8-2024

Advancing Unmanned Ground Vehicle Path Planning With Quantified Map Uncertainty

Israel Afriyie
iafriyi@clemson.edu

ADVANCING UNMANNED GROUND VEHICLE PATH PLANNING WITH QUANTIFIED MAP UNCERTAINTY

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Civil Engineering

---

by
Israel Afriyie
August, 2024

Accepted by:
Pamela Murray-Tuite, Committee Chair
Matthias Schmid
Mashrur Ronnie Chowdhury

ABSTRACT

This thesis addresses the complex challenge of path planning for Unmanned Ground Vehicles (UGVs) in areas where traditional navigation systems are inadequate, such as unstructured or off-road military zones. Recognizing the limitations of current path planning algorithms, which primarily focus on optimizing for the shortest path and often fail to account for variability and risks, this research proposes an enhanced Hyperstar algorithm. This approach not only considers the fastest route but also integrates maximum delays and visibility risks into its computation, ensuring a balance between swift mission completion and concealment from adversaries.

Utilizing terrain maps and incorporating uncertainties in map data through Monte Carlo simulations, the study evaluates the algorithm's effectiveness across various scenarios, including different levels of visibility risk, maximum delay, and bidirectional navigation challenges. The algorithm's adaptability is demonstrated through numerical examples and terrain tests, highlighting its ability to offer multiple route choices so the vehicle can adjust to environmental changes.

DEDICATION

I dedicated this thesis to my family and friends for their love and support.

ACKNOWLEDGMENTS

I would like to express a special thanks to my advisor Dr. Pamela Murray-Tuite for providing me with the opportunity to work on this research with the proper guidance direction, and suggestions throughout the work process. Special thanks go my committee for their guidance in shaping the thesis. This work was supported by the Simulation Based Reliability and Safety Program for modeling and simulation of military ground vehicle systems under the technical services contract No.W56HZV-17-C-0095 with the U.S. Army DEVCOM Ground Vehicle Systems Center (GVSC) and by Clemson University's Virtual Prototyping of Autonomy Enabled Ground Systems (VIPR-GS), under Cooperative Agreement W56HZV-21-2-0001 with the US Army DEVCOM Ground Vehicle Systems Center (GVSC).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Modern navigation systems have significantly evolved, thanks to the latest technology that assists in finding the quickest paths for journeys. These systems work well in cities and highways, where the roads are well-defined. But in places that do not have clear roads or any roads at all, such as military zones, unmanned ground vehicles (UGVs) are critically important as they can operate where manned vehicles cannot (Roy et al., 2018). UGVs require planning to avoid delays and detection (Zhao et al., 2021). They perform tasks such as gathering intelligence, identifying enemy locations, transporting necessary supplies, safely handling bombs, discreetly detecting hazardous substances like chemicals or radiation, aiding in the rescue and evacuation of injured soldiers, and establishing communication in difficult areas (Chabini & Lan, 2002). The goal of this thesis is to develop path-planning techniques that balance between quick mission completion and avoiding being seen to boost the effectiveness and safety of UGVs. To navigate through complex and uneven terrains swiftly and safely, a global path planning strategy is essential. Some existing path planning algorithms focus on optimizing a single criterion, such as the shortest distance path; however, for effective navigation in off road environments, especially for military vehicles, a path planning approach that simultaneously considers travel time variability and visibility is required to ensure reliable and efficient traversal.

Travel times were inputs to the path planning. The range of travel times were computed for different types of terrain under both dry and wet conditions, considering the variations in soil friction coefficients for these conditions. The longest and shortest times

it could take the vehicle to traverse a segment were then chosen for use in the planning. The difference between the longest time and the shortest time was the maximum delay. This information allows estimation of the best and worst-case scenarios for travel time and planning routes to minimize potential delays.

Another criterion was visibility risk, which combines how likely a vehicle is to be seen by adversaries with the potential consequences of destruction. The greater the risk of the vehicle being seen, the higher the chance of it being compromised. Thus, managing this risk in the path planning is crucial for a successful mission where the vehicle remains undetected. The offroad challenge is addressed through a framework that starts by discretizing the environment into a grid and leverages the key terrain classifications such as soil types, slope variations, and surface roughness, to estimate vehicle speeds across the terrain. Inaccuracies in elevation maps can introduce a level of uncertainty, particularly in assessing the likelihood of detection. This thesis uses a method for creating possible realizations of maps (based on reported map error rates) that show which areas are visible. The actual path planning is performed using a modified version of the Hyperstar algorithm. The modification incorporates the likelihood of being seen to the previously considered travel time while accounting for potential delays (Bell et al., 2012; Hosseinloo et al., 2012) (Spiess & Florian, 1989). Instead of just going for the fastest route, the Hyperstar algorithm offers the vehicle multiple route options. The modification prioritizes the vehicle's safety, allowing for decisions between these routes, giving the vehicle flexibility. This thesis outlines a methodology for generating a suite of potential routes with various deviations, aiming to enhance the adaptability of navigation systems within unstructured

environments. While the study stops short of demonstrating the significance of such adaptability, it lays the groundwork for future research that could transform navigation practices across diverse and challenging terrains.

1.1 Research Objectives

The essence of this research lies in enhancing the efficiency and safety of UGV operations, addressing challenges of navigating in unstructured environments. Recognizing the nature of this challenge, this study uses a methodology that not only seeks efficiency in travel but also prioritizes minimizing exposure to potential threats. The following objectives are designed to guide the exploration and development of advanced pathfinding solutions. This research aims to:

- Develop a cost-effective path with options for UGVs from a starting point to a destination that minimizes travel time and visibility risk (detection by the enemy vehicle) in an unstructured environment,
- Quantify the visibility risk, taking into account the errors present in maps, and
- Incorporate a method to assess delay in an unstructured environment.

1.2 Contribution

In our research, we found a path in an unstructured environment by modifying a path-finding algorithm called the Hyperstar algorithm (Bell et al., 2012). Due to possible delays and the risk associated with the visibility of each link, there is a growing interest in algorithms that can provide several potentially optimal paths, collectively known as a hyperpath. Florian and Spiess‘s algorithm (Spiess & Florian, 1989) for generating

hyperpaths in transit networks is adapted in an unstructured environment by assuming that military vehicles follow the safest possible route whenever a choice of path arises. Traditional algorithms such as Dijkstra's (Saranya et al., 2016) and A*(Chabini & Lan, 2002) emphasize finding the shortest path based on predefined criteria but do not preplan the potential need for route adaptability in response to environmental changes or inaccuracies. The challenge with these approaches is their inherent limitation in flexibility; they are designed to calculate the best route from start to finish without accounting for the possibility of needing to switch paths midway. Having just a single path can be particularly problematic in scenarios where UGVs encounter unexpected obstacles or changes in the terrain that render the original path impractical or unsafe. Moreover, when multiple routes are considered by some algorithms (Eppstein, 1998), these alternatives are often not interconnected, restricting the UGV's ability to transition smoothly between paths. This limitation is noted by Sacramento et al. (2019), where the focus has been on improving the efficiency and reliability of single-route planning, with less emphasis on the flexibility required in complex or hazardous environments. This research seeks to address these gaps by implementing a modified version of the Hyperstar algorithm in an unstructured environment which not only plans multiple routes, but the paths are interconnected, allowing for the vehicle to switch in response to environmental changes. Most existing literature considers visibility (detection) in a deterministic way, ignoring the errors in the terrain maps they rely on (Nackaerts et al., 1999). This thesis considers the errors in the maps in the determination of visibility risk.

1.3 Thesis Organization

The structure of this thesis is as follows. Chapter 2 begins with the concept of path planning and its importance in various fields like autonomous vehicle and robot navigation as well as the idea guiding the establishment of these path planning algorithms. The chapter goes ahead to identify some gaps in literature with regards to path planning especially in handling uncertainties and considering multiple objectives. Chapter 3 explains how the environment was modelled and how risk was represented. The chapter also delves further into how we computed travel time for the paths to be generated. An approach for estimating maximum delays in travel time is also shown in this chapter with a description of how travel time, delay, and visibility were considered. Chapter 4 illustrates implementing the algorithm to generate multiple paths in the environment and analysis of the results. Chapter 5 summarizes the research major findings and discusses the limitations.

CHAPTER 2

LITERATURE REVIEW

Path planning is a critical component in autonomous vehicles, and various other fields requiring navigation through space. It involves generating a viable route from a starting point to a destination while avoiding obstacles and optimizing for certain criteria (e.g., shortest distance, minimal time, or lowest energy consumption). Path planning in this context (movement through an off-road, adversarial environment) requires consideration of three interactive elements illustrated in Figure 1.

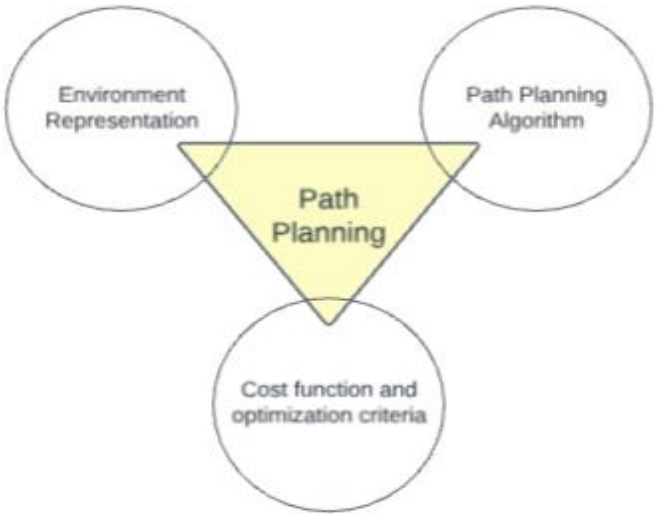

Figure 1: Path Planning Problem Elements

2.1 Environment Representation

The initial stage in solving a path planning problem is to represent the environment geometrically or graphically. Several methods of representing the environment have been developed for use in path planning. These representations are classified into two groups(Hua et al., 2022) :

- Classical Approach
- Reactive Approach

2.1.1 Classical Approach

The classical approach to path planning often involves creating a static model of the environment. In this model, all the obstacles and goals are mapped out in advance (Lv & Feng, 2017). The classical approach works well in predictable settings but can struggle with unexpected obstacles or changes (Zhang et al., 2020). Other challenges with respect to this approach are that it is highly computationally expensive, and it fails to respond to uncertainty in the environment (Marzouqi & Jarvis, 2006). The classical approach can also be further divided into Cell Decomposition, Roadmap, and Artificial potential field approaches (Adzhar et al., 2020).

2.1.1.1 Cell Decomposition

The cell decomposition method divides the region into non-overlapping cells. Pure cells are the cells without obstacles while corrupt cells are the cells that contain obstacles (Lingelbach, 2015). The cell decomposition method can be further divided into adaptive, approximate, and exact methods.

- Adaptive method.

  Adaptive cell decomposition follows the basic idea of regular cell decomposition by using information from open areas (Wang et al., 2022). This method is designed to minimize the quantity of cells in an open space, capitalizing on the redundancy of free space information in standard cell decomposition (Wang et al., 2022). The adaptive method faces challenges, especially in changing environments. Whenever

there is a change in the surroundings, it often requires updating the entire map to reflect these changes accurately (Lingelbach, 2015). Also, the approach is computationally expensive (Lv & Feng, 2017).

- Approximate cell decomposition.

  The approximate cell decomposition is when the grid is laid over a planning space. The cells are designed with predetermined shapes and sizes, making them simple to use. This method is termed approximate because the boundaries of physical objects do not always align with the predefined cell boundaries. However, the issue relating to this method is that an object smaller than the grid size will result in the entire grid being occupied and will be labelled as occupied leading to a conservative estimate of the free space and some space that is passable will be labelled as impassable (Zhu & Latombe, 1989).

- Exact cell decomposition method.

  In this approach, the cells laid over the environment do not have a predefined shape and size, but they are determined based on the predefined shape and size of the obstacle within it (Zhu & Latombe, 1989). The boundaries of the cell correspond to the boundaries of the free space available. However, this method is quite difficult to apply in an off-road environment where there are no predefined obstacles (Zhu & Latombe, 1989).

2.1.1.2 Roadmap approach

In this approach, the links between the robot's free space are represented as a collection of one-dimensional curves (Varadhan et al., 2006). In comparison to cell decompositions,

roadmaps offer a significant advantage in terms of the quantity of nodes a planner must look through to identify a path (Hauser & Latombe, 2010). Some examples of the road map approach are explained below:

- Voronoi Diagrams

  The method involves paths or edges that are equidistant from every point in the obstacle region. The paths are generated so they stay away from obstacles. This way, if the robot follows these paths, it will naturally avoid obstacles. However, most of the generated paths from this approach are inefficient (Hauser & Latombe, 2010).

- Probabilistic Roadmaps

  Probability road maps tend to generate paths in a large dimensional space. The approach randomly samples large space points and connects them into a roadmap. One issue with this method is its inefficiency in narrow, confined spaces due to the points are chosen randomly and the likelihood of capturing a random point in a small space is low, resulting in a lack of connectivity. (Kavraki et al., 1996).

- Visibility Graphs

  The visibility graphs method comprises straight-line segments that connect the nodes of polygonal obstacles without crossing the interiors of the obstacles. These straight lines form the path which the vehicle will traverse (Kumar et al., 2022). This approach is not ideal because the calculated path is tangential to the obstacles, which can lead to the robot colliding with them. (Gao et al., 2019).

2.1.1.3 Artificial Potential Field Approaches

Artificial potential fields are another major type of representation used in path planning. This approach applies a mathematical function over the entire area of the robot's travel. (Hwang & Ahuja, 1992). This method considers the robot as a point influenced by fields generated by goals and obstacles in the environment, like an electron in an electric field. (Sabudin et al., 2021). The closer the robot is to an obstacle or goal, the stronger these forces are. The robot moves in the direction these forces push or pull it (Agarwal & Akella, 2022). This method can work in both simple and complex spaces. However, one issue with this method is that sometimes the robot might get stuck in a spot that is not the goal because of these forces (Loquercio et al., 2018).

2.1.2 Reactive approach

The reactive approach can handle uncertainty. The reactive approach is a type of motion planning that is designed to respond to changes in the environment in real time. It is typically used in robot navigation where the robot needs to navigate through a dynamic environment such as a warehouse or a manufacturing facility (Silva & Ribeiro, 2003). A reactive motion planner works on continuously monitoring the environment and the robot’s current state and generating a sequence of actions in response to any changes it detects. Reactive motion planners are often used in conjunction with other types of motion planning algorithms such as probabilistic or sampling-based planners to provide a more robust and flexible solution for navigating through dynamic environments (LaValle & Kuffner, 2001).

2.2 Cost Function and Optimization Criteria

Path planning involves computing optimal paths for robots or vehicles to navigate from a starting point to a target location while considering various criteria such as path length, computational complexity, completeness, and specific cost functions (Liu et al., 2017). Cost functions play a significant role in evaluating the quality of paths and are often used to minimize certain parameters like path length, time, energy consumption, or threat potential (Vlaski & Sayed, 2019).

Several studies have concentrated on optimization criteria and cost functions to either maximize or minimize specific objectives. For instance, Karaman and Frazzoli (2011) emphasize the significance of minimizing cost functions related to path length or execution time in their optimization criteria. Similarly, Shorakaei et al. (2014) focus on minimizing both path length and potential path threats in their optimization objectives for unmanned aerial vehicles. Additionally, path planning criteria often involve addressing challenges such as obstacle avoidance, energy optimization, and dynamic environments. Karaman and Frazzoli (2011) investigated multiple destination path planning for mobile robots, aiming to find paths that pass through the fewest obstacles while optimizing the mobile node's trajectory.

Another form of cost criteria that is considered in off-road path planning is risk. Risk management is a vital factor that must be considered for safe and effective (Revela, 2001). One aspect of this risk management involves stealth, where minimizing exposure to potential observers or other hazards becomes essential. Stealth is important in off-road navigation because it considers the unique and potentially difficult circumstances

encountered in an off-road environment (Revela, 2001). Marzouqi and Jarvis (2006) consider the risk of interaction with unfriendly agents and model the area in the line of sight of enemy towers as risky. However, the enemy movements and behavior are likely dynamic. Carr (2014) discusses the usefulness of visibility analysis for calculating intervisibility and zones of visual influence across various application areas. Revela (2001) introduced a novel approach for stealth navigation of a mobile robot in a cluttered environment. The robot must plan a covert path to reach a specified destination while minimizing its exposure to hostile sentries.Roy et al. (2018) uses a binary risk representation, but the areas deemed risky may need to be travelled anyway (in certain circumstances) and the area deemed safe might have some risk associated with it.

Risk is usually a product of the consequence and its probability (Hosseinloo et al., 2012). In path planning, risks are typically categorized as 'Low-Probability-High-Consequence' (LPHC) or 'High-Probability-High-Consequence' (HPHC) (Hosseinloo et al., 2012). LPHC risks are those where the chance of occurrence is low, but the consequences, like destruction, are severe. Due to their rarity, quantifying the incident probability of LPHC risks is challenging, so the focus is more on minimizing the consequences (Chen et al., 2021). HPHC risks, on the other hand, have both a high chance of occurrence and severe consequences. Depending on available data, efforts may aim to minimize either the incident probability, the consequences, or both.

2.3 Path Planning Algorithms

Various path planning algorithms have been developed to address the challenges in unstructured environments. Examples of these challenges include the presence of

obstacles, rough terrain, and unknown surroundings leading to uncertainty in the environment. With these considerations, the selection of a suitable algorithm is a key for successful operation. Path planning algorithms can be divided into traditional algorithms and modern intelligent algorithms (Wang et al., 2022). Traditional path planning algorithms such as Dijkstra, and A* rely on precomputed paths based on static maps of the environment (Wei & Ren, 2018). On the other hand, modern intelligent path planning algorithms incorporate advanced techniques like machine learning and neural networks to allow for real time learning in dynamic environments with changing conditions (Albizu et al., 2020). Another way of grouping path planning approaches is global path planning and local path planning (Wei & Ren, 2018). Global path planning involves planning paths in a completely known environment (position and shape of obstacles) (Wei & Ren, 2018). Global path planning first needs to establish an abstract global environment map model based on known global environment information and then uses the search algorithm to obtain the global optimal path on the whole region (Hua et al., 2022). At present, commonly used methods for global path planning include genetic algorithms, fast random search tree algorithm and bee colony algorithm (Hua et al., 2022). There are other heuristic algorithms such as the A* and Rapidly exploring Random Tree (RRT), which explores the search space by randomly building a space-filling tree (Xu et al., 2022), and Probabilistic Roadmap Method (PRM) which constructs a network of randomly sampled points in the free space of the environment (Xu et al., 2022). Local path planning is real time dynamic path planning based on local information collected by sensors during the navigation of the off-road vehicle (Xu et al., 2022). The positions of obstacles in the environment are

completely unknown so the robot/vehicle moves about the environment with sensors (Xu et al., 2022). Local path planning algorithms are designed to react to changes in the immediate surroundings (Lv & Feng, 2017). These algorithms focus on making fast decisions using current data from sensors like LIDAR, radar, and cameras. This data helps the system quickly understand what's around it (Hodson, 2022).

In the area of robotic and autonomous navigation, there is a growing interest in developing new ways for vehicles to find their paths. Researchers are creating hybrid algorithms that mix different path planning strategies to better handle unpredictable spaces with moving obstacles (Sabudin et al., 2021). These hybrid algorithms aim to combine the strengths of global and local path planning strategies to navigate complex environments efficiently while considering dynamic obstacles and other constraints (Chen et al., 2019; Dinelli et al., 2023). Furthermore, Chen et al. (2019) proposed a hybrid path planning algorithm tailored for unmanned surface vehicles operating in complex environments with dynamic obstacles. Yan et al. (2021) conducted an analysis of a hybrid global path planning algorithm for different environments, emphasizing the importance of hybrid algorithms in achieving safe and efficient path planning for automated guided vehicles and enhancing traffic safety. Yan et al. (2021) introduced a hybrid PSO-GSA path planning algorithm that combines the local search capability of the gravitational search algorithm (GSA) with the social thinking ability of the particle swarm optimization (PSO) algorithm for robot path planning in static environments with danger zones. The development of hybrid path planning algorithms has also shown a significant advancement in the field of robotics, enabling efficient navigation in complex and dynamic environments. By combining global and local planning strategies,

these algorithms offer enhanced adaptability and performance for various robotic applications (Chen et al., 2019; Bell et al., 2012).

Zhao et al. (2021) uses various techniques in creating alternate paths for navigation. These paths have been generated using two main types of algorithms: the k-shortest path algorithms, as introduced by Eppstein (1997), and the completely disjoint path algorithms, as proposed by Chen et al. (2019). Nonetheless, these approaches exhibit certain limitations for navigational purposes. Specifically, k-shortest path algorithms require an extra step of path validation to ensure they meet certain criteria, which becomes cumbersome and less efficient with a large set of paths (Guo et al., 2022). Furthermore, these paths often have significant overlap. In contrast, disjoint path algorithms may exclude the most direct path or produce alternatives that are too lengthy to be practical (Guo et al., 2022). More recent studies have focused on identifying k partially disjoint paths that adhere to specific constraints, with the link penalty method by Chen et al. (2021) being a notable example in this area. Yan et al. (2021) developed a method to identify distinct path sets by evaluating the spatial differences between paths, choosing those that maximize dissimilarity based on the shortest distance between them. Zhao et al. (2021) offered a solution to give vehicles different route options by incrementally increasing link impedance, a concept further refined by Guo et al. (2022). through an algorithm that minimizes the sharing of links, utilizing Dijkstra's algorithm and a logarithmic link penalty approach.

Additionally, the challenge of creating multiple paths is crucial for analyzing vehicles' route selection behaviors, as reviewed by Ding et al. (2023). The ideal route set should filter out improbable routes that vehicles are unlikely to choose and highly similar routes

that are indistinguishable to travelers (Wagner & Choset, 2015). To address this, both deterministic and probabilistic methods for route creation have been adopted.

The routing approach used in our study is related to the prior literature on multiple paths and global planning. Table 1 illustrates some of the most closely related literature on global path planning algorithms. Some key criteria of path planning algorithms include risk aversion, path optimality generation, heuristics, adaptability to terrain and robustness.

- Risk aversion in path planning involves considering uncertainties and potential hazards in the environment to ensure safe navigation. By incorporating risk-averse strategies, robots can avoid dangerous situations and prioritize safety during path traversal (Wagner & Choset, 2015).
- Path optimality is crucial for identifying the most efficient route from the start to the destination. Optimal path planning aims to minimize certain criteria such as distance traveled, time taken, or energy consumption while ensuring the robot reaches its destination successfully (Ziebart et al., 2009).
- Heuristics play a significant role in guiding the search process during path planning. Heuristic functions provide estimates of the cost to reach the goal from a given state, aiding in decision-making, and enhancing computational efficiency (Helmert, 2006).
- Adaptability to terrain is essential for robots to navigate diverse environments effectively. Path planning algorithms that can adapt to different terrains, such as rough terrain or dynamic environments, enable robots to adjust their paths based on the surrounding conditions (Qian et al., 2020).

- Robustness in path planning ensures that the planned path remains viable even in the presence of uncertainties or changes in the environment. Robust path planning algorithms can handle variations in terrain, obstacles, or other factors without significant degradation in performance (Duan & Qiao, 2014).

Risk aversion in path planning is crucial, as it involves formulating strategies against uncertainties and potential hazards to ensure safe navigation. Algorithms like A* incorporate this principle by balancing risk and safety. Specifically, the A* algorithm shows medium risk-averse performance, operates with a forward search direction, and provides medium path optimality while incorporating heuristics for high adaptability to terrain and robustness, as noted by Karaman and Frazzoli (2011) and supported by other sources (Tian et al., 2021; Ganganath et al., 2015).Yen's algorithm, in contrast, delivers a similar level of risk aversion and path optimality without relying on heuristics and maintains a forward search direction, indicating its efficiency in pathfinding (Chen et al., 2021). The Spiess and Florian approach, on the other hand, is characterized by a high-risk aversion and a reverse search direction (Spiess & Florian, 1989). This approach achieves high path optimality and demonstrates significant robustness, even though it lacks adaptability (Spiess & Florian, 1989). The Genetic algorithm can be combined with approaches like the Probabilistic Road Map to achieve medium risk aversion and high path optimality. This combination shows promise due to its forward search direction even though it does not apply heuristics within its process (Alam et al., 1996). The RRT algorithm exemplifies a medium stance on risk aversion and path optimality, prioritizing robustness in unpredictable terrains without the use of heuristic functions (Liu et al., 2017).

This robustness is a key feature of the Spiess and Florian algorithm as well, which guarantees reliable performance amidst environmental changes. Overall, these algorithms demonstrate a spectrum of capabilities in path planning, from high adaptability and robustness to effective risk (uncertainty) management. Each has its unique approach to handling the complexities of path planning in diverse environments, underscoring the importance of choosing the right algorithm based on the specific requirements of the task at hand.

Table 1: The criteria for some existing path planning algorithm

Algorithm Type	Literature	Risk-Averse Performance	Search Direction	Path Optimality	Heuristic	Adaptability to terrain	Robust
A*	(Foead et al., 2021; Hong et al., 2021; Yan, 2023)	Medium	Forward	Medium	Yes	High	High
Dijkstra	(Javaid, 2013)	Low	Forward	High	No	Medium	Medium
Hyperstar	(Bell, 2009; Bell et al., 2012; Ma et al., 2013; Verbas & Mahmassani, 2015)	High	Reverse	High	Yes	High	High
Yen's Algorithm	(Yen, 1971)	Medium	Forward	High	No	High	Medium
Spiess and Florain	(Bell, 2009; Spiess & Florian, 1989)	High	Reverse	High	No	N/A	High
Genetic	(Alam et al.,1996)	Medium	Forward	High	No	High	High
RRT	(Ding et al., 2023)	Medium	Forward	Medium	No	Medium	High

2.3.1 Hyperstar Algorithm

Traditional path planning algorithms, such as Dijkstra's algorithm and the A* algorithm, have been foundational in the development of navigation systems. Dijkstra's algorithm is celebrated for its simplicity in finding the shortest path in a weighted graph (Javaid, 2013).

The A* algorithm introduces heuristics to improve on Dijkstra's approach by prioritizing paths that are seemingly closer to the goal, enhancing efficiency (Chen et al., 2021). However, both algorithms primarily focus on path generation and do not inherently account for the management of navigation-related risks (Chen et al., 2021).

In contrast, the Hyperstar algorithm is a global path planning algorithm that emerges as a solution designed to address the limitations of traditional path planning algorithms by incorporating risk (uncertainty) directly into its computational framework. The algorithm was first developed by Spiess and Florain to be used in transit networks (Spiess & Florian, 1989). In their model it was assumed that with a fixed set of transit lines, travelers can choose an optimal path known as a hyperpath that will allow them to reach their destination at a minimum cost while considering delay time. Spiess and Florian (1989) found that the choice of path can be determined by finding the fastest path on average. This type of problem is solved using linear programming and this ended up with a method like a well-known pathfinding method (Dijkstra's algorithm) to find the best path, starting from the destination and working backward. Spiess and Florian's study considered a similarity between how often a link operates over a given route (frequency) and the longest waiting (maximum delay) (Spiess & Florian, 1989). Spiess and Florian's method was improved by considering additional factors in choosing the path such as the node potential which helps guide the search of the algorithm from start to finish, making the approach like the A* algorithm. This method was called the Hyperstar algorithm (Bell et al., 2012). The original routing strategy problem formulated by Bell et al. (2012) has a service frequency for each link that may be subjected to delay. The service frequency is defined as the inverse

of maximum delay. Vehicles are assumed to be interested in all potentially optimal paths when adopting a minmax exposure to delay strategy. Hosseinloo et al. (2012) also modified the Hyperstar algorithm in generating paths for transporting hazardous materials. They addressed the risk associated with transporting hazardous materials by focusing on Low-Probability-High consequences (LPHC) events. Their strategy focused on reducing the consequences rather than the incident probability (Hosseinloo et al., 2012). The advantage of the Hyperpath algorithm over single route algorithms is that reasonable detours are considered at the route generation stage. If routes are only planned assuming no delays, the vehicle might end up choosing a path that is the fastest under perfect conditions but could easily experience delay. On the flip side, if an assumption is made for the worst-case scenario for every part of the journey (maximum delays everywhere), the vehicle might miss out on faster routes because it is overly focused on avoiding delay. The Hyperstar algorithm, as previously developed, balances these two extremes. It does not just pick the fastest route under ideal conditions, nor does it avoid delay at all costs. Instead, it plans for the possibility of delays and considers alternate routes (detours) that might become preferable if there are delays or other issues on the main route. This way, when the vehicle is on the road and learns about problems ahead, it already has a plan for a good alternative route, making the journey more flexible and potentially quicker and less subject to delay. As the delays within the network grow, the algorithm devises alternative routes for the vehicle. This ensures that the vehicle can still reach its destination efficiently, despite any unexpected hold-ups along the way (Bell et al., 2012).

In this thesis, the Hyperstar algorithm is adopted specifically for navigating UGVs in unstructured environments where visibility risk plays a role in path selection. Prior research has explored the use of risk assessment models in autonomous navigation, but often these models do not directly influence path planning decisions (Wagner & Choset, 2015). In sum, while traditional path planning algorithms provide a solid foundation for navigation system development, their limitations in adaptability and risk management are evident in the context of unstructured environments. Integrating visibility risk into the Hyperstar algorithm for path planning process represents a contribution to the field, potentially offering a more holistic solution to the challenges faced by UGVs.

CHAPTER 3

METHODS

This chapter explains the framework used in this study. Given the complexity of navigating in unstructured environments, a comprehensive approach combining quantitative analysis, simulation techniques, and a shortest path-based algorithm was adopted. This includes a representation of the environment, estimation of potential visibility by an opposition, estimation of travel time and maximum delay time for each link in the network, and modification of the Hyperstar algorithm. Figure 2 presents a structured overview of the framework employed in the thesis.

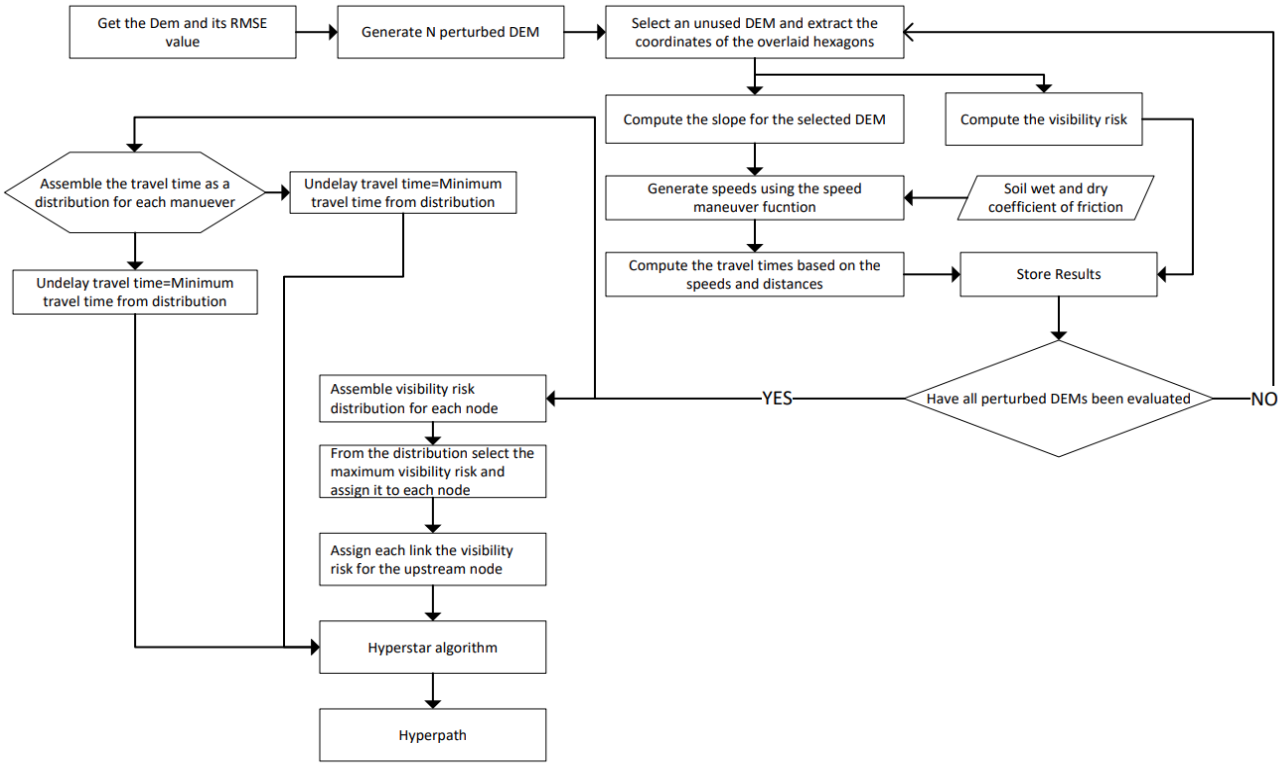

Figure 2: An Overview of the Framework

Step 1: Environment Representation

This step translates the environment into a graph structure. The environment is discretized into identically sized cells, with each cell centroid acting as a node and adjacent cells are connected by a link. This technique was chosen primarily because it offers UGVs a well-defined map, breaking down the area of spaces into structured units. In our approach, a hexagonal grid was created to represent our environment in ArcGIS Pro version 3.0.3 using the hexagon tessellation tool. The hexagon grid provides more alternate maneuvers from a centroid(node) than a rectangular grid.

Characteristics of the environment are associated with the hexagons. These characteristics include the terrain elevation, slope, and soil type - from which we determine the soil coefficient of friction in both dry and wet state. The soil map and elevation were extracted from the US web soil database (U.S. Geological Survey, 2013). The slope map was generated from ArcGIS Pro version 3.0.3 using the slope tool.

Step 2: Risk Representation

Visibility risk is a combination of the relative visibility (see equation 1) and the assumption that being seen could lead to the maximum consequence of mission failure (consequence value of 1) (Hosseinloo et al., 2012).

$$visibility\ Risk = Incident\ probabilty \times consequences \quad (1)$$

The incident probability which is known as visibility probability for this study was computed using equation 2.

The path-planning methodology includes consideration of potential visibility to an opposition. Leveraging elevation data and utilizing the viewshed function tool from

ArcGIS Pro version 3.0.3, we assess visibility for our study area. The visibility is categorized on a scale, ranging from the least to the most detectable locations, based on how frequently a node is visible to observers. The digital elevation models (DEMs) used in the visibility calculation and path planning can be derived from aerial photographs or digitizing contour maps (U.S. Geological Survey, 2013). For any DEM, the recorded elevation value of a point may differ from the actual elevation due to an error that might have been generated. These errors could stem from factors such as outdated information, data collection methods, or rapidly changing environmental conditions (Hodson, 2022). A primary source of elevation data is the U.S. Geological Survey (USGS) and Defense Mapping Agency (U.S. Geological Survey, 2013). Both provide the users a statistical measure of the level of accuracy of a map in the form of Root Mean Square Error (RMSE) (Hodson, 2022). Much of the current literature approaches visibility deterministically, overlooking the inaccuracies present in the terrain maps (Fisher, 1998). A probabilistic perspective on visibility, considering the errors within these maps offers a more thorough evaluation of visibility in navigation.

To account for these uncertainties, we produced several elevation maps (perturbed maps) based on the original map's reported RMSE using Monte Carlo simulation. Each of these maps represents a plausible scenario considering the known uncertainties in the map data. For each perturbed map generated, the visibility was computed using the viewshed function. The visibility distribution provided the data to identify the maximum visibility from observers, which was then used to calculate the visibility probability, which is adopted as the incident probability in our analysis. Maximum visibility across all perturbed

DEMs was selected to prepare for the worst-case scenario concerning visibility risk (given that consequences are equal). The incident probability was computed using equation 2.

$$Incidient\ probability = \frac{Maximum\ visibility\ observers\ across\ all\ perturbed\ DEM}{Total\ Observers} \quad (2)$$

Step 3: Quantify Travel time and Maximum Delay

Rehrer et al. (2022) developed a cross-country mobility (CCM) model that generates the speeds of vehicles in both dry and wet environment conditions. However, their approach failed to incorporate the vehicle characteristics and dynamics which can greatly affect the movement of the vehicle in an unstructured environment. In discretization of the environment with a set of nodes (centroids of hexagon grids) connected by edges, a vehicle may be able to move freely between those nodes. Graph based path search algorithms are constrained by the grid structure and this may result in paths that seem unrealistic (Khatiwada et al., 2023). A new model was developed by the vehicle dynamics research team we partnered with. This model utilizes vehicle properties and the categorization of terrain features, including the type of soil, variations in slope, and the roughness of the surface, for estimating the speed over a given area. The model assigns travel time cost based on the vehicle maneuver (M), and this was achieved by comparing the slopes of the preceding edges and successive edges to determine the turning maneuver (ß) and assign a maximum speed to the maneuver link.

Generating maneuver links

The Hyperstar algorithm relies on travel times of links to compute its optimal path. To get the travel time for each link, we adopted a function created by the vehicle dynamics team known as the speed maneuver function. The function gives the maximum speed the vehicle would take to make a particular maneuver. To generate all the possible maneuvers with their speeds associated to them, the following steps were followed.

1. We put a hexagonal grid over the study area and find the center of each hexagon. Using the Euclidean distance formula, we calculate the distance between nodes in the environment. This distance shows the minimum distance a vehicle travels between nodes in the network (Keogh et al., 2001).

$$Distance(p,q) = \sqrt{(p1-q1)^2 + (p2-q2)^2} \quad (3)$$

Where (p,q) are the nodes in the network.

2. Based on these distances, we create connections between nodes to show which links a vehicle can use to reach a specific node. Figure 3 illustrates these connections.

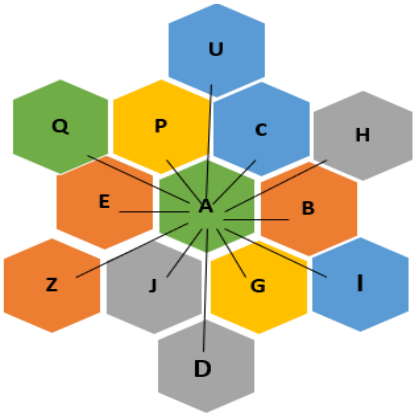

Figure 3: Node connection

3. Each connection represents a possible entry link (from node i to j) that a vehicle can use to reach a particular node. For example, in Figure 4, the entry link to node A is link E-A, and the adjacent links in the forward direction are AC, AB, AI, AD, AU, AH, and AG.

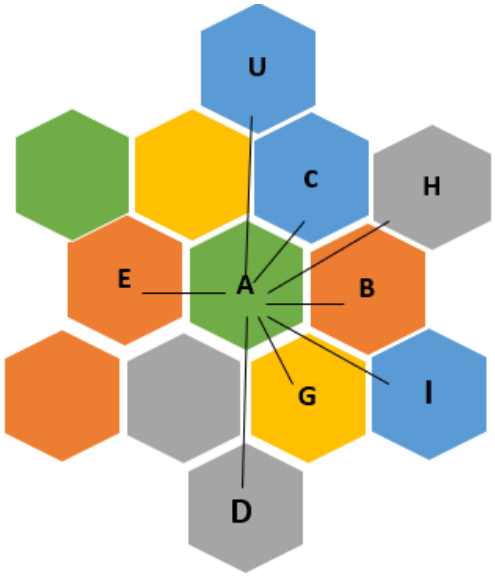

Figure 4 : Entry link E-A and its adjacent links.

4. With this concept, a link-to-link matrix was created using the link ID of the entry link and the adjacent links as shown in Table 2. This matrix ensures that there is a connection between the entry links and adjacent links in the network hence forming a maneuver between the entry link and adjacent links. Figure 5 illustrates the diagram of some possible maneuvers (m_n) represented with broken lines within a network.

Table 2: Sample from link-to-link adjacency matrix

Adjacent links →

Entry links ↓

	EA	AU	AC	AH	AB	AI	AG	AD	AZ	AQ	AP	AJ
E-A	0	1	1	1	1	1	1	1	0	0	0	0

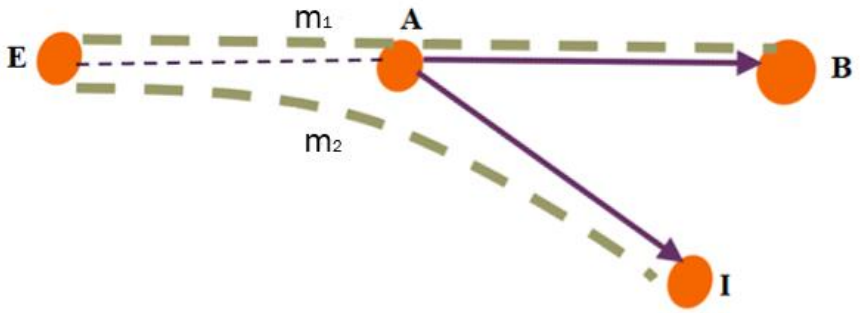

Figure 5: Some possible maneuvers

5. Travel times from the maximum speeds for each maneuver are recorded in matrix form using the link IDs. This approach reduces computation time and data storage. Each row corresponds to the link ID for an entry link in the network. Each column corresponds to the link ID for the adjacent links. The elements of the matrix represent the travel times recorded for the connection between the entry link (row) and the adjacent link (column). These values indicate the travel times from the entry link to its adjacent links. They show how long it takes a vehicle to travel from the entry link to each of the adjacent links.

6. For example, link E-A has an ID of 1 and link A-B has an ID of 2. The elements of the matrix show the travel time for maneuvering from E through A to node B. Figures 6 and 7 illustrate an example of how the travel times were compiled to be used in the Hyperstar algorithm. These data were compiled for both maximum delay time and undelay travel time. Figures 6 and 7 represent the matrices of the travel times used in the Hyperstar algorithm.

$$maximum\ delay\ time = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ - \\ - \\ n \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & - & - & m \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & - & - & a_{1m} \\ a_{21} & a_{22} & a_{23} & - & - & - \\ a_{31} & a_{32} & a_{33} & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ a_{n1} & - & - & - & - & a_{nm} \end{pmatrix} \end{array}$$

Figure 6: Data representation of maximum delay

$$Undelayed\ travel\ time = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ - \\ - \\ n \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & - & - & m \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & - & - & a_{1m} \\ a_{21} & a_{22} & a_{23} & - & - & - \\ a_{31} & a_{32} & a_{33} & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ a_{n1} & - & - & - & - & a_{nm} \end{pmatrix} \end{array}$$

Figure 7: Data representation of undelay travel time

From each perturbed DEM, the slope and maneuver angle for each maneuver link were computed based on the elevation values. With the elevations values and the soil coefficient of friction in both dry and wet soil states (U.S. Geological Survey, 2013), vehicle speeds in dry and wet conditions were computed for each maneuver link based on the direction.

Travel times were computed from the vehicle speeds that were generated. From the set of travel times generated for each maneuver, the minimum travel time is the undelayed travel time (C^{ij}). The maximum delay (D^{ij}) a vehicle can experience on a maneuver was computed as the difference between the maximum and minimum travel time from the distribution. The maximum delay, undelayed travel time and visibility risk were used as input for the Hyperstar algorithm to generate paths that minimize travel time with potential delays and visibility risk.

3.2 Concept of Modified Hyperstar Algorithm

We adapted the Hyperstar algorithm for use in identifying the optimal path for vehicles in off-road environments by adding visibility as a criterion. Each link in the discretized environment may be subjected to delay and visibility. A high link preference (F^{ij}) corresponds to low maximum delay and low visibility risk and vice versa. This approach ensures that vehicles are interested in all potentially optimal paths when adopting a minimax exposure to delay and visibility risk strategy to ensure efficient traversal and safety from detection. The algorithm is structured into two sections. In the first section, from the destination node to the origin, the optimal strategy (hyperpath) and expected total travel time (U_i) from each node $i \in V$ to the destination node are computed (Spiess & Florian, 1989). In the second part from the origin to the destination, the proportion of link preference for each potentially used link (based on visibility risk and maximum delay) is computed and assigned to each optimal path.

Notation

The following notation (L,Q,h,y,U,s,r,C,H,D) are consistent with Bell et al. (2012) whilst (x,R,P,v) are not:

l *Set of links*

l^+ *Set of outgoing links*

l^- *Set of incoming links*

V *Set of nodes*

C^{ij} *Link travel time on link between node i and node j*

D^{ij} *Maximum delay on link*

R^{ij} *Visibility risk on link*

P^{ij} *Proportion of link preference*

Q_i *Exposure at node i arising from maximum delay at node i*

h_i *Heuristic travel cost from node i to origin.*

y_i *Chance of using node i*

U_i *Travel cost from node i to destination*

f_i *Sum of link preference for all outgoing links of node i*

x_i *Exposure at node i arising from the probability that node i is visible.*

H *Set of optial links known as the Hyperpath*

F^{ij} *Link pref ernce based on maximum delay and visibility risk*

s *Start node*

r *End node*

Modified Hyperstar Algorithm

0 *Initialization* $U_i \leftarrow \infty$, $i \in V - \{s\}$, $U_s \leftarrow 0$ (*Bell et al.*, 2012) .

$f_i \leftarrow 0,\ i \in V;$

$y_i \leftarrow 0,\ i \in V - \{r\}, y_r \leftarrow 1$

$H \leftarrow \varnothing$

1 *Select a link* l_1 *with minimum* $h_i + U_j + C^{ij}$;

$l \leftarrow l - \{l_1\}$

2 *Update node i*

if $U_i \geq U_j + C^{ij}$ *then*

if $U_i = \infty$ *and* $f_i = 0$ *then* $\beta \leftarrow 1$ *else* $\beta \leftarrow f_i U_i$ (*Bell et al.*, 2012) .

$$U_i \leftarrow \leftarrow \frac{\left(\beta + \left(\frac{1}{D^{ij}} \times (1 - R^{ij})\right)\left(U_j + C^{ij}\right)\right)}{f_i + \frac{1}{D^{ij}} \times (1 - R^{ij})}$$

$$f_i \leftarrow f_i + \left(\frac{1}{D^{ij}} \times (1 - R^{ij})\right) \text{ and}$$

$H \leftarrow H + \{l_1\};$

3 *Proportion of link preference* *if* $l = \varnothing$ *or* $h_i + U_j + C^{ij} > U_r$ *go to step* 3 *else go to step* 1

for every link l_1 *in decreasing order of* $h_i + U_j + C^{ij}$ (*Bell et al.*, 2012) ,

$$\text{if } l_1 \in H \text{ then } P^{ij} \leftarrow \left(\frac{\frac{1}{D^{ij}} \times (1 - R^{ij})}{f_i}\right) y_i \quad \text{and } y_j \leftarrow y_j + P^{ij} \text{ else } P^{ij} \leftarrow 0$$

Part 1

Step A

a) Node labels U_i representing the travel cost to reach the destination are initially assigned a value of infinity except the destination node, which is set to zero (Bell et al., 2012).

b) The variable f_i , $i \in V$, representing the combined link preference of all selected links at node i is initially set to zero (Bell et al., 2012).

The link preference variable (F^{ij}) , which is the product of our two criteria, is determined for all links in the network using equation (9).

$$F^{ij} = \frac{1}{D^{ij}} \times (1 - R^{ij}),\ D^{ij} > 0 \ \ and \ \ 0 \leq R^{ij} \leq 1 \quad (9)$$

This principle indicates that routes with longer delays are less favored for use within the network, and likewise, those with higher visibility risk are less favored. Evaluating links based on these two criteria, maximum delay and visibility, ensures that the most timely and discreet routes are prioritized.

c) Node probability labelled as y_i, is initialized as 0 for all the nodes in the network except the start node which is initialized as 1.

d) 'L' denotes the set of links that has not been evaluated and H denotes the set of links in the hyperpath.

Step B

Each link in the network is sorted in ascending order of ($hi+Uj+C^{ij}$) (Bell et al., 2012; Hosseinloo et al., 2012).

The variable 'C^{ij}' denotes the minimum travel time (undelayed travel time) from node i to node j. This was obtained from our travel time distribution that was generated.

a) The travel cost from node j to the destination node in a network, accounting for maximum delay and visibility risk, is denoted by 'Uj'.

b) 'hi' denotes the heuristic travel cost from a node i to the start node. This was computed from the Euclidean distance between the current node and the end node and the ideal speed (vehicle's top speed without considering the terrain) of the vehicle.

Step C

a) Select a link with the minimum ($hi+Uj+C^{ij}$) (Bell et al., 2012; Hosseinloo et al., 2012).

b) Check if the travel cost ($Uj +C^{ij}$) from node j of a link to the destination and the minimum travel time for the selected link, is smaller than the current travel time Ui. If this condition is true, then 'Ui' and 'fi' must be updated based on both maximum delay and visibility risk. This means traversing on that selected link 'will potentially allow a faster arrival at the destination than any previously considered link through node i. Therefore, the selected link is regarded as a better link than all the links analyzed so far based on time. Ui is improved to a new value (smaller value) to improve path to the destination.

c) This step is repeated for all the links in the network or when a link sorting value ($hi+Uj+C^{ij}$) of a selected link is greater than $U_{(start\ node).}$ Then this part of the algorithm will terminate (Bell et al., 2012; Hosseinloo et al., 2012)

Part 2

a) This part computes the proportion of link preference for a link to be used by the vehicle. This process is done by sorting all the links in the optimal strategy (H) in ascending order of ($hi+Uj+C^{ij}$). Once sorted, the algorithm computes the proportion of link preference (P^{ij}).

b) After computing the proportion of link preference and assigning it to each link, the vehicle is expected to follow the link with the highest probability.. However, in

unexpected delays and disruptions, the vehicle has several alternate links to choose from to get to the destination.

CHAPTER 4

RESULTS AND DISCUSSION

The preceding chapter presented the modified Hyperstar algorithm and overall path planning approach for offroad contexts. In this chapter, the focus shifts from theoretical exploration to the application. The new modified Hyperstar algorithm was tested on numerical examples and terrain maps to illustrate how the algorithm operates under potential delays and visibility risk.

3.1 Numeric Examples

To showcase the proposed method, the approach underwent testing on a small-scale network as well as on a detailed terrain map. Numerical examples are presented to illustrate the trade-offs faced by the vehicle. Table 3 shows the cases that were considered for the new modified Hyperstar algorithm.

Table 3: Cases considered for modified Hyperstar algorithm

Case	Test	Reason
1	Evaluate how different levels of visibility risk affect the preferred paths when the maximum delay for each path remains a constant value for each link in a network.	To verify that the link preference formula presented in equation (10) produces reasonable outcomes such that the lower the visibility risk, the higher link preference.
2	Evaluate how different levels of maximum delay time affect the preferred paths when the visibility risk remains a constant value for each link in a network.	To reproduce Bell et al.'s (2012) results to make sure that the link preference equation in (12) yields a reasonable result when visibility risk is held constant.

3	Compare scenarios of low medium, medium and high visibility risk and scenarios of low, medium and high delay to determine whether the number of optimal paths changes	This case examines the algorithm's range in providing multiple route choices under different combinations from the most favorable to the most adverse environment.
4	Test the relationship between the link preference (maximum delay and visibility risk) and the actual proportion of link preference that each link receives in specially designed networks. In essence, it seeks to clarify if the preferred links are selected based on their maximum delay and visibility risk parameters.	To understand the limitations in the new modified Hyperstar algorithm, providing insights into areas where its performance may fall short or where it may not meet specific criteria.
5	Test on terrain maps (Idaho, Nevada)	To assess how well the modified Hyperstar algorithm would perform in offroad scenarios.
6	To evaluate the new modified Hyperstar algorithm in a bidirectional scenario	To search for limitations in the algorithm

3.1.1 Case 1

If the maximum delay (D) is constant across all links, then the link preference in the modified Hyperstar algorithm will be a function of visibility risk (R^{ij}) for each link. The link preference initially presented in equation 9 becomes equation 10 below:

$$F^{ij} = k(1 - R^{ij}) \tag{10}$$

Where k is a constant that represents $\frac{1}{D}$.

When updating the cost (U_i) at node *i*, the constant part ($\frac{1}{D}$) of the link preference (F^{ij}) will not influence the order in which nodes are updated since it is the same for every link. Only the visibility risk (R^{ij}) and its relation to the link preference will impact the node updating process, as it affects the value of $\left(U_i \right)$ as shown equation 11. This means that the decision to select a particular path within the network is influenced primarily by how likely it is to be visible, rather than potential maximum delay.

$$U_i \leftarrow \frac{\left(\beta + (1 - R^{ij}) \left(U_j + C^{ij} \right) \right)}{f_i + (1 - R^{ij})} \tag{11}$$

The calculation will prioritize links with lower visibility risks, as the factor $(1 - R^{ij})$ will be higher for links with lower visibility risks, making (F^{ij}) larger and thus the link more preferred. A small network shown in Figure 8 shows the network used by Bell et al. (2012) as a numeric example in generating multiple paths in minimizing the consequences of hazardous material using the Hyperstar algorithm. Every link has three parameters: undelayed travel time (C), maximum delay time (D) and visibility risk (R) as shown in Table 4. The average speed on all links is assumed to be 50 km/h, which influences the undelayed travel time. (C) for each link Bell et al. (2012). The start and destination node for this network are 1 and 4, respectively.

Table 4: Link Parameters

Links	C	R	D
1-2	8	0.5	1
2-4	16	0.8	1
1-3	24	0.7	1
2-3	10	0.6	1
3-4	16	0.2	1

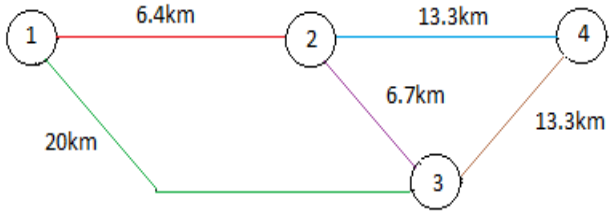

Figure 8 Example network

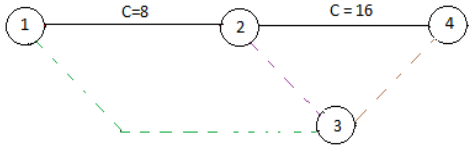

Figure 9: Optimal route based on undelay travel time

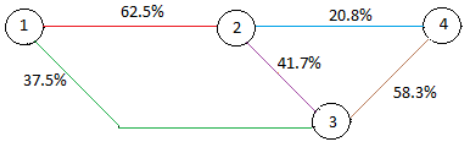

Figure 10: Optimal route based on proportion of link preference

Figures 9 and 10 display optimal path results from node 1 to 4 for a network diagram shown in Figure 8 considering undelayed travel time and link preference, respectively. In Figure 9, the direct path (1-2-4) is optimal when only considering the fastest route, as typically calculated by standard algorithms like Dijkstra and A*. However, Figure 10 reveals a different optimal path (1-2-3-4) when applying equation (10) which factors in a constant maximum delay alongside visibility risk. This equation prioritizes links with a higher preference ratio, indicating a lower visibility risk. At the first decision point, node 1, vehicles face a choice between links (1,2) and (1,3). The choice falls on link (1,2) due to its higher link preference ratio. At node 2, the undelayed travel would have normally directed vehicles to use link (2,4), but with the visibility risk, the route shifts the preference

to link (2,3). Consequently, the route progresses to node 3 and finally to node 4, adopting the path (1-2-3-4) based on a balance of travel time and low visibility.

The chosen route (1-2-3-4) illustrates selecting the paths with a higher link preference that offer reduced visibility risk, showcasing a balance between travel time and visibility risk. Although path 1-2-4 might be quicker, the alternative 1-2-3-4 is chosen to better conceal the vehicle's movement.

3.1.2 Case 2

This case evaluates the maximum delay across network links under a specific assumption: the visibility risk (R) for all links is constant. If the visibility risk (R) is constant across all links, then the link preference in the modified Hyperstar algorithm will be based on the maximum delay (D^{ij}) for each link. The link preference initially presented in equation 9 becomes equation 12:

$$F^{ij} = \frac{1}{D^{ij}} \times k \tag{12}$$

Where k is a constant that represents $(1 - R)$.

When updating the cost (Ui) at node i, the constant part (R) of the link preference will not influence the order in which nodes are updated since it is the same for every link. Only the maximum delay and its relation to the link preference will impact the node updating process, as it affects the value of (U_i) as shown in equation 13. This means that the decision to select a particular path is influenced primarily by maximum delay.

$$U_i \leftarrow \frac{\left(\beta + \frac{1}{D^{ij}}\left(U_j + C^{ij}\right)\right)}{f_i + \frac{1}{D^{ij}}} \tag{13}$$

In determining the proportion of link preference, the calculation will prioritize links with shorter maximum delays, as the factor $\left(\frac{1}{D^{ij}}\right)$ will be higher for links with lower maximum delays, making the link preference (F^{ij}) larger and thus the link more preferred.

The network diagram in Figure 8 was used for case 2 with varying maximum delay (D) and a constant visibility risk (R). The link parameters are provided in Table 5 with the speed of the vehicle as 30km/h. Both the undelayed travel time (C) and maximum delay time (D) were the same as used by Hosseinloo et al. (2012).

Table 5: Network parameters for case 2

Links	C	R	D
1-2	13	0	6
2-4	16	0	10
1-3	19	0	4.5
2-3	15	0	3
3-4	24	0	6

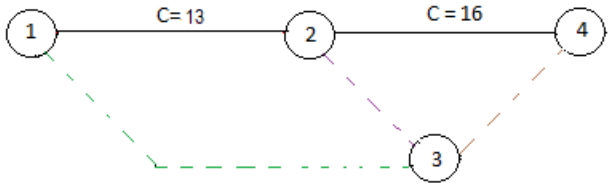

Figure 11: Optimal route based on undelay travel time.

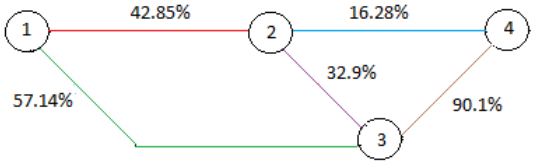

Figure 12: Proportion of link preference for varying maximum delay

From Figure 11, the optimal path from the network diagram in Figure 8 based on undelayed travel time would be (1-2-4). With the visibility risk held constant, the optimal path is now based on maximum delay and undelayed travel time only. The higher the percentage of link preference, the more favorable the link is because it has a lower maximum delay. From Figure 12, at node 1 the vehicle has a choice of links (1,2) and (1,3) to move to the next node and to get to the destination node. Route (1-2-3-4) is the route that the vehicle is inclined to choose because it has a high percentage of link preference in terms of maximum delay, similar to what Bell et al. (2012) produced using an 8 node by 8 node grid network.

3.1.3 Case 3

This evaluation utilized a 5 by 5 node square grid network as shown in figure 13 with undelayed travel times (C) as 1+M, where M is a random number as used by Bell et al. (2012). The maximum delay time (D) and visibility risk (R) were random numbers generated in the range of 0 – 1 inclusive (Bell et al., 2012). The undelayed travel time and the link parameters such as maximum delay and visibility risk for all the three cases are shown in Table A in the appendix. Three cases were considered, and the undelayed travel time is the same for all three scenarios. The start and destination nodes were chosen as 1 and 20, respectively. The maximum delay and visibility risk for all the scenarios are shown in appendix A. For scenario A, high maximum delay and high visibility risk are assigned to all links, simulating conditions with maximum delay time and visibility risk. Scenario B, with a medium maximum delay and visibility risk (R), represents a moderate level of operational challenge. The final scenario (scenario C) imposed low maximum delay and visibility risk on all network links.

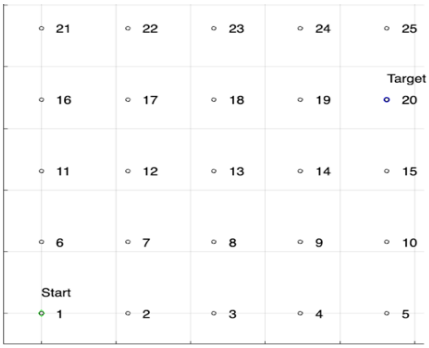

Figure 13: Square grid structure

Scenario A

In this scenario, each link has two factors: high maximum delays and high visibility risk. Specifically, a high maximum delay and visibility risk were both assigned a random number "B" and “K,” respectively, as shown in Table A in the appendix.

Figure 14: Optimal links for scenario A

This resulted in the generation of a hyperpath with 27 links as shown in figure 14. The high number of optimal links enhances the vehicle’s ability to adapt to unexpected changes. As shown in figure 14, each link has a proportion of link preference associated with it. From its starting position at node 1, the vehicle assesses two potential links, each characterized

by a unique proportion of link preference. In moving towards its destination node, the vehicle selects the link that has the highest proportion of link preference.

Scenario B

In this case, each link has medium maximum delays and visibility risks. The random numbers "B" and “K” representing the maximum delay and visibility risk were scaled down by multiplying them by a factor of 0.5 as shown in the appendix. This adjustment means that while maximum delays and visibility risk still exist, they are less severe compared to scenario A.

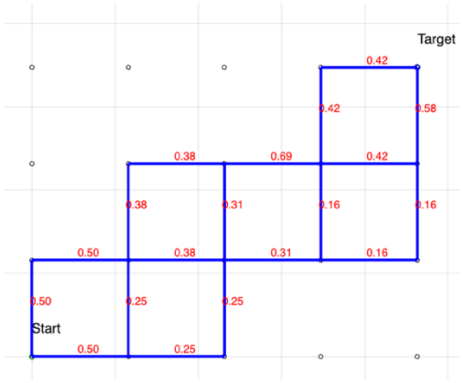

Figure 15: Optimal routes for scenario B

As observed from Figure 15, a hyperpath with 19 optimal links was generated. In Figure 15, each link is annotated with a numerical value representing its proportion of link preference. Links with higher proportions are deemed more favorable, directly influencing the vehicle's route selection as it progresses from the starting node toward its destination.

Scenario C

In this network, each link is marked by low maximum delays and visibility risks. However, to reflect their low severity, the random numbers "B" and “K” representing the maximum delay and visibility risk were scaled down by multiplying by a factor of 0.3 as shown in

the appendix. Consequently, each link's maximum delay and visibility risk are now less than before, maintaining their low levels.

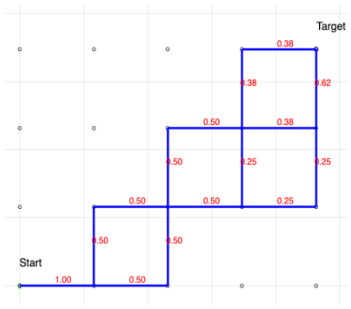

Figure 16: Optimal routes for scenario C

As observed from Figure 16, 15 preferred links were generated. This outcome illustrates that as maximum delay and visibility risk decrease across the network, the total number of preferred links tends to become less. This reduction occurs because the algorithm prioritizes finding the most direct and quick paths from one point to another, with little maximum delay and risk of detection. With maximum delays and risks being low across all routes, these factors become less critical in differentiating between paths. Consequently, the algorithm can more easily identify the fastest routes, as there is no significant delay or risk to avoid. This reduces the need for alternative routes since the primary ones are already optimal in terms of speed. In such environments, the Hyperstar algorithm functions similarly to traditional shortest-path algorithms, focusing mainly on minimizing travel time. Hence, in scenarios where delays and visibility risks are minimal, the algorithm simplifies its selection process to a smaller set of preferred routes.

Table 6 below shows the number of optimal paths generated for the three scenarios (A,B,C) considered in Case 3. As the maximum delay and visibility risk increase, so too does the number of optimal links within the hyperpath.

Table 6: Number of optimal links produced based on the level of criteria

Scenarios	Level of Criteria	Number of optimal links
A	High	25
B	medium	19
C	Low	15

3.1.4 Case 4

This analysis was considered to determine the relationship between the link preference, determined by parameters such as maximum delay and visibility risk, and the actual proportion of link preference. The case was considered for both visibility risk and maximum delay by doubling some of the links with respect to the criteria as shown in Figures 17 and 18. The link preference formula from equation (9), was used considering visibility risk with a fixed maximum delay time set to 1 for the first scenario, and for the second scenario, the visibility risk was constant and the maximum delay for some links were double. The question was whether this would lead to a proportional increase in the preference metric. Travel times were computed as 1+M, adhering to Bell et al.'s (2012) approach, with 'M' representing a random variable from 0 to 1 a shown in Table 7. The start node and end node for the network were assigned to be 1 and 4 respectively.

Table 7: link parameters for Case 4

Links	C(mins)
1-2	1.42
2-3	1.87
3-4	1.86
1-8	1.10
1-7	1.09
7-5	1.95
5-4	1.14
6-5	1.96
1-6	1.46
8-3	1.78

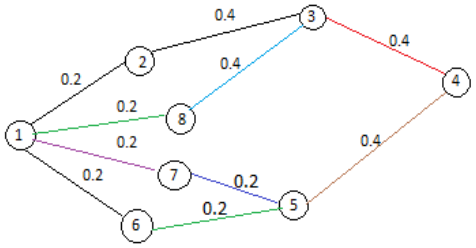

Figure 17: Network with visibility risk

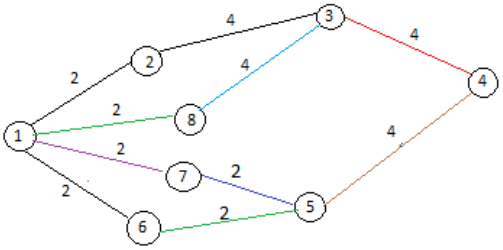

Figure 18: Network with maximum delay

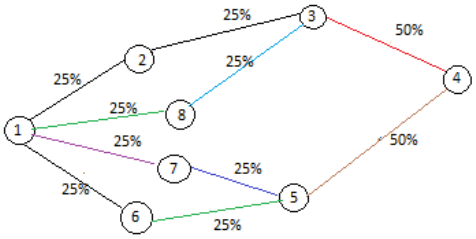

Figure 19: Link preference by visibility risk.

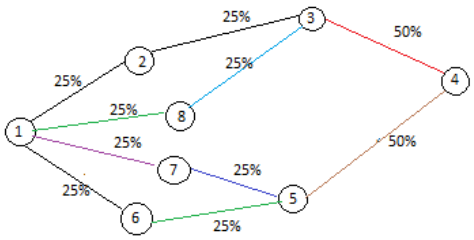

Figure 20: Link preference by maximum delay

In the analysis, both scenarios showed an unexpected outcome: the proportion of link preference remained the same for all links, as depicted in Figure 19 and 20. For instance, Link (2-3) was anticipated to have half the link preference of Link (7-5), yet their preferences ended up being identical. This outcome explains the limitation of the link preference formula as shown in equation (9) - that it is not just influenced by visibility risk or maximum delay. One factor is the number of available link options at each node for continuing the path. When a node (let's call it "i") has multiple outgoing links, the link preference for incoming links to "i" gets divided among the outgoing links based on their preferences. However, if node i has only a single outgoing link, the entire link preference from the incoming links transfers directly to this lone outgoing link.

3.1.5 Case 5

Case 5 concentrates on assessing the newly modified Hyperstar algorithm when bidirectional links are present. Figure 21 presents a bidirectional network diagram from Figure 13. The blue lines indicate the forward direction from start to target while the red lines indicate the reverse direction from finish to start. Table 8 details the link parameters used in Case 5, providing a basis for this bidirectional testing. Start and end nodes for this case were 1 and 20, respectively.

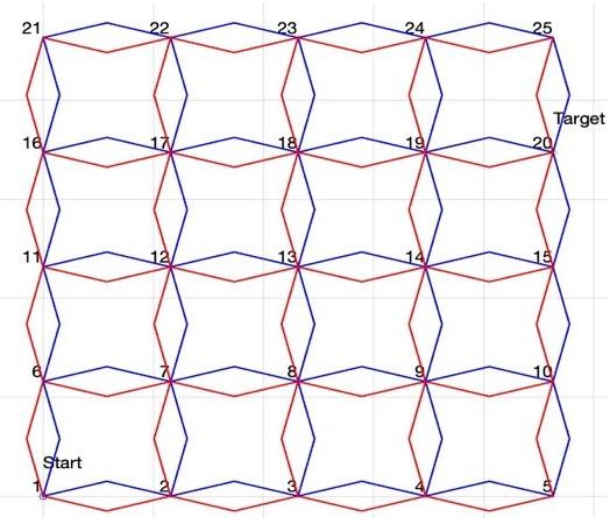

Figure 22: Example network for Bi- direction Network

Table 8: Link parameters for Case 5

Links		C(mins)	R	D(mins)
1	2	1.94	0.28	0.51
1	6	1.01	0.93	0.32
2	3	1.53	0.03	0.26
2	7	1.58	0.75	0.86
3	4	1.70	0.09	0.39
3	8	1.17	0.28	0.11
4	5	1.13	0.53	0.56
4	9	1.31	0.21	0.54
6	7	1.32	0.52	0.56
6	11	1.10	0.27	0.41
7	8	1.67	0.68	0.63

7	12	1.16	0.20	0.92
8	9	1.48	0.95	0.52
8	13	1.91	0.75	0.05
9	10	1.72	0.43	0.19
9	14	1.08	0.79	0.33
11	12	1.03	0.67	0.47
11	16	1.82	0.37	0.16
12	13	1.04	0.95	0.12
12	17	1.20	0.57	0.16
13	14	1.01	0.42	0.91
13	18	1.55	0.15	0.25
14	15	1.02	0.37	0.84
14	19	1.60	0.58	0.74
15	20	1.09	0.50	0.06
16	17	1.66	0.66	0.50
16	21	1.76	0.74	0.88
17	22	1.81	0.90	0.13
17	18	1.72	0.26	0.41
18	23	1.51	0.65	0.36
18	19	1.83	0.52	0.96
19	20	1.70	0.67	0.36
19	24	1.26	0.43	0.56
21	22	1.15	0.81	0.52
22	23	1.81	0.31	0.81
23	24	1.67	0.18	0.68
24	25	1.70	0.30	0.10

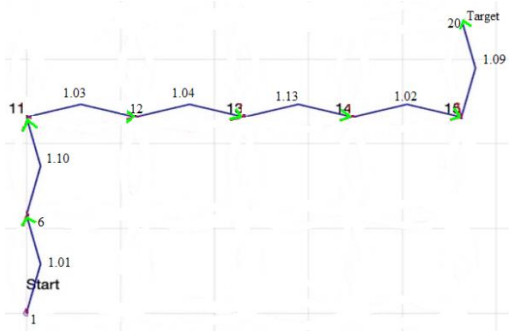

Figure 22: Optimal routes for Bi-direction network based on undelayed travel time

Figure 22 illustrates the results of generating the optimal path using a bidirectional approach based on undelayed travel time. The optimal path, selected solely based on

undelayed travel time, consists of 7 optimal links. Due to the bidirectional nature of the network, some links have lower costs in one direction than in the other. Starting from node 1, the vehicle uses link 1-6 in the forward direction because it has the lowest cost. The undelayed travel times are indicated on top of each link in Figure 22**.**

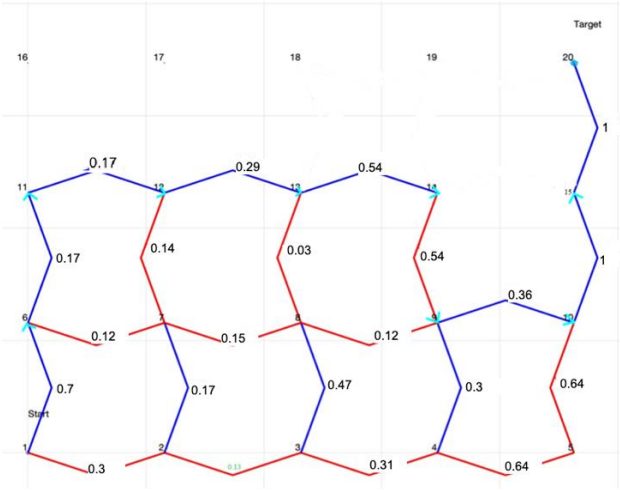

Figure 23: Optimal routes for Bi-direction network

However, when additional criteria such as visibility risk, maximum delay, and undelayed travel time are considered, the high cost of link 14-15 results in a lower link preference. Consequently, the vehicle avoids this link and takes link 14-9 instead. This new route is depicted in Figure 23 indicated by arrows, along with the proportion of link preference. This detour adds extra links, making the route longer.

Table 9: list of links and link cost

Forward direction links		Reverse Direction links	
Links	Link cost	Links	Link cost
1-6	20.23	6-7	21.32
6-11	21.03	12-7	22.24
11-12	21.67	13-8	23.24
12-13	23.01	14-9	22.24
13-14	20.13	10-5	20.63
14-15	23.57	2-1	21.23

15-20	21.21	3-2	20.21
10-15	20.20	4-3	19.46
2-7	20.10	5-4	20.32
3-8	19.34	19-14	22.23
4-9	20.20	20-19	23.42
9-10	20.12	19-18	20.67
13-18	20.10	-	-

3.2 Test on Terrain Map

The approach was evaluated using two distinct terrain maps: an off-road environment in Nevada, specifically near the Churchill County Area, and another in Idaho, close to Foxtail Lake and Elmore County Area. Idaho and Nevada maps are of areas approximately 309,341 square meters and 994,798 square meters, respectively. The elevation for the Idaho map (Figure 24) ranges from 820 meters to 900 meters. The elevation for the Nevada map (Figure 25) ranges from 1194 meters to 1207 meters.

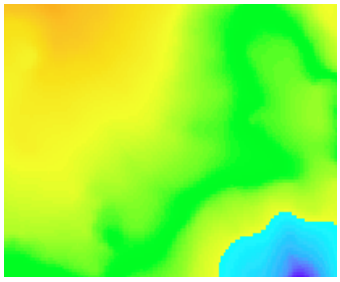

Figure 24: Elevation map for Idaho

Figure 25: Elevation map for Nevada

The areas were segmented into hexagonal grids, with each hexagon covering an area of 779.4 square meters and centroids distanced 30 meters apart as shown in figure 32. The High Mobility Multipurpose Wheeled Vehicle (HMMWV) was selected for the test vehicle. The vehicle has a base width of 3.80 meters, a track width of 1.819 meters, a

maximum road wheel angle of 36 degrees, a gradeability of 60%, and a top speed of 50 km/h (Hosseinloo et al., 2012).

Using a Monte Carlo simulation approach, multiple maps were generated based on the elevation data provided by the maps of Idaho and Nevada, with Root Mean Square Error (RMSE) values of 0.3m and 0.35m (U.S. Geological Survey, 2013). These simulations aided in computing risk visibility.

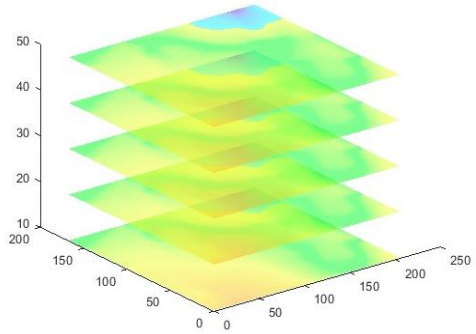

Figure 26: Perturbed Elevation for Idaho

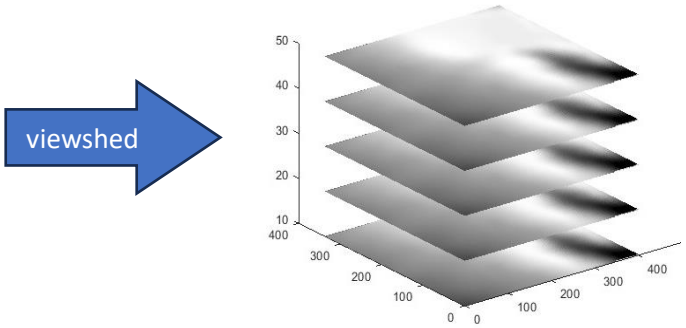

Figure 27:Visibility results for Idaho

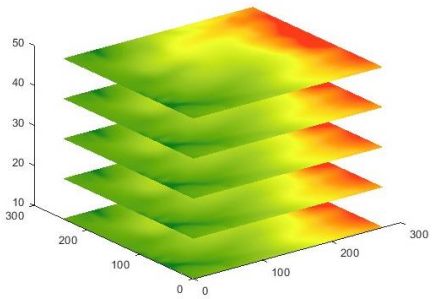

Figure 28: Perturbed elevation for Nevada

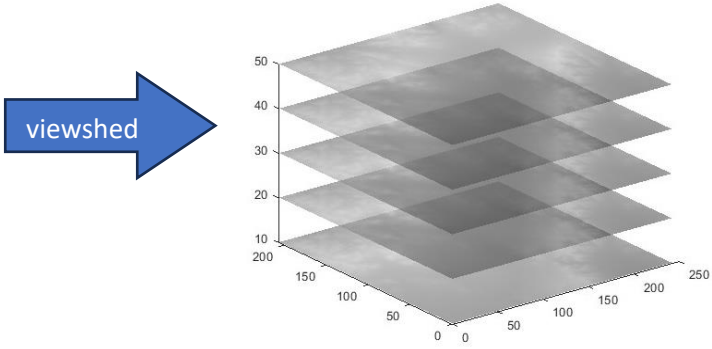

Figure 29: Visibility risk results for Nevada

In the visibility analysis conducted for both Nevada and Idaho maps, it was determined that the visibility risk stabilizes after the generation of a certain number of elevation maps.

Each graph in Figure 30 and 31 illustrates some evolutions of maximum visibility risk at sampled nodes. This value stabilizes approximately between 35 to 50 simulations. Forty DEMs were then created to generate the visibility risk for both terrain maps.

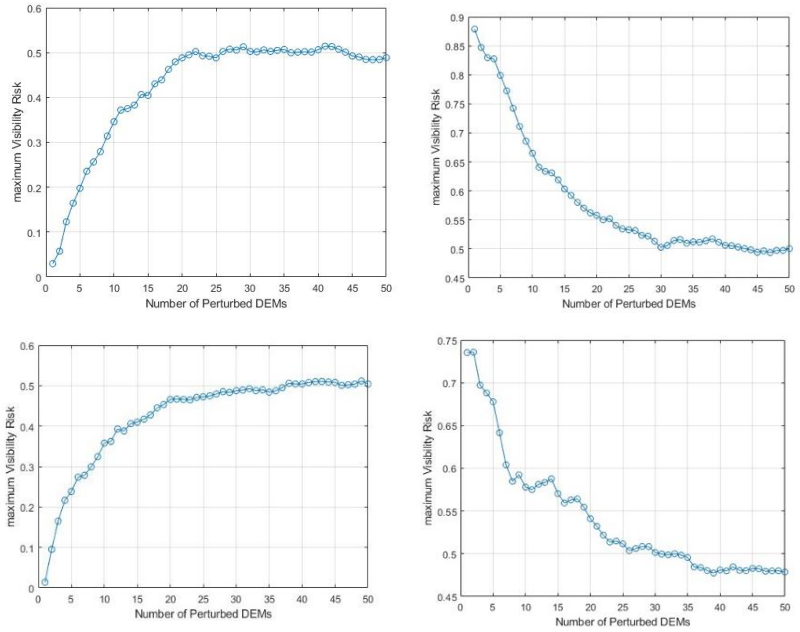

Figure 30: A plot of visibility risk against number of perturbed DEMs for Idaho map

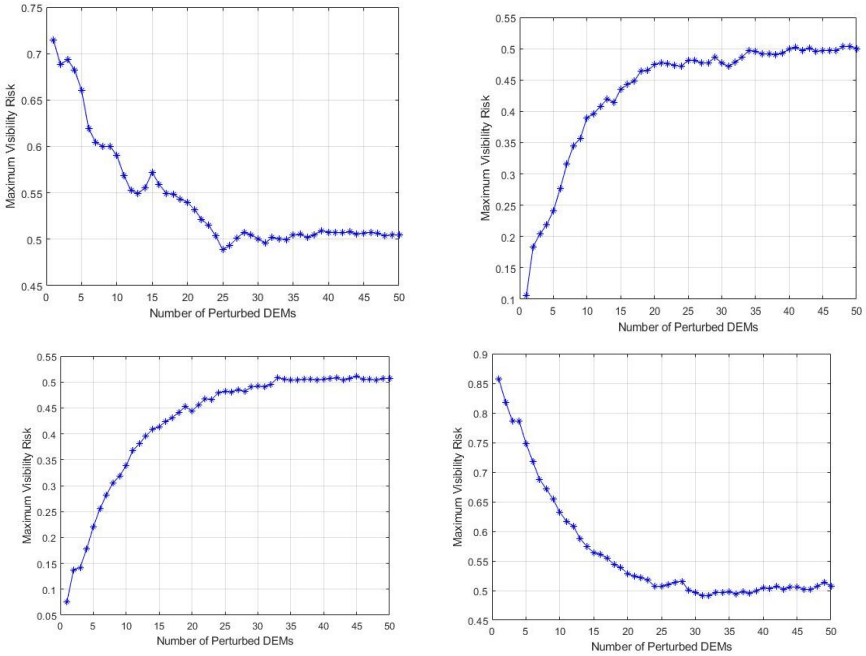

Figure 31: A plot of visibility risk against number of perturbed DEMs for Nevada map

Figure 32: Environment representation

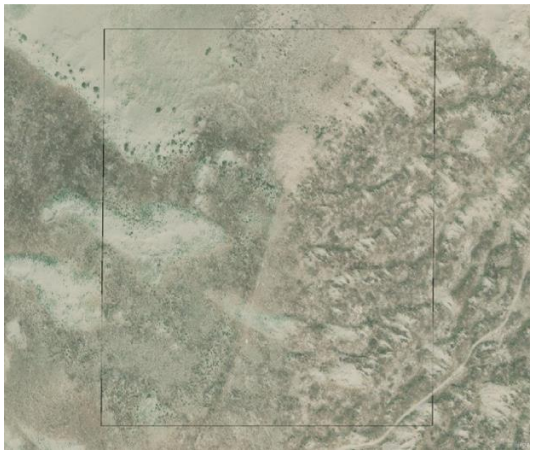

Figure 33: Terrain map: Idaho

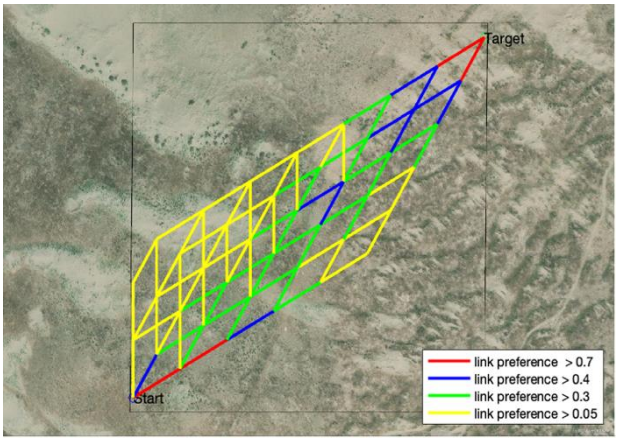

Figure 34: Optimal path with link preference for Idaho

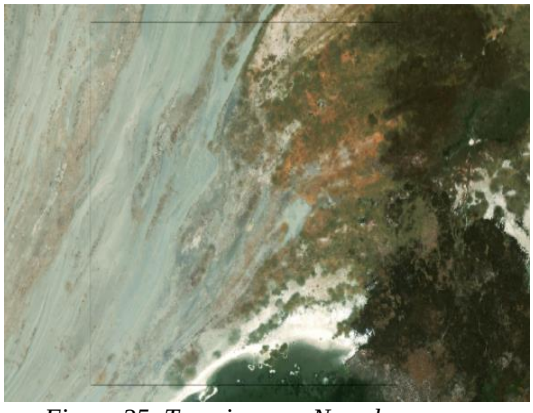

Figure 35: Terrain map: Nevada

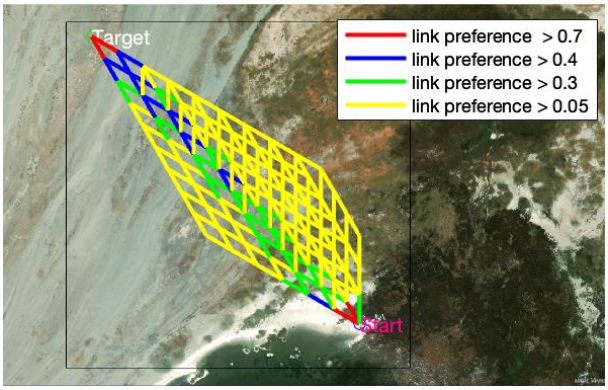

Figure 36: Optimal path with link preference for Nevada

Figures 34 and 36 utilize color-coded links to differentiate between those with high link preference and those with low link preference. Selecting routes with high link preference (marked in green, red, and blue) is strategically important to minimize delays and avoid detection. These preferred routes are designed based on the terrain, ease of navigation, and visibility. However, in situations such as poor soil quality or any other factors that could potentially delay or increase the risk of detection, the vehicle has alternate routes to follow to get to its destination. In such scenarios, alternative routes marked in yellow are identified. This approach ensures that the vehicle has flexible routing options, enhancing both the effectiveness and safety of operations in challenging off-road environments.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSION

This thesis presented a study on path planning for unmanned ground vehicles (UGVs). The modification of Hyperstar algorithm (Bell et al., 2012; Hosseinloo et al., 2012) addresses the challenge of path planning for offroad conditions, which is helpful in military applications and other applications with incomplete information (e.g., during and after disasters). By considering the errors in maps, the proposed methodology allows for more reliable and safe planning of UGVs in an unstructured environment where traditional commercial navigation systems are inadequate. By applying the model developed by the vehicle dynamics team, several travel times were generated based on soil conditions in both wet and dry states and a variety of perturbed DEMs. Both maximum delay and undelayed travel times were estimated from these travel times which were assigned to each maneuver link in the network. A link preference function (used in the modified Hyperstar algorithm) was developed to incorporate both visibility risk and maximum delay for each maneuver.

The modified Hyperstar algorithm was tested on 6 cases with different conditions including terrain maps. In Case 1, the test focused on the impact of varying visibility risks while keeping the maximum delay constant across all network links. It was observed that links with lower visibility risks were prioritized. Case 2 analyzed how changes in the maximum delay affected path preferences when visibility risks remained constant. This case confirmed that links with shorter maximum delays were preferred when visibility risk was not a variable. Case 3 compared low, medium, and high visibility risk scenarios with

corresponding delay levels to check the number of optimal paths. This case illustrated the algorithm's ability to offer multiple route choices from the most to the least favorable conditions. It was found that as maximum delay and visibility risk decreased, the number of preferred links also reduced, showing the algorithm's tendency to simplify its route selection to the fastest paths when risks were minimal. Case 4 was to determine the relationship between link preference (defined by maximum delay and visibility risk) and the proportion of link preference each link receives. Surprisingly, despite variations in these parameters, the link preference proportion remained constant in special network conditions, showing a limitation of the modified Hyperstar algorithm. This highlighted that the number of available link options at a node and the distribution of preferences across these links were significant factors. Case 5 evaluated the bidirectional capabilities of the modified Hyperstar algorithm, showing its adaptability to conditions requiring movement not necessarily directly toward the target node. The test involved links with different parameters for each direction, providing a basis for the ability of the algorithm to handle such scenarios. Case 6 tested the overall approach on terrain maps in Idaho and Nevada. The analysis demonstrated the algorithm's ability to adapt to varying terrain and visibility risks.

For the test on terrain maps, multiple realizations of the environment using Monte Carlo simulations allowed computation of multiple visibility risks, which stabilized after 35-50 simulations. These maps also allowed generation of a variety of travel times. In conclusion, the enhanced Hyperstar algorithm showed competence in handling different risk levels and making strategic route choices, although some limitations were noted.

The visibility risk was considered by taking the maximum number of observers who could see the point across all the DEMs divided by the total observers. By moving beyond deterministic models, this thesis addresses the inherent uncertainties within environment modeling. Despite these approaches, there remain limitations that need to be acknowledged. For instance, the algorithm's performance is dependent on the accuracy of input data, such as terrain and visibility information. While the use of Monte Carlo simulation offers a framework for accounting for visibility risk, the quality of the output paths is only as good as the data and models used. Also, the method employed to allocate link preference proportions for vehicle navigation might not accurately reflect the actual maximum delay and visibility risk for some links. This may result in an over or underestimation of some paths' suitability.

5.2 FUTURE WORK

Future research could extend this work in several directions. Firstly, exploring the integration of real-time data such as congestion levels, average speeds, and weather data like rain, smoke, or fog into the Hyperstar algorithm could enhance its adaptability, allowing for more dynamic path planning as conditions change. Moreover, the link preference equation, which combines the two criteria, could be improved by incorporating a weighting method. This would dynamically weigh the two criteria, ensuring a proper balance between them and accurately reflecting the proportion of link preference. With advancements in sensor technology and data analytics, there is a scope for UGVs to learn from past missions to navigate through complex environments more efficiently.

REFERENCES

Adzhar, N., Yusof, Y., & Ahmad, M. A. (2020). A review on autonomous mobile robot path planning algorithms. *Advances in Science, Technology and Engineering Systems*, *5*(3), 236–240. https://doi.org/10.25046/aj050330

Alam, T., Qamar, S., Dixit, A., & Benaida, M. (2020). Genetic algorithm: Reviews, implementations, and applications. *arXiv preprint arXiv:2007.12673*.

Albizu, A., Fang, R., Indahlastari, A., O'Shea, A., Stolte, S. E., See, K. B., Boutzoukas, E. M., Kraft, J. N., Nissim, N. R., & Woods, A. J. (2020). Machine learning and individual variability in electric field characteristics predict tDCS treatment response. *Brain Stimulation*, *13*(6), 1753–1764. https://doi.org/10.1016/j.brs.2020.10.001.

Agarwal, S., & Akella, S. (2022). Area Coverage with Multiple Capacity-Constrained Robots. *IEEE Robotics and Automation Letters*, *7*(2), 3734–3741. https://doi.org/10.1109/LRA.2022.3146952

Bell. G.H. (2009). Hyperstar: A multi-path Astar algorithm for risk averse vehicle navigation. Transportation Research Part B 43(1), 97–107.

Bell, M. G. H., Trozzi, V., Hosseinloo, S. H., Gentile, G., & Fonzone, A. (2012). Time-dependent Hyperstar algorithm for robust vehicle navigation. *Transportation Research Part A: Policy and Practice*, *46*(5), 790–800.

Carr, J. (2014). An introduction to genetic algorithms. *Senior Project*, *1*(40), 7.

Chabini, I., & Lan, S. (2002). Adaptations of the A* Algorithm for the Computation of Fastest Paths in Deterministic Discrete-Time Dynamic Networks. In IEEE Transactions On Intelligent Transportation Systems (Vol. 3, Issue 1).

Chen, B. Y., Chen, X. W., Chen, H. P., & Lam, W. H. K. (2021). A fast algorithm for finding K shortest paths using generalized spur path reuse technique. *Transactions in GIS*, *25*(1), 516–533. https://doi.org/10.1111/tgis.12699

Chen, Z., Zhang, Y., Zhang, Y., Nie, Y., Tang, J., & Zhu, S. (2019). A Hybrid Path Planning Algorithm for Unmanned Surface Vehicles in Complex Environment with Dynamic Obstacles. *IEEE Access*, *7*, 126439–126449. https://doi.org/10.1109/ACCESS.2019.2936689

Dinelli, C., Racette, J., Escarcega, M., Lotero, S., Gordon, J., Montoya, J., Dunaway, C., Androulakis, V., Khaniani, H., Shao, S., Roghanchi, P., & Hassanalian, M. (2023).

Configurations and applications of multi-agent hybrid drone/unmanned ground vehicle for underground environments: A review. *Drones*, *7*(2), 136.

Dinelli, C., Racette, J., Escarcega, M., Lotero, S., Gordon, J., Montoya, J., Dunaway, C., Androulakis, V., Khaniani, H., Shao, S., Roghanchi, P., & Hassanalian, M. (2023). Configurations and Applications of Multi-Agent Hybrid Drone/Unmanned Ground Vehicle for Underground Environments: A Review. *Drones*, *7*(2), 136. https://doi.org/10.3390/drones7020136

Ding, J., Zhou, Y., Huang, X., Song, K., Lu, S., & Wang, L. (2023). An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. *Journal of Computational Science*, *67*, 101937. https://doi.org/10.1016/j.jocs.2022.101937

Duan, H. & Qiao, P. (2014). Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning". International Journal of Intelligent Computing and Cybernetics, 7(1), 24-37. https://doi.org/10.1108/ijicc-02-2014-0005.

Eppstein, D. (1998). Finding the k Shortest Paths. *SIAM Journal on Computing*, *28*(2), 652–673. https://doi.org/10.1137/S0097539795290477.

Fisher, P. F. (1991). First experiments in viewshed uncertainty: the accuracy of the viewshed area. *Photogrammetric engineering and remote sensing*, *57*(10), 1321-1327.

Foead, D., Ghifari, A., Kusuma, M. B., Hanafiah, N., & Gunawan, E. (2021). A Systematic Literature Review of A*Pathfinding. *Procedia Computer Science*, *179*, 507–514. https://doi.org/10.1016/j.procs.2021.01.034

Ganganath, N., Cheng, C., & Tse, C. K. (2015). A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains: IEEE Transactions on Industrial Informatics, 11(3), 601-611. https://doi.org/10.1109/tii.2015.2413355.

Gao, Y., Chen, F., & Wang, Z. (2019). Hybrid dynamic route planning model for pedestrian microscopic simulation at subway station. Journal of Advanced Transportation, 2019(1), 5207814.https://doi.org/10.1155/2019/5207814.

Guo, Y., Yao, D., Li, B., He, Z., Gao, H., & Li, L. (2022). Trajectory planning for an autonomous vehicle in spatially constrained environments. *IEEE Transactions on Intelligent Transportation Systems*, *23*(10), 18326-18336.

Hauser, K., & Latombe, J. C. (2010). Multi-modal motion planning in non-expansive spaces. *International Journal of Robotics Research*, *29*(7), 897–915. https://doi.org/10.1177/0278364909352098

Helmert, M. (2006). The fast downward planning system. Journal of Artificial Intelligence Research, 26, 191-246. https://doi.org/10.1613/jair.1705

Hong, Z., Sun, P., Tong, X., Pan, H., Zhou, R., Zhang, Y., & Xu, L. (2021). Improved A-Star algorithm for long-distance off-road path planning using terrain data map. *ISPRS International Journal of Geo-Information*, *10*(11), 785. https://doi.org/10.3390/ijgi10110785

Hosseinloo, S. H., Kanturska, U., Bell, M. G., & Fonzone, A. (2012). Multi-path multi-criteria routing of hazardous materials in time-dependent networks. In *Transport of Dangerous Goods: Methods and Tools for Reducing the Risks of Accidents and Terrorist Attack* (pp. 151-165). Springer Netherlands.

Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, *2022*, 1-10. https://doi.org/10.5194/gmd-15-5481-2022.

Hua, C., Niu, R., Yu, B., Zheng, X., Bai, R., & Zhang, S. (2022). A global path planning method for unmanned ground vehicles in off-road environments based on mobility prediction. *Machines*, *10*(5), 375.

Huang, J. (2008, September). Combining entropy weight and TOPSIS method for information system selection. In *2008 IEEE conference on cybernetics and intelligent systems* (pp. 1281-1284).

Hwang, Y. H. and Ahuja, N. (1992). A potential field approach to path planning. IEEE Transactions on Robotics and Automation, 8(1), page.23-32. https://doi.org/10.1109/70.127236.

Javaid, M. A. (2013). Understanding Dijkstra Algorithm. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2340905.

Karaman, S. & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 30(7), 846-894. https://doi.org/10.1177/0278364911406761.

Kavraki, L. E., Svestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, *12*(4), 566-580.

Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, *3*, 263-286. https://doi.org/10.1007/pl00011669.

Khatiwada S., Murray-Tuite P., & Schmid, M. (2023) Multi-criteria multi-agent path planning in unstructured off-road environments, Ground Vehicle Systems Engineering andTechnologySymposium,page.14. http://gvsets.ndia-mich.org/

Kumar, G. K. S., Aine, S., & Likhachev, M. (2022). Euclidean distance-optimal post-processing of grid-based paths. Proceedings of the International Conference on Automated Planning and Scheduling, 32, page.321-328. https://doi.org/10.1609/icaps.v32i1.19816.

LaValle, S. M. & Kuffner, J. (2001). Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), 378-400. https://doi.org/10.1177/02783640122067453.

Lingelbach, F. (2015). *On Probabilistic Completeness of Probabilistic Cell Decomposition*. http://arxiv.org/abs/1507.03727

Liu, L., Wang, J. J., & Wang, X. Y. (2016). Single machine due-window assignment scheduling with resource-dependent processing times to minimise total resource consumption cost. *International Journal of Production Research*, *54*(4), 1186–1195. https://doi.org/10.1080/00207543.2015.1056323

Loquercio, A., Maqueda, A. I., del-Blanco, C. R., & Scaramuzza, D. (2018). Dronet: learning to fly by driving. IEEE Robotics and Automation Letters, 3(2), 1088-1095. https://doi.org/10.1109/lra.2018.2795643.

Lv, T., & Feng, M. (2017). A smooth local path planning algorithm based on modified visibility graph. *Modern Physics Letters B*, *31*(19-21),1740091. https://doi.org/10.1142/S0217984917400917

Ma, J., Fukuda, D., & Schmöcker, J. D. (2013). Faster hyperpath generating algorithms for vehicle navigation. *Transportmetrica A: Transport Science*, *9*(10), 925–948. https://doi.org/10.1080/18128602.2012.719165

Marzouqi, M. S., & Jarvis, R. A. (2006). New visibility-based path-planning approach for covert robotic navigation. *Robotica*, *24*(6), 759–773. https://doi.org/10.1017/S0263574706002931

Moore, I. D., Grayson, R. B., & Ladson, A. R. (1991). Digital terrain modelling: a review of hydrological, geomorphological, and biological applications. *Hydrological processes*, *5*(1), 3-30. https://doi.org/10.1002/hyp.3360050103.

Nackaerts, K., Govers, G., & Orshoven, J. Van. (1999). Accuracy assessment of probabilistic visibilities. *International Journal of Geographical Information Science*, *13*(7), 709–721. https://doi.org/10.1080/136588199241076.

Qian, H., Wang, Z., Wang, J., & Chen, G. (2020). Stability-guaranteed and high terrain adaptability static gait for quadruped robots. Sensors, 20(17), 4911. https://doi.org/10.3390/s20174911.

Rehrer, S. E., Griffin, A. W., & Renner, M. (2022). Cross country mobility (CCM) modeling using triangulated irregular networks (TIN).

Ravela, S., Draper, B. A., Pinette, B., Ravela, S., Weiss, R., Draper, B., Hanson, A., & Riseman, E. (2001, November). Stealth Navigation: Planning and Behaviors. *Proceedings of ARPA Image Understanding Workshop*. https://www.researchgate.net/publication/2375767

Roy, J., Wan, N., Goswami, A., Vahidi, A., Jayakumar, P., & Zhang, C. (2018). A hierarchical route guidance framework for off-road connected vehicles. *Journal of Dynamic Systems, Measurement, and Control*, *140*(7), 071011.

Sabudin, E. N., Omar, R., Debnath, S. K., & Sulong, M. S. (2021). Efficient robotic path planning algorithm based on artificial potential field. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(6), 4840-4849.

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, *102*, 289–315.

Saranya, C., Unnikrishnan, M., Ali, S. A., Sheela, D. S., & Lalithambika, V. R. (2016). Terrain based D∗ algorithm for path planning. *IFAC-PapersOnLine*, *49*(1), 178–182. https://doi.org/10.1016/j.ifacol.2016.03.049.

Shorakaei, H., Vahdani, M., Imani, B., & Gholami, A. (2014). "Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm." Robotica, 34(4), 823-836. https://doi.org/10.1017/s0263574714001878.

Silva, C. & Ribeiro, B. (2003). Navigating mobile robots with a modular neural Architecture. Neural Computing & Applications, 12, page.200-211. https://doi.org/10.1007/s00521-003-0383-y.

Spiess, H., & Florian, M. (1989). Optimal strategies: a new assignment model for transit networks. *Transportation research part b: methodological*, *23*(2), 83-102.

Tian, F., Zhou, R., Li Z.,Gao, Y., Cao, D., Chen, L. (2021). Trajectory Planning for Autonomous Mining Trucks Considering Terrain Constraints. IEEE Trans. Intell. Vol., 6, 772–786.

U.S. Geological Survey, ScienceBase, 2013. [Online]. Available: https://www.sciencebase.gov/catalog/item/5f7784fc82ce1d74e7d6cbf2.[Accessed:17-Feb-2024].

Varadhan, G., Krishnan, S., Sriram, T. V. N., & Manocha, D. (2006). A simple algorithm for complete motion planning of translating polyhedral robots. *International Journal of Robotics Research*, *25*(11), 1049–1070. https://doi.org/10.1177/0278364906071199

Verbas, I. Ö., & Mahmassani, H. S. (2015). Finding least cost hyperpaths in multimodal transit networks methodology, algorithm, and large-scale application. *Transportation Research Record*, *2497*, 95–105. https://doi.org/10.3141/2497-10

Vlaski, S., & Sayed, A. H. (2021). Distributed Learning in Non-Convex Environments-Part I: Agreement at a Linear Rate. *IEEE Transactions on Signal Processing*, *69*, 1242–1256. https://doi.org/10.1109/TSP.2021.3050858.

Wagner, G., & Choset, H. (2015). Subdimensional expansion for multirobot path planning. *Artificial intelligence*, *219*, 1-24.

Wang, R., Lu, Z., Jin, Y., & Liang, C. (2022). Application of A* algorithm in intelligent vehicle path planning. Mathematical Models in Engineering, 8(3), page.82-90. https://doi.org/10.21595/mme.2022.22828.

Wei, K. & Ren, B. (2018). A method on Dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. Sensors, 18(2), 571. https://doi.org/10.3390/s18020571.

Xu, Z., Deng, D., & Shimada, K. (2021). Autonomous UAV Exploration of Dynamic Environments Via Incremental Sampling and Probabilistic Roadmap. *IEEE Robotics and Automation Letters*, *6*(2), 2729–2736. https://doi.org/10.1109/LRA.2021.3062008

Yan, L., Wang, P., Yang, J., Hu, Y., Han, Y., & Yao, J. (2021). Refined path planning for emergency rescue vehicles on congested urban arterial roads via reinforcement learning approach. *Journal of Advanced Transportation*, *2021*(1), 8772688. https://doi.org/10.1155/2021/8772688.

Yan, Y. (2023). Research on the A Star Algorithm for Finding Shortest Path. *Highlights in Science, Engineering and Technology*, *46*, 154-161.

Yen, J. Y. (1971). Finding the K Shortest Loopless Paths in a Network. In *Source: Management Science* (Vol. 17, Issue 11).

Zhang, X., Lai, J., Xu, D., Li, H., & Fu, M. (2020). 2D Lidar-Based SLAM and Path Planning for Indoor Rescue Using Mobile Robots. Journal of Advanced Transportation, 2020(1), 8867937.

Zhao, H., Zhang, B., Sun, J., Lei, Y., & Yu, H. (2021). Spot-Welding path planning method for the curved surface workpiece of body-in-white based on a memetic algorithm. The International Journal of Advanced Manufacturing Technology, 117(9-10), 3083-3100. https://doi.org/10.1007/s00170-021-07728-6.

Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., & Srinivasa, S. (2009). Planning-based prediction for pedestrians. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 3931–3936. https://doi.org/10.1109/IROS.2009.5354147

APPENDIX A

Table A link parameters case 3

			Scenario A		Scenario B		Scenario C	
Link		**C**	**R**	**D**	**R**	**D**	**R**	**D**
1	2	1.37	0.94	0.41	0.47	0.21	0.28	0.12
1	6	1.32	0.46	0.009	0.23	0.004	0.14	0.002
2	7	1.40	0.13	0.87	0.06	0.44	0.03	0.26
2	3	1.96	0.30	0.99	0.15	0.49	0.092	0.298
3	8	1.44	0.77	0.73	0.39	0.37	0.23	0.219
3	4	1.25	0.07	0.25	0.035	0.13	0.02	0.078
4	9	1.23	0.18	0.39	0.091	0.19	0.056	0.12
4	5	1.00	0.76	0.55	0.38	0.28	0.23	0.17
5	10	1.82	0.03	0.68	0.02	0.34	0.0098	0.21
6	11	1.04	0.70	0.13	0.35	0.069	0.21	0.041
6	7	1.44	0.89	0.71	0.45	0.35	0.27	0.215
7	12	1.87	0.70	0.58	0.35	0.29	0.21	0.17
7	8	1.37	0.81	0.37	0.41	0.19	0.24	0.11
8	13	1.20	0.45	0.94	0.35	0.48	0.14	0.28
8	9	1.03	0.10	0.01	0.05	0.0062	0.032	0.0037
9	14	1.99	0.23	0.003	0.12	0.0018	0.071	0.001
9	10	1.51	0.72	0.36	0.36	0.18	0.18	0.05
10	15	1.87	0.07	0.78	0.03	0.39	0.022	0.109
11	16	1.60	0.67	0.76	0.33	0.38	0.20	0.13
11	12	1.88	0.26	0.34	0.13	0.17	0.081	0.24
12	17	1.77	0.55	0.85	0.27	0.42	0.17	0.23
12	13	1.30	0.61	0.15	0.30	0.075	0.18	0.045
13	18	1.16	0.91	0.36	0.45	0.18	0.27	0.109
13	14	1.92	0.74	0.44	0.37	0.22	0.22	0.13
14	19	1.62	0.70	0.89	0.35	0.44	0.21	0.27
14	15	1.85	0.62	0.21	0.31	0.10	0.18	0.062
15	20	1.67	0.16	0.54	0.079	0.27	0.04	0.16
16	21	1.71	0.087	0.38	0.043	0.19	0.026	0.11
16	17	1.39	0.28	0.038	0.143	0.01	0.08	0.011
16	24	1.34	0.29	0.966	0.145	0.48	0.087	0.29
17	22	1.12	0.79	0.76	0.39	0.38	0.23	0.22
17	18	1.27	0.87	0.16	0.43	0.078	0.26	0.047
18	23	1.79	0.49	0.40	0.24	0.20	0.15	0.12
18	19	1.71	0.60	0.73	0.30	0.37	0.18	0.22
19	24	1.26	0.59	0.83	0.29	0.42	0.17	0.24
19	20	1.34	0.85	0.81	0.42	0.40	0.25	0.23
21	22	1.57	0.98	0.87	0.49	0.44	0.29	0.26
23	24	1.88	0.43	0.54	0.21	0.27	0.13	0.16
24	25	1.35	0.56	0.95	0.28	0.47	0.17	0.28