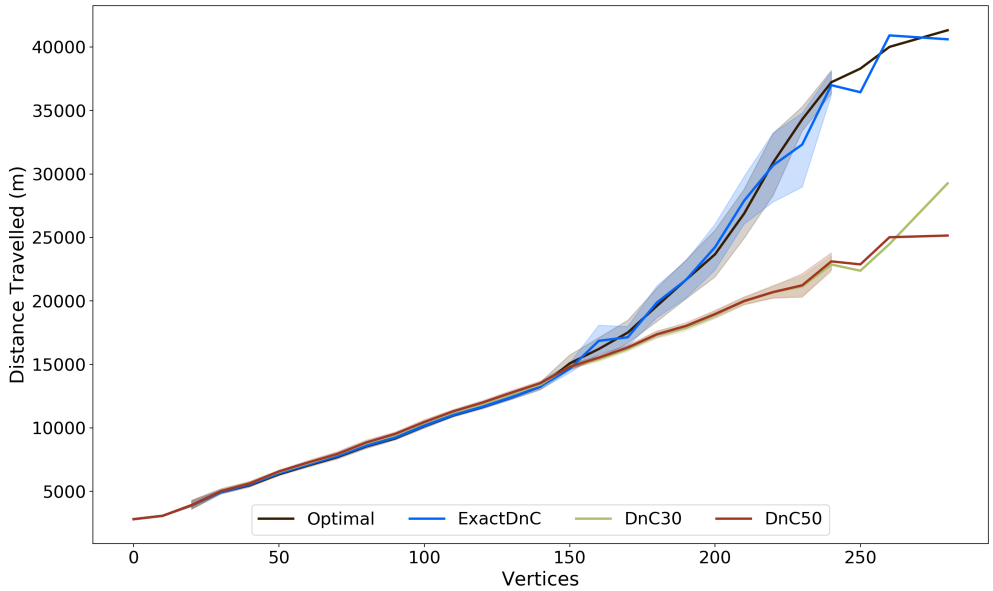


Figure 5.11: UAV path distance with velocity constraints lifted. For the largest problems, the Optimal and ExactDnC methods return paths that are as much as double the paths of the lossy methods DnC30 and DnC50.

Table 5.1: A comparison of graph vertex counts vs. maximum number of variables in the partitioned subgraphs.

Vertices	Optimal		ExactDnC		DnC30		DnC50	
	Vars	Time	Vars	Time	Vars	Time	Vars	Time
0	72	0	52	0	28	0	28	0
20	370	167	355	168	187	0	118	0
40	1142	392	878	405	358	83	227	3
60	2116	745	1463	656	579	128	342	163
80	3887	1000	2964	936	811	397	474	193
100	6300	1000	5532	985	1121	915	712	653
120	8713	1000	7860	1000	1682	986	936	960
140	11977	1000	11977	1000	2108	992	1207	983
160	14819	1000	14819	1000	2721	994	1666	996
180	19254	1000	19254	1000	2256	1000	1353	988
200	21749	1000	21749	1000	3266	996	1709	992

5.6 Conclusions and Future Work

I formulated a MILP that solves the Selective Coverage Path Problem and proved that the coverage problem can be exactly partitioned while retaining the optimal solution under certain conditions. I also developed a limited loss partitioning method which found an optimal partition within a bounded loss in coverage reward. The lossy method allowed the coverage problem to be subdivided in cases where the exact partition was not possible. Lossy partitioning may also provide better solutions for scenarios where the solver fails to complete before allotted time expires, although I have not fully explored this possibility. I validated both the exact and lossy methods through simulation.

A demonstration of looseness of the lower bound established by Proposal 5.4.3 can be seen in the simulation results. The lossy solutions achieve a solution quality that is very close to the optimal, even when the bound allows up to 50% of coverage to be lost. For smaller solutions, I speculate that there are sufficient resources to cover all areas. In larger graphs, where the orienteering aspect of the problem becomes significant, the solver time elapses before an optimal path is found. Partitioning the graph seems to localize the loss and allow the solver to get closer to optimal solutions on the smaller subgraphs. Studying this loss localization and other graph partitioning strategies remains the subject of further study.

Chapter 6

Conclusions and Future Directions

In this thesis, I have investigated three variations of coverage path planning for a moving ground vehicle. Initially, I looked at finding the minimum performance requirements for the UAV to provide current coverage of the path immediately ahead of a ground vehicle travelling through the environment. The plan I developed, a variation of the classic lawn mower plan, ensures complete path coverage without prior knowledge of the ground vehicle's entire route. I demonstrated that the Conformal lawn mower path can be no longer than twice the length of the optimal plan, and therefore require no more than twice the UAV velocity, for any path with maximum curvature constrained to less than half of its width.

In environments where, due to security or other reasons, the path information is unavailable, I considered the predictive case and established necessary and sufficient velocities for successful coverage. The lack of information about the ground vehicle's future path requires the UAV to cover territory that is ultimately extraneous to the ground vehicle's needs, and dramatically increases the UAV velocity requirements.

I concluded the investigation by considering a solution that decreases the total area to be covered by the UAV. The selective method predicts hidden areas in the environment and plans a cost efficient route to cover those areas only. I formulated a MILP that solves for the maximum coverage with the shortest path. However, without intervention, as the size of underlying graph grows, the solver times out, even on relatively small problems. I proved that the coverage problem can be optimally subdivided and solved under certain conditions. A relaxed version of the optimal cut which found the optimal subset of cuts within an allowable loss was also developed. The relaxed approach allowed the coverage problem to be subdivided further, trading off a bounded decreased coverage for shorter solution times.

I further validated both the optimal and relaxed approaches through simulation.

Over the course of developing this thesis, I came across many additional opportunities for exploration. These are the paths not taken.

6.1 Flight Planning

6.1.1 Robust Response to Path Errors

The coverage planning developed in this thesis is brittle, and prone to coverage errors – missed cells that happen to be located on the edge of the optical footprint as the UAV passes. In many cases, these are errors due to rounding and floating point representation. Looking forward, these errors can only increase when operating in a more realistic scenario, where the actual path of the UAV may deviate significantly from the requested plan due to environmental effects such as wind. There are two possible mechanisms for handling this error: offline planning ahead of time to minimize the impact, and online planning in response to coverage failures, both of which I briefly sketch here.

Offline Planning to Prevent Errors

To account for errors, it is necessary to establish a maximum likely path deviation (either through experiment or calculation) and based on the path variation, estimate the maximum likely error, d_{err}, for any given path. Then, using the error estimate, the effective optical footprint can be established as

$$f_{\mathit{eff}} = f(1 - \frac{d_{err}}{f}).$$

With this reduced footprint, and following the derivation of Chapter 3, and in particular (3.8), then the expected path length required increases by $\frac{1}{f_{err}}$. In cases where additional errors need to be accounted for, the effective footprint may be further reduced.

Online Recovery After Error

Recovery of missed sections (due to unexpected wind, etc.) while simultaneously providing coverage is likely to be extremely difficult, primarily due to time requirements. On one level it seems a simple enough problem: evaluate the coverage results live, and if there is

a missed section, insert a new trajectory into the UAV's plan that allows the area to be revisited. However, the UAV is already running just ahead of the deadline. Unless the UAV is capable of significantly increased speeds for recovery intervals, it won't be possible to maintain both the leading coverage edge and go back to fill in missed sections before they expire. Online, this problem may be better solved by coordinating multiple UAVs.

6.1.2 Constant vs. Direction of Flight Orientation

The type of UAV employed can affect the options available for flight planning. While a fixed wing UAV can only be oriented in the direction of flight, a rotor-craft UAV is holonomic and flies equally well in any direction and any orientation (maintaining a primarily upright/constant Z position).

6.2 Collaboration Improvements

6.2.1 Multi-UAV Coordination

The problem of providing coverage to a moving vehicle appears to be highly episodic in nature, particularly considering the differences in travel distance without refuelling between a ground vehicle (just about any ground vehicle) and a UAV with its limited flight time. Investigating how best to subdivide the coverage area and develop policies for multiple UAVs working in collaboration is an interesting problem. From the data, it is apparent that at a reduced velocity, the UAV is initially successful until the deadline or ground vehicle catches up. Working in teams, the UAVs could coordinate their efforts to map the path and create a complete solution.

6.2.2 Iterative Collaboration

Recognizing that there is a limit to the peak velocity and acceleration that may be employed by a UAV leads to another possible avenue of research: developing a solution that allows the UAV and the ground vehicle to negotiate the route together. The resulting path should ensure full coverage for the ground vehicle, while remaining within the operational capabilities of the UAV.

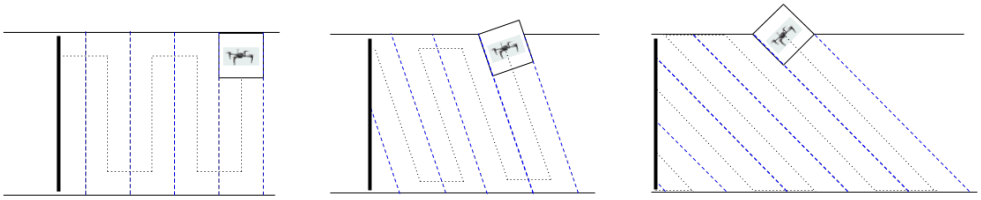
6.2.3 Continuous MILP Optimization

If we view the coverage problem a relative to the ground vehicle instead of with respect to the ground vehicle moving in the environment, then we can consider each coverage task arriving at a certain time and moving towards the ground vehicle until it expires. The role of the planner then becomes to incorporate each new coverage requirement as it appears into the coverage graph, resolving the MILP after each insertion. Without further consideration, my research shows that this could quickly run into trouble as the MILP grows and solution times become unpredictable. One method to maintain a predictable solution rate could be to opportunistically cut the graph when appropriate conditions arise. If Lemma 5.4.1 is satisfied, then the graph is cut and the early portion solved without consequence. Otherwise, the planner maintains an estimated solution time for the current graph and a list of possible cut locations and associated costs. If the graph grows to the point where the solution time becomes unpredictable, the next best cut with relaxation is made. This method of implementing a receding horizon MILP may allow the solver to maintain bounded $\epsilon-$optimal solutions within predictable times allowing (or possibly improving) online planning.

6.3 Analysis Improvements

6.3.1 Improving the Sufficient Bound

It is possible that an improved sufficient bound might be found by allowing the angle of the sweep line to deviate from parallel to the deadline (perpendicular to the vehicle path) as illustrated in Figure 6.1. The risk this entails is a longer possible period required to completely sweep ahead of the deadline on any single sweep. As the angle increases, the ends of the sweep line must move towards the boundary edge to ensure complete coverage. Extending the sweep line ensures that there are no holes induced on the perimeter as the sensor footprint rotates. The increasing angle also increases the overall length of the sweep line because the UAV is now travelling along the hypotenuse of the triangle. While these two effects combine to increase the length of the sweep, and therefore increasing the necessary velocity of the UAV, the resulting increase may be less than that required by the Conformal plan for the same path. There is another reason to be wary of allowing the angle to deviate from parallel: a pathological case where a long slow curve builds up significant deviation before making a hard turn in the other direction. The result is a possible sweep line of indeterminate length. A mechanism is required to enforce a bound on the length



(a) $\theta = 0$ (b) $SmallAngle$ (c) $\theta = \frac{\pi}{4}$

Figure 6.1: The sweep lines expand as the angle increases.

of the generated sweep line and correct the coverage plan as required. Another recovery mechanism is required to handle cases where the induced angle causes missed areas of the path as the sweep lines are not guaranteed to be integer divisible into the curvature of the corridor.

An approach that provides those bounds might be as follows: consider bounding the path with two parallel lines. The sweep line of the UAV is set as perpendicular to the bounding parallel lines. Assuming the bounding parallel lines are separated by less than $D < 2w$, then a bound for that path can be established as $\frac{D}{f} = 2\frac{w}{f}$, the same as the worst case for the Conformal method. If the separation distance is greater than $2\frac{w}{f}$, then the Conformal plan can be used to guarantee a sufficient path instead. Alternatively, the path can be sectioned and each of the sections planned individually. In this manner, sufficient velocity for the path is bounded by the width of the widest subsection. This approach may be complicated by transitions between the subsections which are not guaranteed to be smooth/cost-free.

6.3.2 Path Optimization

There may be further opportunities to minimize the overall flight distance by eliminating unnecessary coverage overlap from the calculated UAV plan. As illustrated in Figures 3.5a, 3.5b, reducing the overlap between scan lines can result in a shorter path. In the plans developed for the selective coverage, there was also a fair amount of overlap as the paths did not necessarily consider the impact of prior passes. These cases seem to indicate that there may be significant room for improvement by implementing a sub-modular reward function in the MILP [60].

6.3.3 Proof of Time Complexity

The complexity of finding an optimal complete coverage plan for a moving ground vehicle is unknown. Based on [3], it seems likely that any algorithm that performs this task will be NP-hard, but this assertion remains unproven.

6.3.4 Measure of Path Complexity

One of the difficulties obscured by assuming a constant velocity is that, in emulation and in the physical implementation, the UAV must slow down significantly, apply additional acceleration, or add additional distance to the trajectory (as we attempt to emulate with the Dubins trajectories) to change direction. Most likely some combination of all three are applicable. Some analysis should be applied to the complexity of the path – what percentage of the time is the UAV applying additional acceleration (either to decrease or increase velocity) vs. flying at a steady state, and how does that affect the flight time for a given path length.

6.4 Simulation and Implementation

6.4.1 Advanced Simulation and Validation

The simulations performed in this research focused on a single integrator model – one that considers only the instantaneous velocity of the UAV in the calculation of the coverage paths. This simplification does not fully encapsulate the costs required to execute a change of direction with a flying vehicle. In order to change direction, the UAV must expend energy to first slow down before accelerating again in the direction of the new line of travel.

In Chapter 3, I compared the Conformal coverage plan with a handcrafted alternative and argue that the handcrafted path is significantly shorter. However, this scenario doesn't consider the cost of turns, nor does it look at the time required for the UAV to accelerate to the specified velocity. In this particular example, the Conformal path actually has slightly fewer turns, and longer straight paths. The additional length the Conformal path induces may actually not be an issue if it means the UAV can sustain its desired velocity for a longer period of time. In Chapter 5, Dubins curves were added to the route to provide the ability to account for the increased cost of turns. However, the adjustment is somewhat coarse with no direct relation between the cost making the turn and the velocity of the

UAV. Further, as the velocity of the UAV increases, the required combination of increased distance, time, and energy to change directions must also increase – a relationship that is not considered as part of this analysis. Future work that considers the trade-off between the UAV coverage velocities and the energy cost of changing direction in the course of flight is required to better characterize the ability of the UAV to provide complete coverage.

6.4.2 Physical Implementation

Finally, there was insufficient time to employ these techniques in a real-world scenario.

References

[1] Stanislaw Adaszewski. Concaveman: a very fast 2D concave hull, 2019. https://github.com/sadaszewski/concaveman-cpp.

[2] Mazda Ahmadi and Peter Stone. Continuous area sweeping: A task definition and initial approach. In *Proceedings, 12th International Conference on Advanced Robotics (ICAR)*, pages 316–323. IEEE, 2005.

[3] Esther M Arkin, Sándor P Fekete, and Joseph SB Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17(1-2):25–50, 2000.

[4] S. Batzdorfer, M. Bobbe, M. Becker, H. Harms, and U. Bestmann. Multisensor Equiped UAV/UGV for Automated Exploration. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W6:33–40, August 2017.

[5] Graeme Best, Wolfram Martens, and Robert Fitch. Path Planning With Spatiotemporal Optimal Stopping for Stochastic Mission Monitoring. *IEEE Transactions on Robotics*, 33(3), June 2017.

[6] Robert E Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, pages 107–121, 2012.

[7] Stanislav Bochkarev and Stephen L Smith. On minimizing turns in robot coverage path planning. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1237–1242. IEEE, 2016.

[8] Boost. The Boost C++ Geometry Library, 2009-2017. https://www.boost.org/doc/libs/1_65_0/libs/geometry/doc/html/index.html.

[9] Shaunak D. Bopardikar, Stephen L. Smith, and Francesco Bullo. On Dynamic Vehicle Routing With Time Constraints. *IEEE Transactions on Robotics*, 30(6), December 2014.

[10] Yasser Bouzid, Yasmina Bestaoui, and Houria Siguerdidjane. Quadrotor-UAV optimal coverage path planning in cluttered environment with a limited onboard energy. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 979–984. IEEE, 2017.

[11] Jose Caceres-Cruz, Pol Arias, Daniel Guimarans, Daniel Riera, and Angel A Juan. Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2):1–28, 2014.

[12] L Cantelli, M Lo Presti, M Mangiameli, CD Melita, and G Muscato. Autonomous cooperation between UAV and UGV to improve navigation and environmental monitoring in rough environments. In *Proceedings 10th International Symposium on Humanitarian Demining*, pages 109–112, 2013.

[13] L Chaimowicz, A Cowley, D Gomez-Ibanez, B Grocholsky, MA Hsieh, H Hsu, JF Keller, V Kumar, R Swaminathan, and CJ Taylor. Deploying air-ground multi-robot teams in urban environments. In *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 223–234. Springer, 2005.

[14] Howie Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9(3):247–253, 2000.

[15] Howie Choset. Coverage for robotics A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, October 2001.

[16] Gordon Christie, Adam Shoemaker, Kevin Kochersberger, Pratap Tokekar, Lance McLean, and Alexander Leonessa. Radiation search operations using scene understanding with autonomous UAV and UGV. *Journal of Field Robotics*, 34(8):1450–1468, 2017.

[17] EJ Cockayne and GWC Hall. Plane motion of a particle subject to curvature constraints. *SIAM Journal on Control*, 13(1):197–220, 1975.

[18] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[19] Carmelo Di Franco and Giorgio Buttazzo. Coverage path planning for UAVs photogrammetry with energy and resolution constraints. *Journal of Intelligent & Robotic Systems*, 83(3-4):445–462, 2016.

[20] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[21] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3):497–516, 1957.

[22] Absalom E Ezugwu, Verosha Pillay, Divyan Hirasen, Kershen Sivanarain, and Melvin Govender. A comparative study of meta-heuristic optimization algorithms for 0–1 knapsack problem: Some initial results. *IEEE Access*, 7:43979–44001, 2019.

[23] Roman Fedorenko, Aidar Gabdullin, and Anna Fedorenko. Global UGV Path Planning on Point Cloud Maps Created by UAV. In *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pages 253–258, September 2018.

[24] Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.

[25] Carmelo Di Franco and Giorgio Buttazzo. Energy-Aware Coverage Path Planning of UAVs. In *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, pages 111–117, April 2015.

[26] Roland Geraerts and Mark H Overmars. A comparative study of probabilistic roadmap planners. In *Algorithmic Foundations of Robotics V*, pages 43–57. Springer, 2004.

[27] Subir Kumar Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, Cambridge, 2007.

[28] Barry Gilhuly and Stephen L. Smith. Robotic Coverage for Continuous Mapping Ahead of a Moving Vehicle. *arXiv:1909.03304 [cs]*, September 2019. arXiv: 1909.03304.

[29] Ben Grocholsky, James Keller, Vijay Kumar, and George Pappas. Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 13(3):16–25, 2006.

[30] LLC Gurobi Optimization. Gurobi Optimizer Reference Manual, 2019.

[31] Takaya Hakukawa, Kenji Uchiyama, and Kai Masuda. Cooperative System with UAV and UGV for Disaster Area Exploration. In *2018 7th International Conference on Systems and Control (ICSC)*, pages 117–122, October 2018. ISSN: 2379-0067, 2379-0059.

[32] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[33] Samira Hayat, Evşen Yanmaz, Timothy X Brown, and Christian Bettstetter. Multi-objective UAV path planning for search and rescue. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5569–5574. IEEE, 2017.

[34] Paul J. Heffernan and Joseph S. B. Mitchell. An Optimal Algorithm for Computing Visibility in the Plane. *SIAM Journal on Computing*, 24(1):184–201, February 1995.

[35] Benjamin Hepp, Matthias Nießner, and Otmar Hilliges. Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *ACM Transactions on Graphics (TOG)*, 38(1):4, 2018.

[36] Shannon Hood, Kelly Benson, Patrick Hamod, Daniel Madison, Jason M O'Kane, and Ioannis Rekleitis. Bird's eye view: Cooperative exploration by UGV and UAV. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 247–255. IEEE, 2017.

[37] W.H. Huang. Optimal line-sweep-based decompositions for coverage algorithms. In *Proceedings 2001 IEEE International Conference on Robotics and Automation (ICRA). (Cat. No.01CH37164)*, volume 1, pages 27–32 vol.1, May 2001. ISSN: 1050-4729.

[38] Jason A Janet, Ren C Luo, and Michael G Kay. The essential visibility graph: An approach to global motion planning for autonomous mobile robots. In *Proceedings of 1995 IEEE international conference on robotics and automation*, volume 2, pages 1958–1963. IEEE, 1995.

[39] Nitin Kamra and Nora Ayanian. A mixed integer programming model for timed deliveries in multirobot systems. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 612–617. IEEE, 2015.

[40] Marisa G. Kantor and Moshe B. Rosenwein. The Orienteering Problem with Time Windows. *Journal of the Operational Research Society*, 43(6):629–635, June 1992.

[41] Orhan Karasakal. Minisum and maximin aerial surveillance over disjoint rectangles. *TOP*, 24(3):705–724, October 2016.

[42] Wajahat Kazmi, Morten Bisgaard, Francisco Jose Garcia-Ruiz, Karl D Hansen, and Anders la Cour-Harbo. Adaptive surveying and early treatment of crops with a team of autonomous vehicles. In *European Conference on Mobile Robots*, pages 253–258, 2011.

[43] Jeongeun Kim, Seungwon Kim, Chanyoung Ju, and Hyoung Il Son. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. *IEEE Access*, 7:105100–105115, 2019.

[44] Jin Hyo Kim, Ji-Wook Kwon, and Jiwon Seo. Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV. *Electronics Letters*, 50(20):1431–1432, 2014.

[45] Byron Knol. A Library to Construct a Visibility Polygon from Line Segments. https://github.com/byronknoll/visibility-polygon-js.

[46] Tomas Lazna, Petr Gabrlik, Tomas Jilek, and Ludek Zalud. Cooperation between an unmanned aerial vehicle and an unmanned ground vehicle in highly accurate localization of gamma radiation hotspots. *International Journal of Advanced Robotic Systems*, 15(1), 2018.

[47] Jianqiang Li, Genqiang Deng, Chengwen Luo, Qiuzhen Lin, Qiao Yan, and Zhong Ming. A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems. *IEEE Transactions on Vehicular Technology*, 65(12):9585–9596, 2016.

[48] Yan Li, Hai Chen, Meng Joo Er, and Xinmin Wang. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, 21(5):876–885, August 2011.

[49] Parikshit Maini, Kevin Yu, PB Sujit, and Pratap Tokekar. Persistent monitoring with refueling on a terrain using a team of aerial and ground robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8493–8498. IEEE, 2018.

[50] Neil Mathew, Stephen L Smith, and Steven L Waslander. Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Transactions on Robotics*, 31(1):128–142, 2015.

[51] Zehui Meng, Hailong Qin, Ziyue Chen, Xudong Chen, Hao Sun, Feng Lin, and Marcelo H Ang Jr. A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction. *IEEE Robotics and Automation Letters*, 2(3):1680–1687, 2017.

[52] Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, New York, NY, USA, 2nd edition, 1998.

[53] Mark Owen, Randal W Beard, and Timothy W McLain. Implementing Dubins airplane paths on fixed-wing UAVs. *Handbook of Unmanned Aerial Vehicles*, pages 1677–1701, 2015.

[54] Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981.

[55] Robert Pěnička, Jan Faigl, and Martin Saska. Physical Orienteering Problem for Unmanned Aerial Vehicle Data Collection Planning in Environments with Obstacles. *IEEE Robotics and Automation Letters*, 4(3):3005–3012, 2019.

[56] Mark Penner. Decomposition of Polygons using Bayazit's Algorithm, 2019. https://github.com/idlebear/polyDecomp.

[57] John Peterson, Haseeb Chaudhry, Karim Abdelatty, John Bird, and Kevin Kochersberger. Online Aerial Terrain Mapping for Ground Robot Navigation. *Sensors*, 18(2):630, February 2018.

[58] Franco P Preparata and Michael I Shamos. *Computational Geometry: an introduction*. Springer-Verlag, 1985.

[59] Shiwen Ren, Yang Chen, Ling Xiong, Zhihuan Chen, and Mengqing Chen. Path planning for the marsupial double-UAVs system in air-ground collaborative application. In *Chinese Control Conference (CCC)*, pages 5420–5425. IEEE, 2018.

[60] Mike Roberts, Debadeepta Dey, Anh Truong, Sudipta Sinha, Shital Shah, Ashish Kapoor, Pat Hanrahan, and Neel Joshi. Submodular trajectory optimization for aerial 3d scanning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5324–5333, 2017.

[61] Martin Saska, Vojtěch Vonásek, Tomáš Krajník, and Libor Přeučil. Coordination and navigation of heterogeneous MAV–UGV formations localized by a hawk-eye-like approach under a model predictive control scheme. *The International Journal of Robotics Research*, 33(10):1393–1412, 2014.

[62] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[63] Michael Ian Shamos. Geometric complexity. In *Proceedings of the seventh annual ACM symposium on Theory of computing*, pages 224–233. ACM, 1975.

[64] Stephen L Smith, Mac Schwager, and Daniela Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Transactions on Robotics*, 28(2):410–426, 2012.

[65] Ethan Stump and Nathan Michael. Multi-robot persistent surveillance planning as a vehicle routing problem. In *2011 IEEE International Conference on Automation Science and Engineering*, pages 569–575. IEEE, 2011.

[66] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. http://ompl.kavrakilab.org.

[67] Thomas C Thayer, Stavros Vougioukas, Ken Goldberg, and Stefano Carpin. Multi-Robot Routing Algorithms for Robots Operating in Vineyards. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 14–21. IEEE, 2018.

[68] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. Sensor planning for a symbiotic UAV and UGV system for precision agriculture. *IEEE Transactions on Robotics*, 32(6):1498–1511, 2016.

[69] Marina Torres, David A. Pelta, Jos L. Verdegay, and Juan C. Torres. Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction. *Expert Systems with Applications*, 55:441–451, August 2016.

[70] Jorge Urrutia. Chapter 22 - Art Gallery and Illumination Problems. In J. R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. North-Holland, Amsterdam, January 2000.

[71] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, February 2011.

[72] J. I. Vasquez-Gomez, J. Herrera-Lozada, and M. Olguin-Carbajal. Coverage Path Planning for Surveying Disjoint Areas. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 899–904, June 2018.

[73] Andrew Walker. Dubins-Curves: an open implementation of shortest paths for the forward only car, 2008–. https://github.com/AndrewWalker/Dubins-Curves.

[74] Y. Wang, T. Kirubarajan, R. Tharmarasa, R. Jassemi-Zargani, and N. Kashyap. Multiperiod Coverage Path Planning and Scheduling for Airborne Surveillance. *IEEE Transactions on Aerospace and Electronic Systems*, 54(5):2257–2273, October 2018.

[75] Steven L Waslander. Unmanned aerial and ground vehicle teams: Recent work and open problems. In *Autonomous control systems and vehicles*, pages 21–36. Springer, 2013.

[76] Yaojin Xu, Long Di, and YangQuan Chen. Consensus based formation control of multiple small rotary-wing uavs. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 909–916. American Society of Mechanical Engineers, 2011.

[77] Huili Yu, Kevin Meier, Matthew Argyle, and Randal W Beard. Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles. *IEEE/ASME Transactions on Mechatronics*, 20(2):541–552, 2015.

APPENDICES

Appendix A

Calculation of the Deadline Boundaries

A.1 Calculation of the Boundary of Deadline Tangents

Recall that the deadline is a line segment in $\mathbb{R}^2$ with endpoints $(D_{x,l}, D_{y,l}), (D_{x,r}, D_{y,r})$, and normal to the path at $P(t + \Delta t)$. For any possible path terminating at the coverage reachable set, the deadline is found by calculating the slope of the coverage reachable set with respect to θ, then using the slope to determine the right and left deadline endpoints. This line, tangent to the boundary for any given θ, can be found using the equations

$$D_x(\theta) = x(\theta) \pm \frac{dx}{d\theta}(\theta) w, \tag{A.1}$$

$$D_y(\theta) = y(\theta) \pm \frac{dy}{d\theta}(\theta) w \tag{A.2}$$

where $x(\theta), y(\theta)$ are defined by (4.2), (4.2) (See Figure 4.2.1). The normalized slope of the reachable set is parameterized by

$$\frac{dx(\theta)}{d\theta} = \frac{\cos\theta(vt - \rho\theta)}{|vt - \rho\theta|} = \cos\theta, \tag{A.3}$$

$$\frac{dy(\theta)}{d\theta} = \frac{-\sin\theta(vt - \rho\theta)}{|vt - \rho\theta|} = -\sin\theta \tag{A.4}$$

since $0 \leq \theta \leq \frac{v*t}{\rho}$, and $v, t, \rho \geq 0$. The full expression of the deadline is

$$D_{x,r}(\theta) = \rho(1 - \cos\theta) - \sin\theta(vt - \rho\theta) + w\cos\theta, \tag{A.5}$$
$$D_{y,r}(\theta) = \rho\sin\theta - \cos\theta(vt - \rho\theta) - w\sin\theta. \tag{A.6}$$
$$D_{x,l}(\theta) = \rho(1 - \cos\theta) - \sin\theta(vt - \rho\theta) - w\cos\theta, \tag{A.7}$$
$$D_{y,l}(\theta) = \rho\sin\theta - \cos\theta(vt - \rho\theta) + w\sin\theta. \tag{A.8}$$

The outer boundary expressed by these end points includes all of the possible positions of the deadline tangent to the ground vehicle's coverage reachable set, assuming the ground vehicle's path has a minimum radius turn for some angle θ, followed by a straight section for $d_{\text{map}} - \rho\theta$. We calculate this minimal boundary numerically using the parameters:

- v_{gv} is the speed of the ground vehicle,
- w is the corridor width, projected normal to the path at $P(t)$,
- Δt is the time interval required for the ground vehicle to traverse d_{map},
- ρ is the minimum turning radius for the ground vehicle, and
- θ is in the range $0 \leq \theta \leq \frac{v_{\text{gv}}\Delta t}{\rho}$.

For the movement reachable set RS_{mov}, we use t_{mov} for the time interval, where we assume t_{mov} is small, approaching 0. Considering the half of the plane where $x > 0$, the furthest extent of the shell is defined by the left end $(D_{x,l}, D_{y,l})$, as the deadline is swept to the right around the coverage boundary. To find the effect of the ground vehicle's movement, we transform the equation for the left end point by the movement function resulting in

$$D_{lx,t}(\theta) = d_{lx}(\theta)\cos\phi - d_{ly}(\theta)\sin\phi + x(\phi), \tag{A.9}$$
$$D_{ly,t}(\theta) = d_{lx}(\theta)\sin\phi + d_{ly}(\theta)\cos\phi + y(\phi). \tag{A.10}$$

An expression in two angles, θ and ϕ, where θ is the angle of the deadline set $D(t)$, with range $0 \leq \theta \leq d_{\text{map}}/\rho$, and ϕ is the angle of S_{gv} with range $0 \leq \phi \leq v\Delta t/\rho$. $x(\phi), y(\phi)$ are solutions to (4.2),(4.2), in the ground vehicle's reachable set. The full expansion of the equations is shown in (A.11), (A.12).

As the deadline D(t) is swept along, tangent to the reachable set, its end points extend beyond the boundary of the reachable set. The boundary that describes the outermost of

$$
\begin{aligned}
D_{lx,t}(\phi,\theta) = &(\rho(1-\cos\theta) - \sin\theta(d_{\text{map}} - \rho\theta) + w\cos\theta)\cos\phi \\
&- (\rho(1-\cos\theta) - \sin\theta(d_{\text{map}} - \rho\theta) + w\cos\theta)\sin\phi \\
&+ \rho(1-\cos\phi) - \sin\phi(vt - \rho\phi) \qquad \text{(A.11)} \\
D_{ly,t}(\phi,\theta) = &(\rho\sin\theta - \cos\theta(d_{\text{map}} - \rho\theta) + -w\sin\theta)\sin\phi \\
&+ (\rho\sin\theta - \cos\theta(d_{\text{map}} - \rho\theta) + -w\sin\theta)\cos\phi \\
&+ \rho\sin\phi - \cos\phi(vt - \rho\phi). \qquad \text{(A.12)}
\end{aligned}
$$

---

these points can be found using the angle of θ that maximizes the distance of the left end point from the origin. To calculate $\theta_{\max}$, we find the derivative of (A.6) with respect to θ, set it to zero and solve.

$$D_y(\theta) = \rho\sin\theta - \cos\theta(vt - \rho\theta) - w\sin\theta. \qquad \text{(A.13)}$$

Taking the derivative with respect to θ, assigning to zero, then solving gives

$$
\begin{aligned}
\rho\cos\theta + \sin\theta(vt - \rho\theta) - \rho\cos\theta - w\cos\theta = 0 \qquad &\text{(A.14)} \\
\sin\theta(vt - \rho\theta) - w\cos\theta = 0 \qquad &\text{(A.15)} \\
\tan\theta = \frac{w}{(vt - \rho\theta)}, \qquad &\text{(A.16)}
\end{aligned}
$$

which we can solve numerically for $\theta_{\max}$.

A.2 Calculation of the Radius of Deadline Lower Bound

Let the radius of the $\underline{D}(t)$ be represented as $r_{\underline{D}}$. In 4.4 we claimed $r_{\underline{D}}$ is calculated using:

$$r_{\underline{D}} = r_{\text{gv}} + \sqrt{r_{\text{gv}}^2 + \left(\frac{L}{2} + \frac{\sqrt{L^2 - 4r_{\text{gv}}^2}}{2}\right)^2},$$

where L is the distance from either end of the deadline to the ground vehicle, when the deadline is located at top dead centre. L is calculated

$$L = \sqrt{d_{map}^2 + \frac{w^2}{2}} - \frac{\pi}{2} r_{\text{gv}}.$$

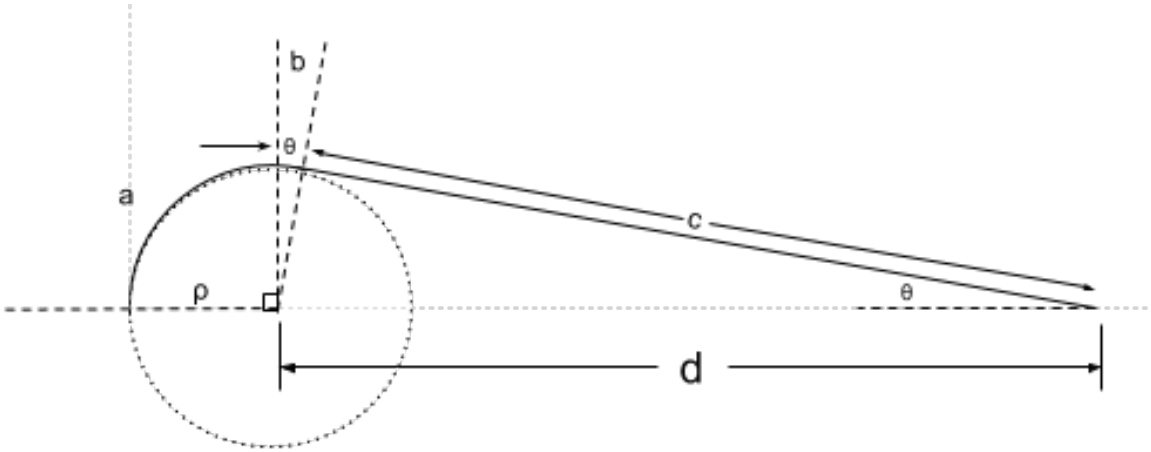


Figure A.1: Calculation of the inside estimate

Conceptually, to find the minimum radius, consider a taut string that runs from the ground vehicle to one end of the deadline, when the deadline is straight ahead of the vehicle, and sitting on the boundary at d_{map}. Now, consider what happens as the deadline is swept around the boundary, maintaining the tension on the string. As deadline is moved along the boundary, the string wraps around the minimum turning radius circle, shortening as it goes. Eventually, the end of the string crosses the x-axis, at some point less than d_{map} from the centre. This point of crossing the x-axis defines the minimum radius of a semi-circle that fits entirely within the deadline boundary.

Figure A.1 illustrates the calculation of the radius estimate. The length of the string L can be divided into three sections. The first section, a, of length $\frac{\pi}{2}r_{\text{gv}}$ wraps around the first quadrant of the minimum turning circle. The second section, b, has length $r_{\text{gv}}\theta$ and extends from the start of the second quadrant of the minimum turning circle to the tangent point where the string leaves the circle and heads towards the x-axis. Due to similar angles, the section of contact of the minimum turning circle has the same angle as the angle of the string when it finally contacts the x-axis. Finally, the third section, c, extends from the tangent to the x-axis and has length of $L - \frac{\pi}{2}r_{\text{gv}} - r_{\text{gv}}\theta$. For convenience, let $l = L - \frac{\pi}{2}r_{\text{gv}}$. Since the triangle defined by the third section, the radius and the x-axis

is right-angled, then we know:

$$\tan\theta = \frac{r}{l - r_{\text{gv}}\theta}$$

Using the small angle approximation[1], we can approximate $\tan(\theta) \approx \theta$. We also know that $\tan(\theta) \geq \theta$, for $0 < \theta < \frac{\pi}{2}$. Therefore,

$$\theta \leq \frac{r_{\text{gv}}}{l - r_{\text{gv}}\theta}$$

$$r_{\text{gv}}\theta^2 - l\theta + r_{\text{gv}} \geq 0. \tag{A.17}$$

This is a quadratic equation in θ – solving using the quadratic formula gives us the expression:

$$\theta = \frac{l \pm \sqrt{l^2 - 4r_{\text{gv}}^2}}{2r_{\text{gv}}}. \tag{A.18}$$

We can discard the larger (plus) root since we know we're looking for small theta – the larger value of theta is invalid for our approximation and our expected use case (where $d_{\text{map}} \gg r \Rightarrow \theta < \frac{\pi}{2}$). The smaller (minus) root approaches zero as d_{map} becomes larger relative to the radius, exactly what we are expecting.

Since we have a right angled triangle, we also know the length of the hypotenuse can be found using Pythagoras's theorem. Therefore,

$$d = \sqrt{r_{\text{gv}}^2 + (l - r_{\text{gv}}\theta)^2}.$$

Substituting our equation for θ from A.18,

$$\begin{aligned} d &= \sqrt{r_{\text{gv}}^2 + \left(l - r_{\text{gv}}\left(\frac{l - \sqrt{l^2 - 4r_{\text{gv}}^2}}{2r_{\text{gv}}}\right)\right)^2} \\ &= \sqrt{r_{\text{gv}}^2 + \left(\frac{l}{2} + \frac{\sqrt{l^2 - 4r_{\text{gv}}^2}}{2}\right)^2}. \end{aligned}$$

The radius, $r_{\underline{D}}$, is calculated by adding the ground vehicle's turning radius:

$$r_{\underline{D}} = r_{\text{gv}} + \sqrt{r_{\text{gv}}^2 + \left(\frac{l}{2} + \frac{\sqrt{l^2 - 4r_{\text{gv}}^2}}{2}\right)^2}.$$

[1] See Appendix A.3 for justification.

A.3 On the Use of the Small Angle Approximation

In the development of the lower bound on the deadline $\underline{D}$ we make use of the small angle approximation. We prove that doing so is appropriate and always gives an underestimate of the true value.

Lemma A.3.1 (Small Angle Approximation)**.** In the circumstances involved in this thesis, the use of the small angle approximation gives an estimate that is always equal to or less than the actual value of $r_{\underline{D}}$.

Proof. (By Construction.) Consider the original equation $r_{\underline{D}}$:

$$r_{\underline{D}} = r_{\text{gv}} + \sqrt{r_{\text{gv}}^2 + (l - r_{\text{gv}}\theta)^2}. \tag{A.19}$$

We have the following facts to consider:

- The $(l - r_{\text{gv}}\theta)$ term of (A.19) increases in size as θ decreases.
- Geometrically, $\theta > 0$ since a non-zero r_{gv} precludes $\theta = 0$ and negative angles are infeasible.
- The function $f(\theta)$ (A.17) is quadratic and strictly increasing for $\theta < \frac{l - \sqrt{l^2 - 4r_{\text{gv}}^2}}{2r}$.

Therefore, our estimate of θ, $\theta = \frac{l - \sqrt{l^2 - 4r_{\text{gv}}^2}}{2r_{\text{gv}}}$ is the maximum value of θ. Therefore, $r_{\underline{D}}$ is inside the true boundary and our estimate can be used to represent the necessary UAV velocity. □

Glossary

boustrophedon decomposition A method of subdividing an area to be covered such that each cell may be covered by a back and forth pattern (modeled on the path taken by an ox in plowing a field). 5

Coverage Path Planning (CPP) The process of constructing a plan over an area that, when followed by a robot, results in all portions of that area being observed. 4, 10

coverage plan A line, or set of connected lines, laid out over a specified area. All parts of the area fall within the range of a UAV's sensors from at least one location in the plan. 3

divide-and-conquer The process of dividing a problem into pieces, then solving the simpler problem represented by the smaller pieces. The final solution is retrieved from the solutions to the sub-problems. 4

Generalized Large Neighbourhood Search (GLNS) An algorithm for solving the (Generalized) Travelling Salesperson Problem. This variation allows vertices in the graph G to be grouped, creating opportunities for both local and global optimizations and constraints. 5

holonomic Holonomic robots are capable of unrestricted movement in any direction. An airborne UAV is an example. 69

lawn mower A method of covering an area by following a pattern of adjacent strips, as one might cut a lawn. 3

Mixed Integer Linear Program (MILP) A sub-field of Linear Programming where the variables and constraints are a mixture of integer and real values. MILPs can

become computationally intractable as the number of variables and constraints grows. 4

NP-hard (Non-deterministic Polynomial hard) – a measure of an algorithm's complexity. NP-hard problems are not known to be solvable in polynomial time, but are verifiable in polynomial time. 5, 11

Orienteering Problem The set of problems that compute the optimal path through a graph, visiting as many vertices as possible, subject to some resource constraint. Constraints may include limitations on travel distance (limited fuel), or carrying capacity (limited space), etc. 11

sensor footprint The area on the ground that falls within the sensor range of the robot. In the case of a UAV using monocular vision system (a single downward facing camera), the sensor footprint is the size of the image area for a given flight altitude. 4

Travelling Salesperson Problem (TSP) A computed shortest path tour of a graph G that visits each vertex at most once and returns to the start. 6, 11

unmanned aerial vehicle (UAV) An Unmanned Aerial Vehicle - an airborne robot carrying various sensors that may be piloted remotely through teleoperation or have a degree of autonomy. Sometimes referred to as a Drone. 1

unmanned ground vehicle (UGV) An Unmanned Ground Vehicle - a land-based robot, either autonomous or remotely operated. 7