

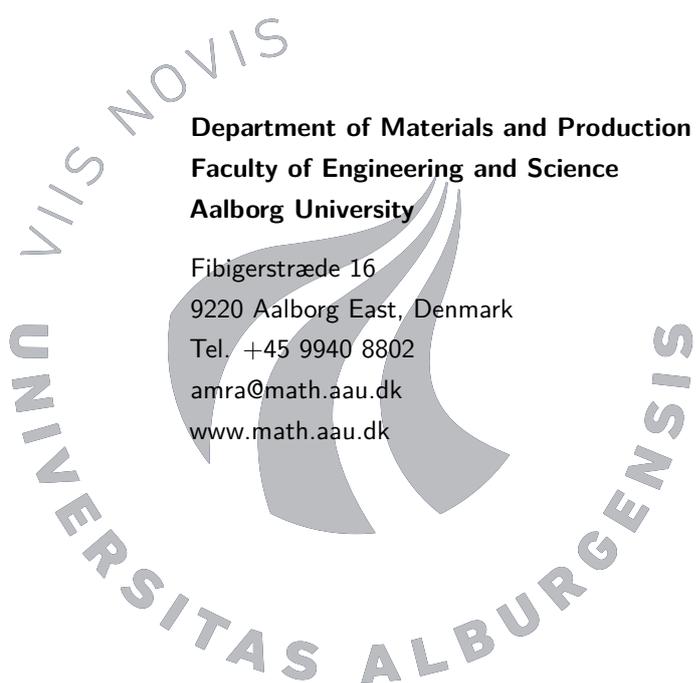
SAFE PATH PLANNING UNDER UNCERTAINTY



GROUP 3.222B
MATHEMATICS-ECONOMICS

SUPERVISOR: INKYUNG SUNG
SECONDARY SUPERVISOR: PETER NIELSEN
03-06-2020

DEPARTMENT OF MATERIALS AND PRODUCTION
AALBORG UNIVERSITY



Department of Materials and Production
Faculty of Engineering and Science
Aalborg University

Fibigerstræde 16
9220 Aalborg East, Denmark
Tel. +45 9940 8802
amra@math.aau.dk
www.math.aau.dk



AALBORG UNIVERSITY
STUDENT REPORT

Title:

Safe Path Planning Under Uncertainty

Theme:

Optimisation

Project period:

3. February 2020 - 3. June 2020

Author:

Emil Ingemann Pedersen

Supervisor:

Inkyung Sung

Secondary supervisor:

Peter Nielsen

Circulation figures : 4

Project number of pages: 33

Appendix number of pages: 20

Total number of pages: 53

Project completed: June 3rd, 2020

10th term at

Department of Materials and Production

Mathematics - Economics

Fibigerstræde 16,

9220 Aalborg East

<http://math.aau.dk>

Abstract:

This thesis investigates how to make a safe path plan for a UAV when some areas in a map are desired to avoid, and furthermore account for uncertainty to where in the map the basis of the areas to avoid is located.

The thesis is initiated with a general introduction to various path planning methods before the problem is stated. Afterwards an introduction is given to both the map setup, and how to traverse the map. Thereby it is possible to elaborate further on the problem, where the uncertainty in regard of the areas to avoid presents multiple scenarios to take into consideration.

The map is traversed by utilising a grid-based graph where the risk minimising path is found by application of dynamic programming with resource constraint in the shape of a Label Setting Algorithm. Furthermore the algorithm is altered to be able to find a robust path, when having multiple scenarios to accommodate for. Lastly the algorithm and general setup is evaluated and discussed with a sensitivity analysis being the basis of discussion.

Preface

This thesis is made in the period from 3rd of February 2020 to the 3rd of June 2020 at the Department of Materials and Production at Aalborg University. The thesis is conducted by Emil Ingemann Pedersen who is a Mathematics-Economics student at 10th semester, specialising in Operations Research. The main theme of the project is *Path Planning under Uncertainty*.

The references to sources are to be found in the bibliography. When referencing directly the *Harvard method* is used. Which means that in the text the source is referred by (*Lastname, Year*). When referring equations, there is written: equation (x.x). Lastly, great appreciation is given to Inkyung Sung and Peter Nielsen for their guidance.

Aalborg University the 3rd of June 2020.



Emil Ingemann Pedersen

Contents

| | | |
|-------|--|----|
| 1 | Introduction | 1 |
| 1.1 | Problem Description | 2 |
| 2 | Literature Review | 5 |
| 3 | Modelling | 7 |
| 3.1 | Probability Risk Map | 8 |
| 3.2 | Problem Setups | 10 |
| 3.2.1 | Problem Statement | 11 |
| 3.3 | Shortest Path Problem w. Resource Constraint | 13 |
| 3.3.1 | Dynamic Programming w. Resource Constraint | 13 |
| 4 | Experiments | 17 |
| 4.1 | Sensitivity Analysis | 17 |
| 4.2 | Time Complexity | 26 |
| 5 | Discussion and Perspectives | 29 |
| 5.1 | Discussion | 29 |
| 5.2 | Future Work | 30 |
| 5.3 | Perspectives | 31 |
| 6 | Conclusion | 33 |
| A | Plots, Algorithms and Tables | 35 |
| A.1 | Modelling Plot | 35 |
| A.2 | 5 Scenarios Robust Paths | 36 |
| A.3 | Robust paths From Deterministic path | 37 |
| A.4 | Experiments Plots | 39 |
| A.5 | Tables | 43 |
| A.6 | Floyd-Warshall Shortest Path | 44 |
| B | R scripts | 45 |
| B.1 | Label Setting Algorithm | 45 |
| | Bibliography | 51 |

CHAPTER 1

Introduction

It is imaginable that since the beginning of even primal animal species, the desire of reaching a target (food, sleeping area, etc.) in the *shortest path* possible has been essential for surviving (Schrijver, 2012). Thereby it does not seem outrageous to claim that *path finding* is an instinct within ourselves. In the technological world of present time, the desire of saving time and assuring that complex tasks are taken care of in a well-founded manner is still increasing. An example of this is the 1950's research in *alternate routing* examining how to find the second shortest route if the shortest was unavailable. The alternate routing contributed to the automatisisation of the long-distance telephone calls in the United States (Schrijver, 2001). Since then the path and route planning have been studied extensively, some of the well-known methods and problems worth mentioning are the Bellman-Ford method from 1958, the Dijkstra method from 1959, the Vehicle Routing Problem from 1959, and later the A* algorithm from 1968 (Schrijver, 2001; Hart et al., 1968). The A* algorithm is often applied for path finding in video games. The algorithm was created as a part of a larger program trying to build a mobile robot that could reason about its own actions, it was called the Shakey program (Cassel, 2017).

Since the introduction of the Shakey program, the world has come a long way in further automating various fields, as it is now possible to get passenger cars to almost drive by themselves (Walker, 2020). However, the field of automated robots and vehicles is still under major research. Another type of vehicle where automatisisation of the field is explored is for aerial vehicles.

An unmanned aerial vehicle, UAV, is an aerial vehicle not carrying any humans but instead is either piloted by a computationally driven program or remotely. UAVs are utilised in a broad sense of fields, e.g., military (Zhang et al., 2015; Wu et al., 2014), and industrial companies are exploiting the capabilities of UAVs to do reconnaissance, delivering goods in impassable areas, etc. (Phung et al., 2017; Boccardo et al., 2015). For the UAVs to perform well in these tasks, a lot of planning needs to be executed. The planning covers a wide range of aspects, besides the bureaucratic aspects, some technical aspects are also carried out, among these the *path planning* of the UAV (Pfeiffer et al., 2007).

In the various applications of path planning of UAVs, it is often found that the UAV needs to avoid certain areas (Souissi et al., 2013; Blackmore et al., 2006; Alotaibia et al., 2018). This could be due to a possible intersect with obstacles, or because some areas might be related to risk exposure. The risk exposure could either be towards the UAV, e.g. if the UAV is carrying out a military task. On the other hand, the UAV could also be exposing others to risk, e.g., endangered animal species, or humans (Mulero-Pázmány et al., 2017).

When having certain areas which are preferable to avoid, while still taking into account the limitation of fuel, and thus distance the UAV can fly, the path planning is non-trivial (Pfeiffer et al., 2007). Furthermore relying on perfect information about the localisation of

the area it is desired to avoid, could jeopardise the task of the UAV (Drawil and Amar, 2013). Therefore the path planning should consider uncertainty in the information about the reason of avoidance, the uncertainty could be due to, mobile objects, e.g., mobile threats or animals moving around, or insufficient localisation tools.

1.1 Problem Description

Appropriate path planning is a necessity for the UAV to carry out a task satisfyingly. Motivated by said necessity and the former introduction, this thesis considers the problem of safe path planning for a single UAV under environmental uncertainty. For analytical simplicity the dynamics of flying the UAV are ignored, this could be dynamics such as turning radius, ascending altitude, etc. It is further assumed that the UAV is not in any way reactive and thereby stays on the path given.

In addition, it is assumed that every area of avoidance has a basis or center point. Hence the desire of avoiding the area should be greatest at the center point. The area of avoidance is described by a Gaussian bivariate normal distribution around the center point (x_z, y_z) , where the uncertainty of the position of the center point is taken into consideration by ξ_u , where $u \in U_s$ denotes the set indexing the scenarios which represent the state of a risk map, $U_s = \{1, 2, \dots, s\}$.

$$f_z(x, y, \xi_u) = \frac{1}{\sqrt{2\pi}\sigma_z} \cdot e^{-\frac{d_z^2}{2\sigma_z^2}} \quad (1.1)$$

Here d_z is the euclidean distance between an arbitrary point, (x, y) , and the center point of avoidance, $d_z = \sqrt{(x - [x_z \pm \xi_u])^2 + (y - [y_z \pm \xi_u])^2}$, and σ_z is the standard deviation.

If multiple areas of avoidance are present in the region, where the UAV is carrying out a task, the risk exposure is described accordingly.

$$F(x, y, \xi_u) = 1 - \prod_{z=1}^M (1 - f_z(x, y, \xi_u)) \quad (1.2)$$

In this thesis the region, the UAV is carrying out a task in, is seen as a discrete map. The map is split into a $K \times K$ grid, seen as a graph $G(N, E)$ with N being the nodes, from now referred to as points, and E being the edges combining the points. A path in the graph is said to start at an origin, o , and end the task at the goal, g . The path is given by a string of points $p = \{p_0, p_1, \dots, p_N\}$ where $p_0 = o$ and $p_N = g$. In this map the cost of travelling from one point, i , to another point, j , is associated with the risk of arriving at the point j .

$$c_{ij} = F(p_j, \xi_u) \quad (1.3)$$

Thus the risk exposure of a given path, $R(p, \xi)$, is seen as the sum of the risk from every point which the path consists of.

$$R(p, \xi_u) = \sum_{i=0}^N F(p_i, \xi_u) \quad (1.4)$$

When a UAV traverse a region, where certain areas are desired to avoid, the UAV is either exposed to threats, and thereby risk, or exposing others, e.g., animals to risk. By

introducing the risk measure of equation (1.4), it is seen that the risk increases along the path, the increasing risk might, at some point, lead to the UAV being detected by enemies or endangered animal species. By introducing $X(p, \xi_u)$ as a random variable, the number of detections, d , the UAV obtains, can be counted. That is, as the risk exposure reaches certain thresholds along a path, p , the number of detections rises. Thus by introducing $X(p, \xi_u)$ the probability of being detected, or scare animals, can be described by a Poisson process.

$$P(X(p, \xi_u) = d) = \frac{1}{d!} (R(p, \xi_u))^d \cdot e^{-R(p, \xi_u)} \quad d \in \mathbb{N}^0 \quad (1.5)$$

Furthermore, as indicated earlier, it is presumed that the UAV might be under some distance constraint, which could be due to fuel consumption, or that the task is urgent and needs to be carried out within a certain timeframe. Thus the distance utilised by the UAV to get from one point to another is taken to be the distance, D , between two points. The distance between two points is seen as the euclidean distance between them $D_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$.

The distance of a given path is thus the sum of the distances between all linked points on that specific path.

This yields an objective function where the desire is to minimise the number of detections of a path while satisfying a time constraint.

$$\begin{aligned} & \text{minimize} && P(X(p, \xi_u) \geq d) \\ & \text{subject to} && D_g \leq \bar{D} \end{aligned} \quad (1.6)$$

Where $D_g = D(p)$ is the accumulated distance covered on the path, and \bar{D} is a threshold of distance utilised to complete the task. Looking further into minimising the number of detections it is seen that

$$\text{Minimise } P(X(p, \xi_u) \geq 1) \equiv \text{Maximise } P(X(p, \xi_u) = 0)$$

In addition $P(X(p, \xi_u) = 0) = e^{-R(p, \xi_u)}$ by definition. Therefore minimising the probability of 1 or more detections is equivalent to minimising the risk across the path.

Preliminary Problem Statement

The preliminary focus of this thesis will be to analyse the map setup and find a way to evaluate the objective function in equation (1.6) by looking into the following statements:

- How to accommodate for the uncertainty having multiple scenarios?
- How can the objective function be solved considering multiple scenarios?

Limitation

The region of the task is taken to be partially known, as the localisation of the areas of avoidance could be incorrect or associated with some uncertainty. Further, the graphical composition of a region is not taken into consideration, as the risk from unspecified objects is the main focus to accommodate for. It is assumed that the altitude of the UAV is fixed,

and thus the UAV is already in an appropriate altitude at the origin. This means that the mapping and path planning is made in 2 dimensions. This thesis only looks into a global path plan, that is, the path is planned on knowledge obtained prior to take off. Thereby local path planning, and alterations to the global path while the UAV is carrying out a task are not taken into consideration. For simplicity of the setup and without loss of generality the pathing generated throughout this thesis takes place from point 0, to point $K \cdot K$, that is, from one end of the map to the other.

Methods

The methods applied to find an optimal solution to the objective function, and thus minimising the risk of a path for the UAV, are combinatorial optimisation, stochastic programming, and dynamic programming.

In operations research, combinatorial optimisation is often applied in setups where the optimal solution to a problem comes from a finite set of solutions. In this thesis, the combinatorial optimisation is a part of finding the path minimising the probability of detection, as the pathing consists of traversing through a map with a graph setup. Exhaustive search methods become non-tractable as the level of detail desired in the map rises, to help this issue *dynamic programming with resource constraint* is introduced.

The uncertainty of the center points regarding the areas of avoidance is investigated with a stochastic approach. This is carried out through application of *stochastic programming* evaluating the path by minimising either the expected number of detections or by finding a robust solution, which minimises the maximum regret.

CHAPTER 2

Literature Review

The practical application of UAVs has intensified over the last decades and the same has the studies concerning how to plan their pathing. The studies considered relevant to the this paper are studies in various schemes of path planning; *in uncertain environments, with obstacles*, and lastly *under threat*. Many different problem structures and methods have been applied in the various studies, further, it is noted that the methods applied and the general map setup overlap between the mentioned schemes.

Path planning with obstacles has been studied by (Blackmore et al., 2011). Here a chance constraint approach is introduced, where the probability of failure, that is, colliding with an obstacle, is minimised. Additionally, the uncertainty of the state of the UAV is considered by adding a Gaussian white noise term at every state. The problem is solved with a self-customised algorithm. (Kothari and Postlethwaite, 2013) also applies a chance constraint approach for guaranteeing a safe path without collision, the *Rapidly exploring Random Trees*, RRT, algorithm is applied with good results, however without considering the time or fuel consumption.

In (Miralles and Sanz-Bobi, 2004) the path planning is carried out by utilising *potential fields*. In this study with a map where obstacles are represented as multivariate Gaussian distributions, with the possibility of collision increasing when pathing closer to the obstacle. Thus the pathing is carried out by navigating in the valleys of probability, which in conclusion is almost the same as utilising *Voronoi diagrams*, both setups are suffering from the same drawback, having long path lengths, mainly as the path length is not considered.

Other studies have looked into the environment which the UAV is passing through, and the knowledge of the said environment. In these studies both (Yang et al., 2014; Bry and Roy, 2011) apply the RRT algorithm, whereas (Yang et al., 2014) suggests an improvement to the RRT algorithm both in sense of iterations and path smoothness and thereby the path length. Both articles consider uncertainty as well, (Yang et al., 2014) by adding a chosen margin to the obstacles or threats, and removing the areas with obstacles from the path-able region, thus assuring a safer path.

In (Bry and Roy, 2011) the path is planned while predicting uncertainty, with moderate satisfaction. The methodology changes completely in (Rathbun and Capozzi, 2002), where an evolutionary algorithm is applied to optimise the path in an uncertain environment, here a weighted trade-off between path length and risk of collision is introduced with varying results. A common denominator for all these is that they look into an environment where the areas of avoidance are seen as obstacles.

The only scheme where the risk is not linked to collision is when examining path planning

under threat, the applications and setups are however still somewhat similar. In (Kim and Hespanha, 2004), threats are modeled as ellipsoids, with monotone increasing risk level towards the center point, the path planning is here carried out by applying a Voronoi diagram, additionally, uncertainty is not considered in any regard.

Another way to model the threats is seen in (Zhang et al., 2015) where the risk of being under threat is described by a Gaussian bivariate normal distribution with the map being split into a grid. The path planning is carried out by application of a new algorithm using reinforcement learning through a reward matrix minimising risk.

A third way of modeling the risk while finding an optimal path is introduced in (Pfeiffer et al., 2007) where the map, the UAV is operating in, is depicted as threat zones, furthermore risk is here related to the number of detections. However the methodology is not described, neither is uncertainty introduced in any aspect.

The focus of this thesis is to create a setup where safe path planning under uncertainty is possible. This is carried out with inspiration of (Zhang et al., 2015) using the Gaussian normal distributions to describe mapping and risk, however in this thesis uncertainty of the exact location is introduced, and thereby various scenarios, which all need to be accommodated for, are present. Moreover, in (Zhang et al., 2015) the objective includes both distance and risk in the same expression. Whereas the focus in this thesis is only regarding the risk and then accounting for the distance covered to execute the task by adding a constraint. In addition, the risk measure is reformulated into number of detections, as this seems more relatable, this is carried out in the same manner as (Pfeiffer et al., 2007).

The combination of a map with areas of avoidance described by Gaussian normal distributions and having various scenarios of how uncertain the positions of the areas of avoidance are is to the knowledge of this thesis, not investigated before. Furthermore, the primary focus on risk makes the objective of this thesis different from the previous studies with a similar map setup. The path planning and risk minimisation are carried out by application of dynamic programming with resource constraint. The method has previously been applied to *shortest path problems with recourse constraint* with success, e.g., in various settings of the known vehicle routing problem (Righini and Salani, 2008). The dynamic programming method applied is in addition altered to a robust setting, from which it is possible to find a path being both feasible and a good solution in all scenarios.

CHAPTER 3

Modelling

The approach for the path planning described throughout this thesis is visualised in figure 3.1, in this chapter the various components of the preprocessing and applications are described, that is, the problem setup required, to describe the path with minimum risk and the methodology applied to find said path. Furthermore, The stochastic approach, regarding the uncertainty of the center points of the areas of avoidance, is also elaborated.

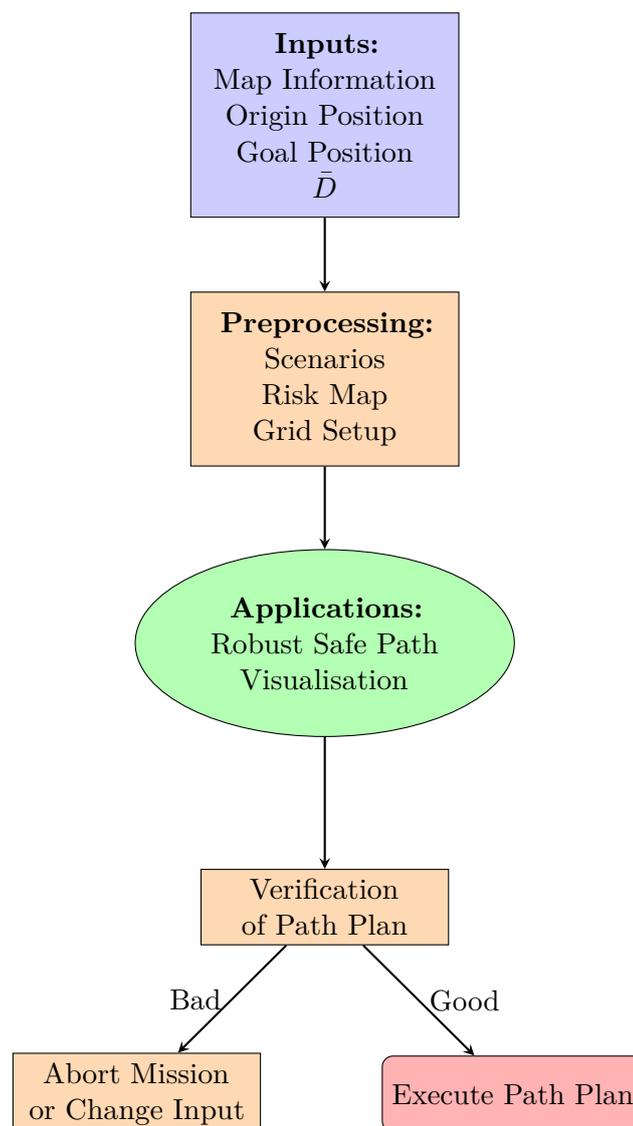


Figure 3.1: Visualisation of the general path planning approach.

3.1 Probability Risk Map

The UAV traverse through a region where, in the path planning, the graphical composition of said region is not taken into consideration. Therefore the map is categorised as a *Probability Risk Map*, PRM, where the risk is modelled as a Gaussian bivariate normal distribution, this is only when the uncertainty, ξ_u , is 0. When $\xi_u \neq 0$ the area of avoidance takes a shape where the uncertainty makes a circular plateau of high risk. The plateau is bounded by ξ_u , thus the areas of avoidance described by equation (1.1), is described in higher detail by

$$f_z(x, y, \xi_u) = \frac{1}{\sqrt{2\pi}\sigma_z} \mathbb{1}_{\sqrt{(x-x_z)^2 + (y-y_z)^2} \leq \xi_u} + \frac{1}{\sqrt{2\pi}\sigma_z} \cdot e^{\frac{-d_b^2}{2\sigma_z}} \mathbb{1}_{\sqrt{(x-x_z)^2 + (y-y_z)^2} > \xi_u} \quad (3.1)$$

where $d_b = \sqrt{(x-x_b)^2 + (y-y_b)^2}$ is the euclidean distance between some arbitrary point, (x, y) , and the closest point (x_b, y_b) on the boundary line of the plateau of risk. The formulation of equation (3.1) describes the risk as a variety of distributions depending on the values of σ and ξ_u , this includes Gaussian bivariate distribution and the uniform bivariate distribution as the limiting cases, when $\xi_u = 0$, or $\sigma = 0$, respectively. Throughout this thesis the distribution is utilised with $\sigma = 1$ and ξ_u varying, which gives multiple scenarios to look into. A visualisation of the probabilistic risk map with a single area to avoid and varying values of ξ_u is seen in figure 3.2.

The reason to utilise the boundary points of the circular plateau stems from the uncertainty, as it is unknown where within the plateau the actual reason to avoid the area is positioned, thus the decrease of risk starts from the boundary and not from the center point. The density is thereby said to be *bivariate Gaussian-tailed uniform*. A map with multiple areas of avoidance is seen in Appendix figure A.1.

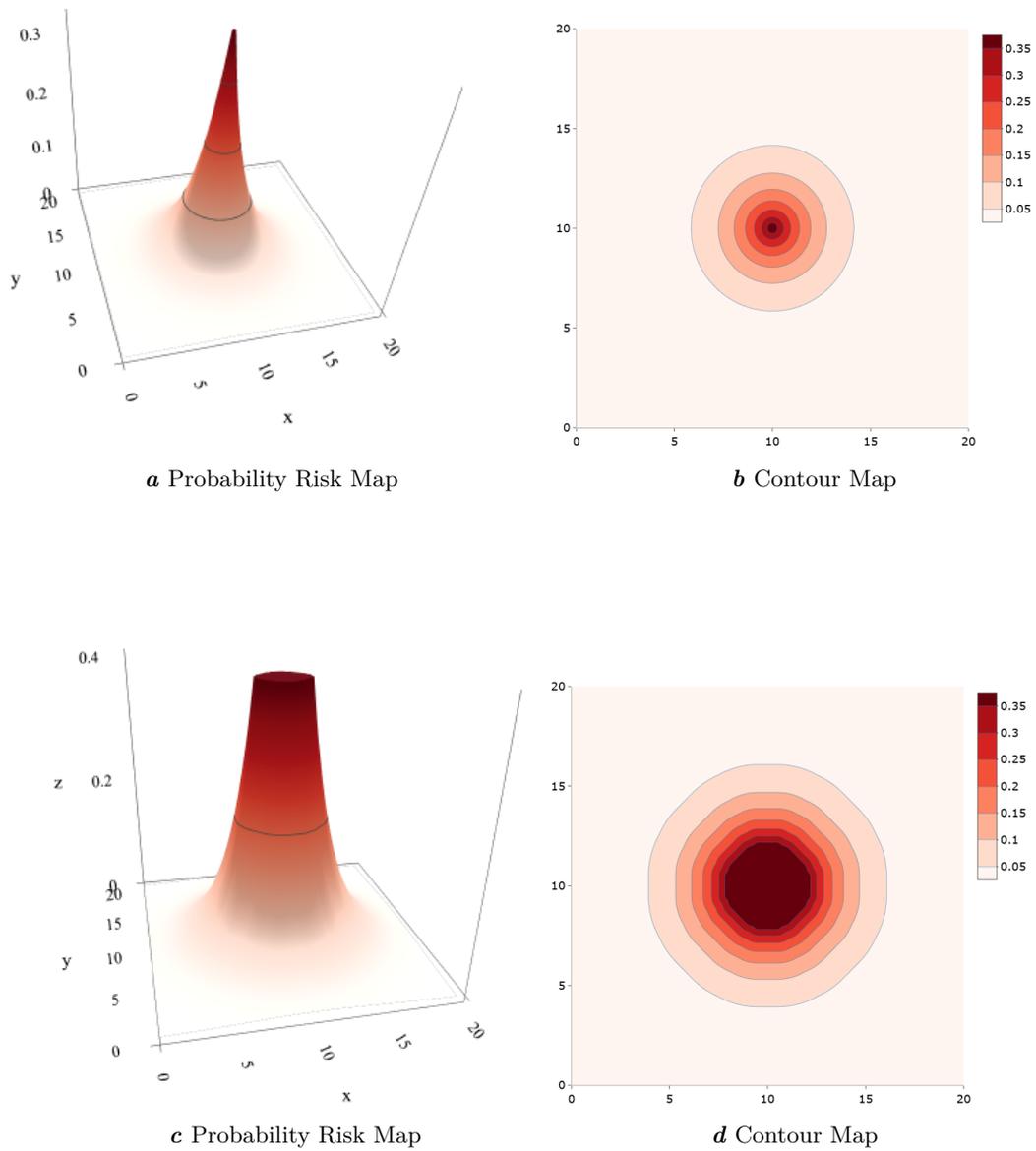


Figure 3.2: Illustrations of area of avoidance with center point (10,10) and $\xi = 0$ in a and b, and $\xi = 2$ in c and d.

3.2 Problem Setups

The risk of a path and the total volume of risk in a map is highly connected with the uncertainty, therefore the scenarios representing various uncertainties influence the path with minimal risk, and thus all scenarios should be considered when minimising the risk and thereby the probability of detection. One possible way to accommodate for this is by looking into the expected value of the risk from equation (1.4) and thereby the expected value of the number of detections. This yields an objective function given by

$$\begin{aligned} & \text{minimize} && P\left(\mathbb{E}\left[X(p, \xi)\right] \geq d\right) \\ & \text{subject to} && D_g \leq \bar{D}. \end{aligned} \tag{3.2}$$

With this objective function the expected value is taken accordingly in equation (1.1) - equation (1.4). By solving equation (3.2) one would obtain a pathing which in the average scenario minimises the probability of detection by the Law of Large Numbers (Shapiro and Philpott, 2007). However, it is likely that the need of carrying out the same task in the same location multiple times, is not what the owner of the UAV requests, and thus the map might change with every task. Furthermore finding the path with minimal probability of detection in the average scenario could still lead to the path having a high probability of detection in a worst-case scenario (Shapiro et al., 2013). A UAV is an expensive tool, and detection of the UAV in a hostile environment would very likely result in the UAV being eliminated (Pfeiffer et al., 2007). Therefore it is also a desire to look into the possibility of finding a solution that incorporates the information from all scenarios in the decided path, and thereby becomes more robust, no matter which scenario unfolds. The robust path is seen as the safest path when multiple scenarios are considered.

The robust path is in this thesis found by minimising the maximum regret. To find the regret of a path it is a necessity to find the lowest risk path for each of the scenarios, $u \in U_s$, the path with minimum risk for a given scenario is denoted, $R(p^*, \xi_u)$. Thereby the regret is given as

$$Reg(p, \xi_u) = R(p, \xi_u) - R(p^*, \xi_u) \tag{3.3}$$

where p is an arbitrary path, and p^* is the path with lowest risk in a given scenario $u \in U_s$. The regret of a path is needed when finding the maximum regret, given the various scenarios $u \in U_s$.

$$MR(p) = \max_{u \in U_s} Reg(p, \xi_u)$$

The maximum regret is thereby the regret of an arbitrary p , in the scenario where the regret, compared to the optimal path of said scenario, is largest. Thus the robust path is found by solving the objective function given by,

$$\begin{aligned} & \text{minimize} && MR(p) \\ & \text{subject to} && D_g \leq \bar{D}. \end{aligned} \tag{3.4}$$

3.2.1 Problem Statement

With the above elaboration on the objective function from equation (1.6) and the clarification on how to optimise the path in regard of risk, while considering all scenarios.

The problem considered throughout this thesis is:

How to minimise the risk of the path for a UAV and thus find optimal solutions to the two new objective functions equation (3.2) and equation (3.4)?

This question is answered by looking into the following statements:

- How to path through the PRM?
- How to find paths minimising the probability of detection?
- Does the introduction of uncertainties change the paths?
- How can a robust path be found without solving (1.6) for each distinct scenario?

Grid

To be able to create a path plan through the PRM, the map is transformed into a graph having $K \times K$ points, the graph is constructed as a grid, see figure 3.3. By introducing the grid-based graph it should be noticed that the detail of the map and thereby the number of possible paths is highly dependant on K . The path planning will throughout this thesis be carried out with a 40×40 grid.

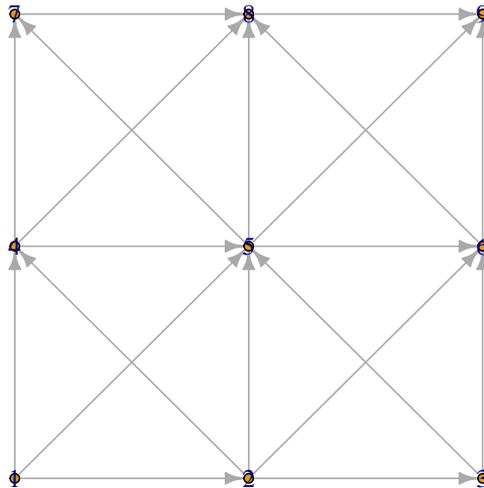


Figure 3.3: Illustration of a 3×3 grid based graph.

By choosing the grid-based graph setup as framework to find a path satisfying the objective functions equation (3.2) and equation (3.4), and traverse the PRM, other setups are discarded. Two other methods utilised in setups like this are *Voronoi Uncertainty Fields*, VUF, and RRT (Tsardoulis et al., 2016). The VUF relies on Voronoi diagrams and is often applied when dealing with obstacles. In this method, the edges define the possible paths that maximises the distance to the obstacles (Masehian and Amin-Naseri, 2004). This seems preferable as minimising the risk is the main focus, however, the method lacks the possibility of considering the length of the path, while still finding a path minimising risk. Furthermore, the risk is not an obstacle and thus it should be possible to path through an area of avoidance. Lastly, the areas of avoidance are of different size given the various scenarios, and a comparison of the paths in different scenarios could become biased when allowing for a more floating graph structure (Toth et al., 2017).

RRT methods are also often utilised in path planning, the RRT, in general, have a simple algorithmic setup which the performance benefits from in terms of computational complexity, therefore the method is mainly applied in real-time or local path planning. Further, the RRT mainly handles obstacles instead of areas of risk, even if the areas

of avoidance were to be depicted as obstacles the methods exhibit problems on passing through narrow spaces (Szadeczky-Kardoss and Kiss, 2006). Additionally, the pathing created with RRT is non-deterministic and might change if the same map is handled multiple times (Jouandeau et al., 2008), therefore one could question the optimality of a path when utilising RRT.

The grid-based graph setup also lacks the ability to find the optimal path to some extent, as the grid-based setup constrains the possible directions to investigate. However increasing the density of the grid by increasing K , would also lead to an increase in the ability to find the optimal path. Therefore the grid-based graph setup is chosen. By employing the grid setup it needs to be chosen which application method to utilise to find the path which minimises risk, and also has the possibility of considering multiple scenarios to find a robust safe path.

3.3 Shortest Path Problem w. Resource Constraint

When having the setup of a directed graph, visualised in figure 3.3, there are two methodologies applied widely to solve the shortest path problem with resource constraint, the first methodology utilises heuristics from Branch-and-Bound family, i.e. Branch-and-Price and Branch-and-Cut (Ladányi et al., 2001). These methods find the shortest path in a set of candidate paths, by branching the set of candidate paths and discarding the paths not satisfying a lower bound criterion. The lower bound criterion is therefore very important, and if a good lower bound is not found the method can be extremely time-consuming (Pinedo, 2012). In the grid-based graph setup, the number of possible paths increases rapidly as K increases. A lower estimate on the number of possible paths in the graph can be set by the *Delannoy number* (Banderier and Schwer, 2005). The Delannoy number is the number of paths in a grid-based graph where it is only allowed to path, upwards, to the right, and along the up/right diagonal, with this construction the number of possible paths in a 3×3 grid would be 13, however by allowing for a diagonal move towards the up/left as in figure 3.3, the possible paths increases and for a 3×3 grid the total number of possible paths becomes 33. Nonetheless, the Delannoy number is used as a generous lower estimate, the Delannoy number in a 20×20 grid is more than $260 \cdot 10^{12}$. Meaning that the memory required to store the set of candidate solutions, with the application of a Branch and Bound based method, could become a problem (Luedtke, 2016).

The second methodology widely applied to the shortest path problem with resource constraint is dynamic programming, in this thesis in the format of *A Label Setting Algorithm*.

3.3.1 Dynamic Programming w. Resource Constraint

Dynamic programming is a methodology which in general solves optimisation problems that include making a string of decisions, decisions that can be seen as subproblems to be solved with the same approach as the original problem. Thereby a solution to the original problem is found by optimal solutions to the subproblems. Further, the subproblem approach benefits the memory required, as it is only good solutions to the subproblems which are memorised (Lew and Mauch, 2007).

The Label Setting Algorithm

The algorithm applies the methodology of dynamic programming, as it utilises a set of labels on each point in the graph, where each label at a point represents a path from the origin to that point. Thereby every path has a corresponding label which consists of the information of the cost in regard of risk, C , and the distance traveled, from now referred to as resource weight, W . Furthermore no labels on a point has the same cost, and if multiple labels are present on a point, then for each label any other label on that point must be different. These ideas are described more thoroughly below, but first (W_i^k, C_i^k) is defined as the label of the path, p_i^k , where i is a point in the graph, and $k \in I_i$, where I_i is the index set of labels on point i . Furthermore, it is taken that all paths are starting from the origin, o , (Dumitrescu and Boland, 2001), and (Desrosiers et al., 1995).

Definition 3.1

Domination. (Dumitrescu and Boland, 2001)

Let (W_i^k, C_i^k) and (W_i^l, C_i^l) be two labels on an arbitrary point i .

It is then said that (W_i^k, C_i^k) *dominates* (W_i^l, C_i^l) iff $W_i^k \leq W_i^l$, $C_i^k \leq C_i^l$, and the labels are not equal.

Definition 3.2

Efficiency. (Dumitrescu and Boland, 2001)

A label (W_i^k, C_i^k) is said to be efficient if it is not dominated by any other label at point i .

A path is said to be efficient if the corresponding label is efficient.

The Label Setting Algorithm finds all efficient labels on every point. At first, no labels are to be found on any point, except $(0, 0)$, at the origin point, o . The algorithm then extends the set of labels, by *treatment* of an existing label on a point, this is done by extending the corresponding path along all outgoing edges of a point, i , denoted $\delta^+(i)$. The algorithm is given in algorithm 1, the implementation is seen in appendix B.1. Here L_i describes the set of labels on point i , further $T_i \in I_i$ is the index of labels on point i which have been treated, furthermore τ_j^w denotes the shortest path from point j to g , in regard of W , the path is found by the Floyd-Warshall algorithm, the algorithm is seen in Appendix A.6. Lastly recall that $G(N,E)$ represents the graph with N being the nodes/points, and E the edges.

Algorithm 1 Label Setting Algorithm (Dumitrescu and Boland, 2001)

0: *Initialisation*
 Set $L_s = \{(0, 0)\}$ and $L_i = \emptyset \forall i \in N \setminus \{o\}$
 Initialise I_i accordingly for every $i \in N$
 Set $T_i = \forall i \in N$

1: *Selecting label*
if $\cup_{i \in N} (I_i \setminus T_i) = \emptyset$ **then** STOP; all efficient labels are generated
else choose $i \in N \wedge k \in I_i \setminus T_i$ such that W_i^k is minimal

2: *Treatment of label*
for all $(i, j) \in \delta^+(i)$ where $W_i^k + w_{ij} + \tau_j^w \leq \bar{D}$ **do**
if $(W_i^k + w_{ij}, C_i^k + c_{ij})$ is not dominated by (W_j^l, C_j^l) for any $l \in I_j$
then Set $L_j = L_j \cup \{W_i^k + w_{ij}, C_i^k + c_{ij}\}$ and
 update I_j accordingly.
 Set $T_i = T_i \cup \{k\}$.
 Go to Step 1.

By application of algorithm 1 each distinct scenario, $u \in U_s$ of the first objective function from equation (1.6) can be solved deterministically. Furthermore the objective function with the expected value of the scenario outcome from equation (3.2) can be solved, by taking the cost on each node to be the sample average of the scenarios, and recall that by the Law of Large Numbers the sample average converges to the expected value. Lastly, the minimax regret objective function from equation (3.4) can be solved by utilisation of algorithm 1. However, solving the minimax regret problem can only be done by solving each distinct scenario deterministically and then comparing their regret by equation (3.3). This would be carried out by finding the optimal path of a specific scenario and then finding the risk of said path in the other scenarios, thereby the regret for that path in each scenario can be found by comparison to the optimal path of that scenario. Nevertheless running the algorithm for all scenarios makes an extensive search for the robust path, in regard to the minimax regret. Therefore alterations are made to the label selection, from step 1, in algorithm 1 and alterations are made to the definition of domination, which is utilised in step 2 of algorithm 1, the alterations are made such that the algorithm finds the path minimising the maximum regret at once instead of having the need of running algorithm 1 for all distinct scenarios.

The new definition of domination relies on the labels on each point of the graph being modified. To be able to consider all scenarios in the same graph, the label structure is altered, such that it includes the costs of risk in every distinct scenario. That is, a label in the multi-cost graph setup is given by, $(W_i^k, C_{1i}^k, C_{2i}^k, \dots, C_{si}^k)$ with $u \in U_s$ being the index of scenarios, and k continues to represent the k^{th} label on a given point i . By utilising the new label structure every path contains the cost of risk in every distinct scenario, thus some part of the algorithm needs to be updated. First, the choice of which label is to be treated is changed, such that it is adjusted to the minimax regret optimisation. This is carried out by choosing the new label to treat in regard of the minimax regret compared to the previous label. This is elaborated further in definition 3.3.

Definition 3.3**Label Selection (minimax regret).**

Let $(C_{1h}^k, C_{2h}^k, \dots, C_{sh}^k)$ be the label treated in the previous iteration denoted l_h^* and $l_i^k = (C_{1i}^k, C_{2i}^k, \dots, C_{si}^k)$ with $i \in N \wedge k \in I_i \setminus T_i$, be the set of labels it is possible to treat in this iteration.

The label chosen is then the label with minimax regret in regard of the previous label, by equation (3.5).

$$l_i^* = \min_{i \in N \wedge k \in I_i \setminus T_i} ML \quad (3.5)$$

where ML is

$$ML = \max_{u \in U_s} l_i - l_h^*$$

Note that the distance of the path represented as W in the labels is not a part of the minimax regret comparisons, but is still a part of every label.

By utilising the new way of choosing the next label to treat, it is a possibility that the algorithm tends towards a path containing higher risk in general, but still satisfies the label selection. To accommodate for this the definition of domination is changed in definition 3.4.

Definition 3.4**Domination Multiple Scenarios.**

Let $(W_i^k, C_{1i}^k, C_{2i}^k, \dots, C_{si}^k)$ and $(W_i^l, C_{1i}^l, C_{2i}^l, \dots, C_{si}^l)$ be two distinct labels on point i .

It is then said that $(W_i^k, C_{1i}^k, C_{2i}^k, \dots, C_{si}^k)$ *dominates* $(W_i^l, C_{1i}^l, C_{2i}^l, \dots, C_{si}^l)$ iff $W_i^k \leq W_i^l$, $C_{ui}^k \leq C_{ui}^l$, $u \in U_s$ and at least one of the inequalities is strict.

By utilising definition 3.3 and definition 3.4 in the label setting algorithm, it is possible to find the minimax regret path without the requirement of running the algorithm for each distinct scenario, $u \in U_s$. The implementations of the minimax regret setup of the label setting algorithm are seen in appendix B.1. Furthermore, it is seen in the experiments, in Chapter 4, that the setup on some occasions finds a path minimising the maximum regret further than what is possible from the optimal paths for each distinct scenario.

CHAPTER 4

Experiments

In this chapter, the various outcomes of the path planning in regard to the objective functions equation (3.2) and equation (3.4) are depicted. The outcomes are organised as a sensitivity analysis, of how often which outcomes occur, further examples of various outcomes are illustrated and analysed.

Throughout the various experiments, 3 different scenarios are considered, $u \in U_s$, where $s = 3$. Here $\xi_1 = 0$, $\xi_2 = 1$, and $\xi_3 = 2$, these scenarios are referred to as low, medium and high uncertainty scenarios, respectively. The probability of each scenario occurring is flat, $\frac{1}{s}$ for each. This could be altered if the movement of objects in an area of avoidance were analysed. Nonetheless, such analysis is deemed out of scope for this thesis, as the areas represent all sorts of objects e.g., animals and threats. Further, it is taken that all areas of avoidance in each PRM exhibit the same uncertainty, thereby each scenario induces the same amount of uncertainty to all areas of avoidance within each PRM.

The experiments are carried out such that all scenarios are solved deterministically, the optimal path for each scenario is denoted 1^* , 2^* , and 3^* , representing the low, medium, and high uncertainty scenarios, respectively. Furthermore the optimal minimax regret path is found by applying definition 3.3 and definition 3.4 in algorithm 1, this path is from now referred to as "Minimax". Lastly, the optimal sample average path is found by letting the cost of traveling to an arbitrary point have the sample average cost, instead of the cost of a specific scenario, this path is referred to as "Sample Average".

All experiments are carried out with a map a size 20×20 . The grid is still 40×40 . The distance threshold, \bar{D} is set to 35. More or less arbitrarily chosen, but nonetheless, the threshold assures that the paths cannot hold on to the outer boundaries of the map.

4.1 Sensitivity Analysis

The sensitivity analysis is constructed to examine how the various settings of algorithm 1 performs.

The analysis is composed by generating 100 PRMs, all with five areas of avoidance, the center points of each area of avoidance are drawn randomly, with the restriction that the distance between all center points is at least 4.

In figure 4.1 it is seen that from the 100 PRMs all paths are equal in 75 of the maps, meaning that the placement of the areas of avoidance, in 75 of the randomly drawn PRMs, is such that the optimal path is not changing despite having varying amounts of risk with the various scenarios introduced. Nonetheless, in 25 instances the paths change with the scenarios, which implies that finding a path that is only optimal in one scenario can introduce unnecessary risk to the path utilised by the UAV, therefore it is preferable to have a solution looking into all scenarios and finding a robust path through the PRMs.

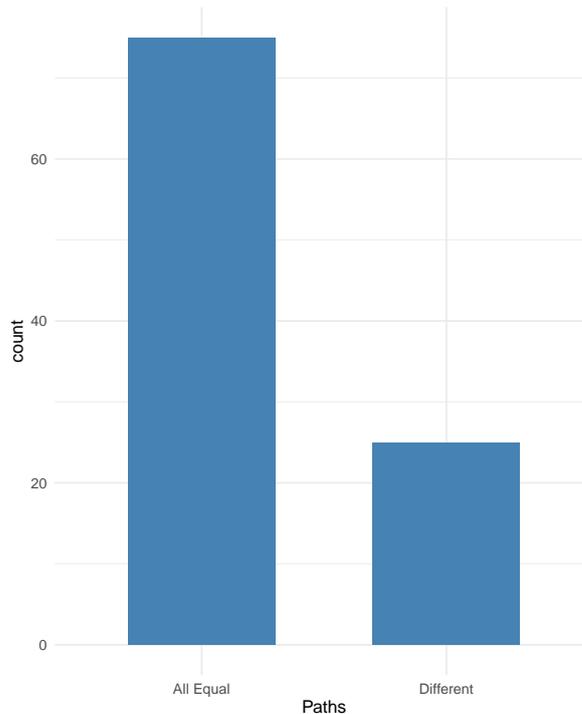


Figure 4.1: Illustration of counts on how often all paths are equal or different.

In the 25 instances, where the optimal path is not the same for each scenario, it is given that one or multiple of the paths are more robust than the rest. This is further examined in the following.

Robust Paths

The robust path is given as the path minimising the maximum regret across all scenarios. When all paths are equal the robust path is trivially found by both the deterministic, sample average, and minimax regret versions of algorithm 1.

By examining the last 25 instances where a robust path is not found trivially, it is however found that the Minimax path is also being the robust path in all instances. A count on how often the 5 various paths turn out to be the robust path is seen in figure 4.2. Here it is seen that the optimal paths from each distinct scenario find a path that turns out to be the robust path, 6, 10, and 11 times, for 1*, 2*, and 3* respectively, the same happens for the Sample Average path 20 times. Note that in some of the instances the robust path is found by multiple of the 5 paths.

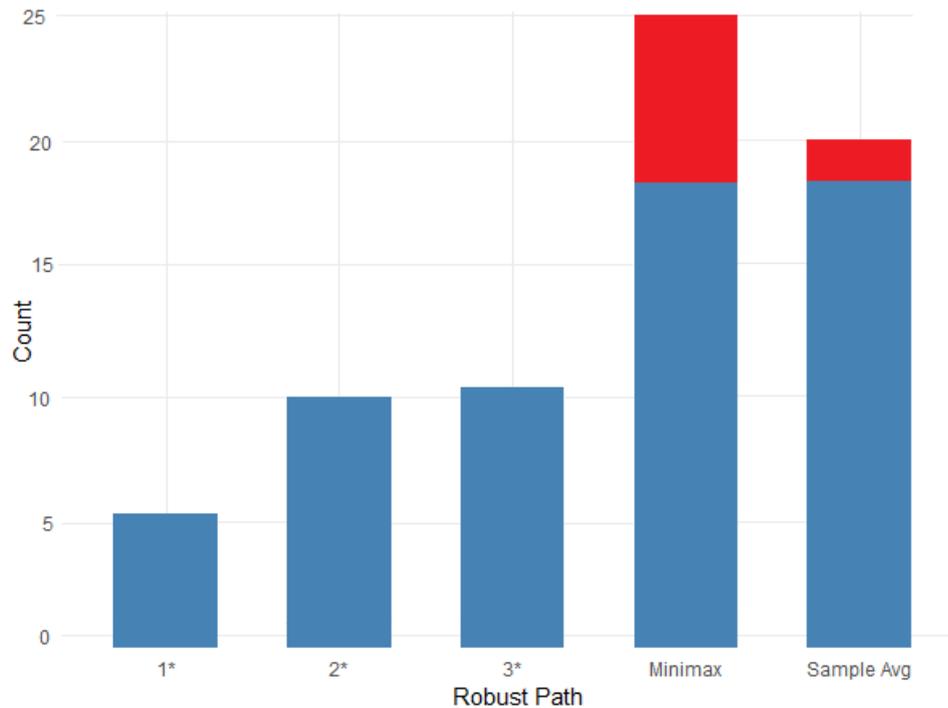


Figure 4.2: Illustration of counts on how often the various paths are also the robust path, in the 25 instances where all paths are not equal. Note that it is a possibility that multiple paths are equal and robust with every PRM.

The Minimax path always finds the robust path, and in 7 of the 25 instances the Minimax path finds a path minimising the regret further than what is possible by the deterministically optimal paths, 1*, 2*, and 3*, the Sample Average path also finds the same robust path as the Minimax path in 2 of these 7 instances but is outperformed in the remaining 5. In figure 4.2 the instances, where the deterministically found paths are outperformed, are represented as the red parts of the bars. Furthermore, the regret of these 7 instances is seen in Appendix table A.5. Lastly, the total regret from all 25 instances where the optimal path is not found trivially by all applied approaches is seen in table 4.1. Here it is evident that the Minimax paths overall outperform the others by having the least regret, further it is seen that 3* exhibits a higher regret than the rest and that the Sample Average paths also are more robust than the deterministic paths in general. From this, it is taken that the Minimax paths is always robust, as the path found by the minimax approach either finds the same path as the most robust path of the three deterministic ones or finds a more robust solution when the given PRMs allows for a higher discrepancy.

| Path | 1* | 2* | 3* | Minimax | Sample Avg |
|--------------|-------|-------|-------|--------------|------------|
| Total Regret | 0.698 | 0.644 | 2.004 | 0.471 | 0.491 |

Table 4.1: The total regret of the various paths, in the 25 instances where the optimal path is not found trivially.

5 Scenarios

For a higher discrepancy in the scenarios, it is investigated what the outcome of letting $s = 5$ such that 5 scenarios are possible instead of 3, with $\xi_1 = 0$, $\xi_2 = 0.5$, $\xi_3 = 1$, $\xi_4 = 1.5$, $\xi_5 = 2$, and the probability of each scenario being flat $\frac{1}{s}$. The same setup with finding the paths being optimal in regard to the various objective functions in 100 randomly drawn PRMs is carried out. Here the minimax regret approach still finds the robust path in all different PRMs, and the sample average path also outperforms the deterministic paths in that regard. Allowing for 5 scenarios does not change the fact that the randomly drawn PRMs in some cases are composed such that the optimal path does not change with the increasing uncertainty induced by the various scenarios. The total regrets in a setting with 5 scenarios are seen in table 4.2, here 27 of the PRMs allowed for different paths with the various scenarios. Resulting in the Minimax path to minimise the maximum regret further than the other paths in 6 of the 27 instances. This is visualised in Appendix figure A.2, with the same setup as for 3 scenarios in figure 4.2.

| Path | 1* | 2* | 3* | 4* | 5* | Minimax | Sample Avg |
|--------------|-------|-------|-------|-------|-------|--------------|------------|
| Total Regret | 0.503 | 0.551 | 0.377 | 0.418 | 0.419 | 0.239 | 0.353 |

Table 4.2: The total regret of the paths in PRMs with 5 scenarios, in the 27 instances where the optimal path is not found trivially.

In the following examples of the different outcomes with 3 scenarios are given, examples of with 5 scenarios would allow for up to 7 different paths, which could become confusing, therefore examples with 3 scenarios are depicted instead. In general, it should be noted that the various examples highlight a problem with the probability of detection, as the paths are found by traversing the map and obtaining the risk for each step made. Thereby a highly detailed path comes from introducing a dense grid, a drawback from this is that every path obtains more risk, as more steps are needed to traverse the map. However, this also affects the probability of detection, which rises with a rise in risk.

All Paths Equal

The first possible outcome is when the PRM setup is given such that the pathing does not change for the various scenarios, thereby the choice of which of any of the objective functions from equation (1.6), equation (3.2), and equation (3.4), to utilise becomes redundant, as they all yield the same path with the same risk. A situation where the distance threshold, \bar{D} and the placement of the areas of avoidance, makes all paths the same is seen in figure 4.3, with uncertainty, $\xi_u = 1$, the path in the remaining scenarios and the related PRMs is seen in Appendix figure A.5. Furthermore, the risk exposure and probability of detection, PD, for each path are given in table 4.3. As all paths are equal a table of the various regrets is omitted.

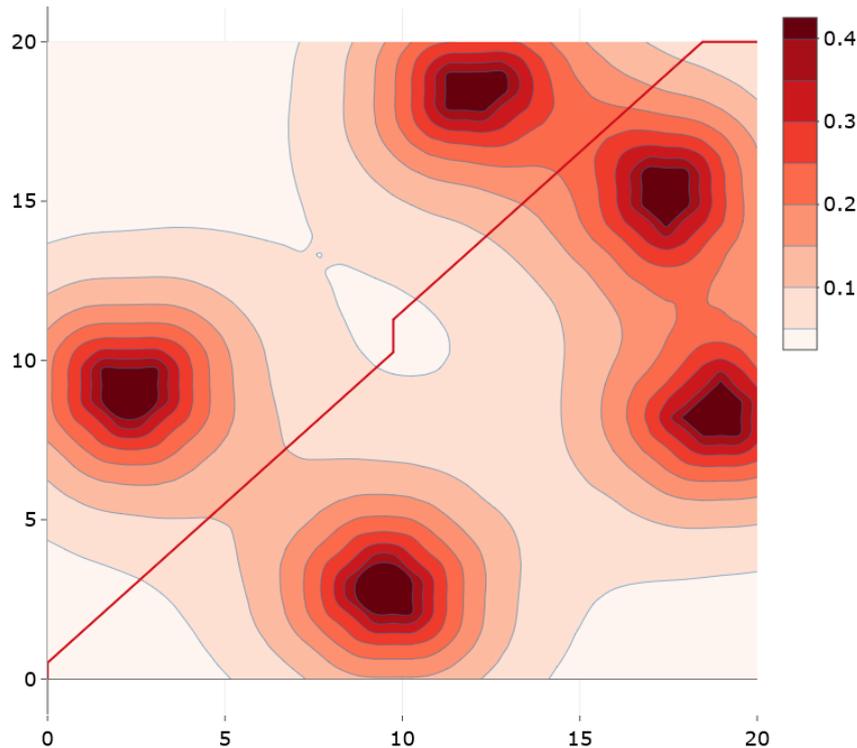


Figure 4.3: Illustration of all paths, as they are equal, in a PRM with areas of avoidance centred at the positions: $(18.8, 8.2)$, $(17.4, 15.3)$, $(2.3, 9.1)$, $(12.0, 18.5)$, and $(9.5, 2.7)$ with $\xi = 1$.

| Path | 1* | | 2* | | 3* | | Minimax | | Sample Avg | |
|---------|-------|-------|-------|-------|-------|-------|---------|-------|------------|-------|
| | R | PD | R | PD | R | PD | R | PD | R | PD |
| ξ_1 | 2.71 | 93.4% | 2.71 | 93.4% | 2.71 | 93.4% | 2.71 | 93.4% | 2.71 | 93.4% |
| ξ_2 | 3.91 | 97.5% | 3.91 | 97.5% | 3.91 | 97.5% | 3.91 | 97.5% | 3.91 | 97.5% |
| ξ_3 | 6.44 | 99.8% | 6.44 | 99.8% | 6.44 | 99.8% | 6.44 | 99.8% | 6.44 | 99.8% |
| Dist | 29.18 | | 29.18 | | 29.18 | | 29.18 | | 29.18 | |

Table 4.3: The risk, R, of the various paths together with the probability of detection, PD for the pathing visualised in figure 4.3.

Different Paths with Different Scenarios

The second possible outcome is when the PRM setup is given such that the path changes with the various scenarios. When the maps are solved deterministically for each distinct scenario, the path might change when the uncertainty rises, further, the amount of total risk within the given PRM rises. This might bias the outcomes of which path is best in regard of the minimax regret if only a comparison between the optimal path for each distinct scenario is made, that is since the overall amount risk it is possible for the algorithm to take into consideration is higher in a scenario with higher uncertainty. On the other hand, the optimal path in a scenario with high uncertainty could be completely different compared to a pathing with low uncertainty. Thus it might not always be the case that

the path of a high uncertainty scenario is outperforming the path of a low uncertainty scenario in regard of minimising the maximum regret, this is evident from the total regrets in table 4.1 where 3^* has more regret than the other paths.

The frameworks depicted in the following, are frameworks highlighting how a path can change as the uncertainty rises, and thus why it is an advantage to look into multiple scenarios at once. Lastly, a setup is visualised where the Minimax path is outperforming the others.

High Uncertainty Affects Path

An example from one of the PRMs from the sensitivity analysis is visualised in figure 4.4, where the optimal path from the high uncertainty scenario is very different from the other paths. The path is optimal in the high uncertainty scenario, but in the other scenarios, the risk acquired by 3^* is very high compared to the rest. This is evident from the risk and PD of all paths in table 4.4, and also from the regret of the various paths from table 4.5. This example highlights the downside of only looking into a worst-case scenario, as a lot of the regret obtained by 3^* is acquired in 1 of the 100 PRMs. Furthermore, this highlights that taking all scenarios into consideration when path planning is advantageous. The paths in the other two scenarios are visualised in Appendix figure A.6.

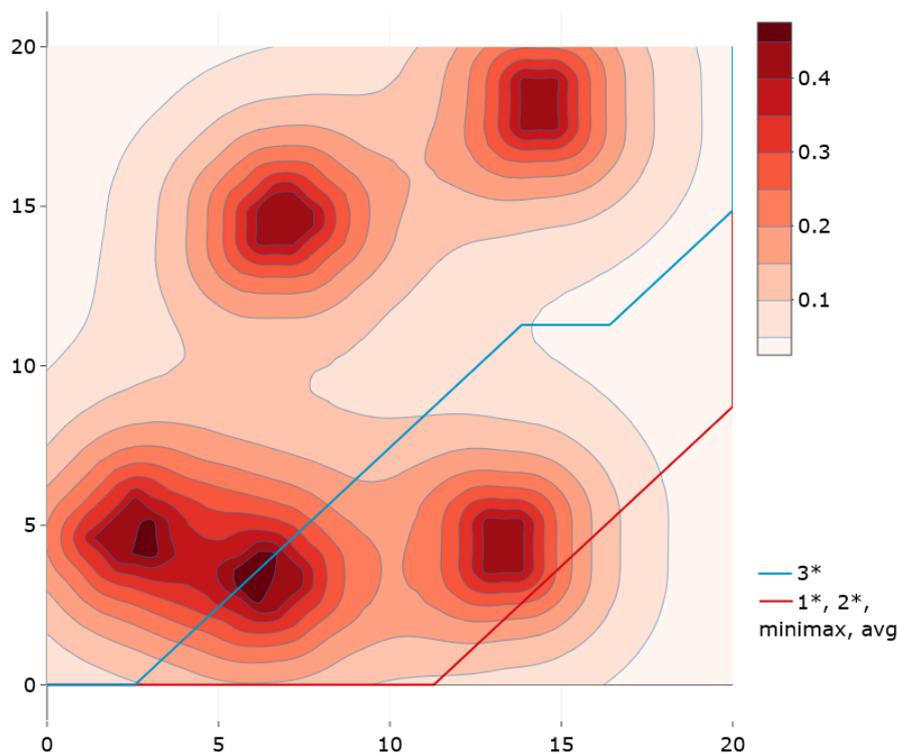


Figure 4.4: Illustration of all paths, where the red path represents both $1^*, 2^*$, minimax, and the sample avg path. The blue path represents 3^* . Both paths in a PRM with areas of avoidance centred at the positions: $(2.5, 4.5)$, $(6.4, 3.4)$, $(7, 14.5)$, $(13.3, 4.4)$, and $(14.4, 18.2)$ with $\xi = 1$.

| Path | 1* | | 2* | | 3* | | Minimax | | Sample Avg | |
|---------|-------|-------|-------|-------|-------|-------|---------|-------|------------|-------|
| | R | PD | R | PD | R | PD | R | PD | R | PD |
| ξ_1 | 3.45 | 96.8% | 3.45 | 96.8% | 3.94 | 98.1% | 3.45 | 96.8% | 3.45 | 96.8% |
| ξ_2 | 5.09 | 99.3% | 5.09 | 99.3% | 5.54 | 99.6% | 5.09 | 99.3% | 5.09 | 99.3% |
| ξ_3 | 8.34 | 99.9% | 8.34 | 99.9% | 8.20 | 99.9% | 8.34 | 99.9% | 8.34 | 99.9% |
| Dist | 34.89 | | 34.89 | | 31.28 | | 34.89 | | 34.89 | |

Table 4.4: The risk, R, of the various paths together with the probability of detection, PD for the pathing visualised in figure 4.4.

| Path | 1* | 2* | 3* | Minimax | Sample Avg |
|---------|-------------|-------------|------|-------------|-------------|
| ξ_1 | 0.00 | 0.00 | 0.49 | 0.00 | 0.00 |
| ξ_2 | 0.00 | 0.00 | 0.45 | 0.00 | 0.00 |
| ξ_3 | 0.14 | 0.14 | 0.00 | 0.14 | 0.14 |
| Max | 0.14 | 0.14 | 0.49 | 0.14 | 0.14 |

Table 4.5: The regret of the various paths visualised in figure 4.4 with risk from table 4.4.

Deterministic Path Robust

In figure 4.5 an example, which is not a part of the sensitivity analysis, is visualised. Here the optimal path of the high uncertainty scenario is also the robust path, together with the Minimax and Sample Average Path. The risk and PD of the various paths are seen in table 4.6 and the regret of the paths are seen in table 4.7. Here it is noted that the high uncertainty makes the path change such that it is no longer preferable to hold on to the boundary of the map. The paths in the other scenarios are visualised in Appendix figure A.9. Despite the two paths being very different the risk across the various scenarios is still close, this is evident from table 4.7. Furthermore, examples where the paths 1* and 2* turns out to be robust, and where none of the paths exhibits the tendency to stay on the boundary of the map, are given in Appendix A.3.

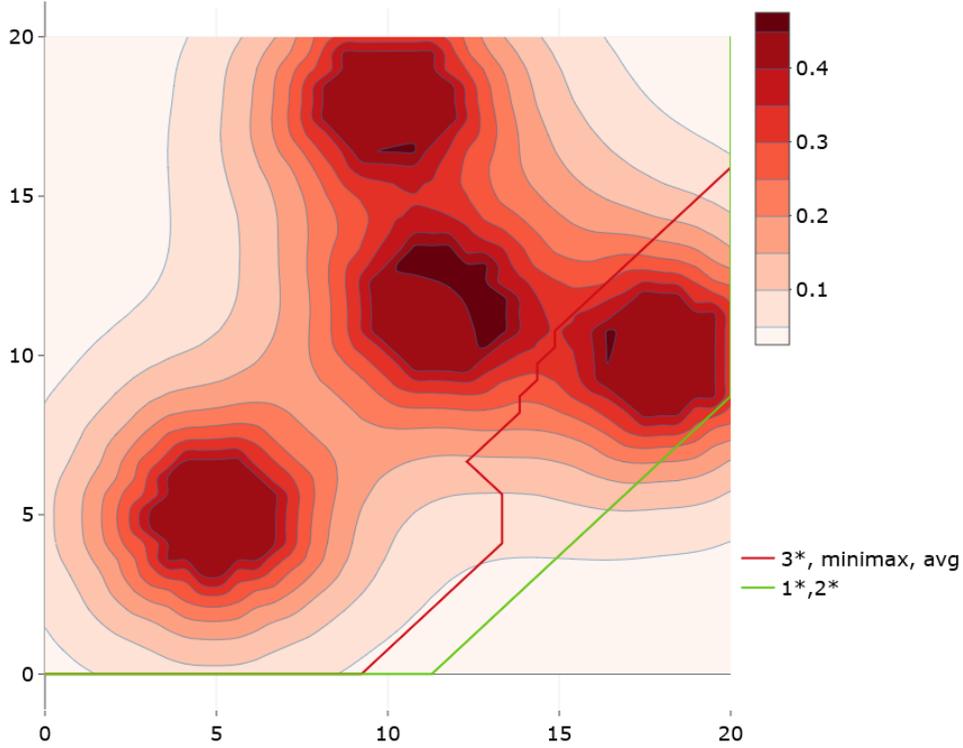


Figure 4.5: Illustration of all paths, where the red path represents both 3^* , minimax, and the sample avg path. The green path represents 1^* and 2^* . Both paths in a PRM with areas of avoidance centred at the positions: (5, 5), (11.5, 11.5), (10, 18), (18, 10), and (9.5, 2.7) with $\xi = 2$.

| Path | 1^* | | 2^* | | 3^* | | Minimax | | Sample Avg | |
|---------|-------|-------|-------|-------|-------|-------|---------|-------|------------|-------|
| | R | PD | R | PD | R | PD | R | PD | R | PD |
| ξ_1 | 2.70 | 93.2% | 2.70 | 93.2% | 2.71 | 93.4% | 2.71 | 93.4% | 2.71 | 93.4% |
| ξ_2 | 3.99 | 97.7% | 3.99 | 97.7% | 4.03 | 98.2% | 4.03 | 98.2% | 4.03 | 98.2% |
| ξ_3 | 6.58 | 99.8% | 6.58 | 99.8% | 6.49 | 99.8% | 6.49 | 99.8% | 6.49 | 99.8% |
| Dist | 34.89 | | 34.89 | | 34.54 | | 34.54 | | 34.54 | |

Table 4.6: The risk, R, of the various paths together with the probability of detection, PD for the pathing visualised in figure 4.5.

| Path | 1^* | 2^* | 3^* | Minimax | Sample Avg |
|---------|-------|-------|-------------|-------------|-------------|
| ξ_1 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 |
| ξ_2 | 0.00 | 0.00 | 0.04 | 0.04 | 0.04 |
| ξ_3 | 0.09 | 0.09 | 0.00 | 0.00 | 0.00 |
| Max | 0.09 | 0.09 | 0.04 | 0.04 | 0.04 |

Table 4.7: The regret of the various paths visualised in figure 4.5 with risk from table 4.6.

Minimax Regret Path Robust

In figure 4.6 an example is visualised, which is not a part of the sensitivity analysis, where all paths are different. Here the Minimax path is the one reducing the maximum regret the most. This is evident from table 4.9, the risk of all paths are seen in table 4.8, with the related PD. Once again all paths are very close in terms of total risk, nonetheless, the Minimax path slightly outperforms the Sample Average path. Furthermore, it is also noted that the PD is lower than the paths in the other PRMs, which is due to only three areas of avoidance being present and thereby less risk in general. It might be difficult to tell the paths apart, therefore the robust path is wider than the others. Furthermore, as all paths are crossing each other in and out the risk obtained by each path is almost the same. The paths in the other scenarios are seen in Appendix figure A.10.

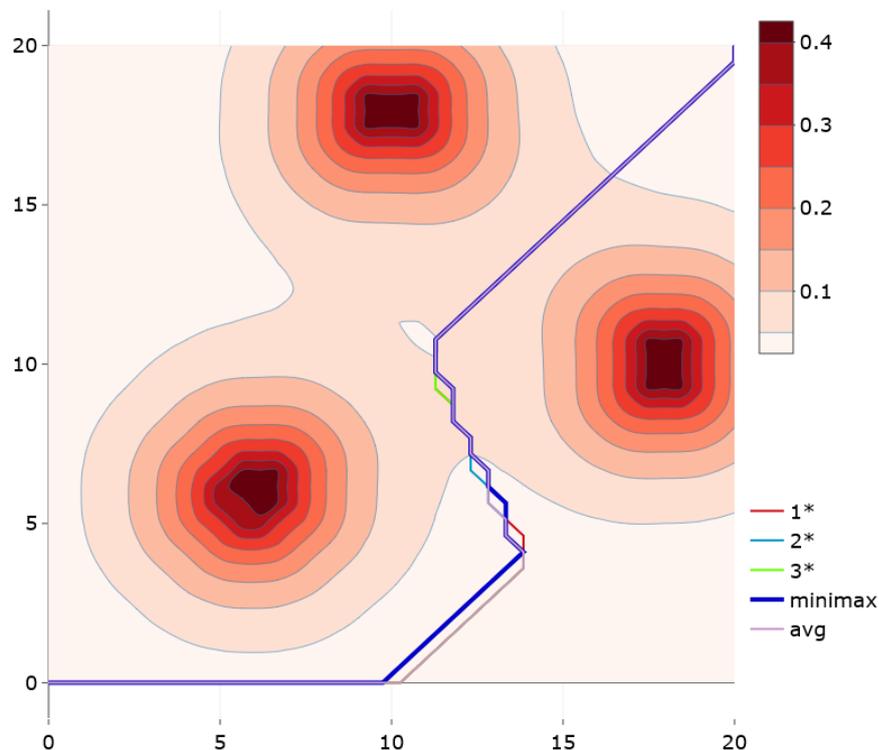


Figure 4.6: Illustration of all paths, where the red path represents 1*, the Turkish blue path represent 2*, the green path represents 3*, the blue path represent minimax, and the dim brown path represents the Sample Avg path. All paths in a PRM with areas of avoidance centred at the positions: (6, 6), (10, 18), (18, 10) with $\xi = 1$.

| Path | 1* | | 2* | | 3* | | Minimax | | Sample Avg | |
|---------|--------|-------|--------|-------|--------|-------|---------|-------|------------|-------|
| | R | PD | R | PD | R | PD | R | PD | R | PD |
| ξ_1 | 1.3561 | 74.2% | 1.3569 | 74.3% | 1.3562 | 74.2% | 1.3565 | 74.2% | 1.3565 | 74.2% |
| ξ_2 | 2.0611 | 87.2% | 2.0588 | 87.2% | 2.0609 | 87.2% | 2.0596 | 87.2% | 2.0599 | 87.2% |
| ξ_3 | 3.2745 | 96.2% | 3.2765 | 96.2% | 3.2744 | 96.2% | 3.2750 | 96.2% | 3.2756 | 96.2% |
| Dist | 34.89 | | 34.54 | | 34.89 | | 34.54 | | 34.89 | |

Table 4.8: The risk, R, of the various paths together with the probability of detection, PD for the pathing visualised in figure 4.6.

| Path | 1* | 2* | 3* | Minimax | Sample Avg |
|---------|--------|--------|--------|---------------|------------|
| ξ_1 | 0.0000 | 0.0008 | 0.0002 | 0.0004 | 0.0004 |
| ξ_2 | 0.0023 | 0.0000 | 0.0022 | 0.0009 | 0.0011 |
| ξ_3 | 0.0001 | 0.0021 | 0.0000 | 0.0005 | 0.0011 |
| Max | 0.0023 | 0.0021 | 0.0022 | 0.0009 | 0.0011 |

Table 4.9: The regret of the various paths visualised in figure 4.6 with risk from table 4.8.

4.2 Time Complexity

In this section, an analysis of time for various grid sizes is carried out. The analysis is executed on an "MSI Leopard Pro GP72VR 7RF" with a 2.8 GHz i7 processor and 8GB DRR4 RAM.

The analysis is deducted on PRMs with areas of avoidance drawn randomly, in similar style to the sensitivity analysis. For all the various densities of the graph the average time spend finding the optimal paths is reported in table 4.10, the average is taken over 50 runs per density. Furthermore if the total time spend on carrying out 50 runs is more than 12 hours, it is taken that the criteria of domination for multiple scenarios from definition 3.4 is too loose, and in these cases no time on completion is reported. In table 4.10 TS is the average time spend in seconds. Further the analysis is carried out under both the deterministic and minimax setting of algorithm 1, for the minimax setting both 3, 5 and 10 scenarios are considered.

| Characteristics | | | Method | | | |
|-----------------|-------|-------|---------------|-----------|-----------|------------|
| | | | Deterministic | Minimax | | |
| K | $ N $ | $ E $ | | $ s = 3$ | $ s = 5$ | $ s = 10$ |
| | | | TS | TS | TS | TS |
| 20 | 400 | 1482 | 3.32 | 6.48 | 10.91 | 20.54 |
| 30 | 900 | 3422 | 18.85 | 43.34 | 78.55 | 153.89 |
| 40 | 1600 | 6162 | 80.54 | 239.48 | 540.11 | - |
| 50 | 2500 | 9702 | 262.2 | 591.66 | - | - |

Table 4.10: Number of nodes and edges, $|N|$ and $|E|$, in each graph, respectively, and the average time spend, TS, in seconds.

The distance threshold is held constant in table 4.10 and therefore it is not evident how this affects the time utilised, nonetheless having a strict distance threshold would limit the possible pathways, and therefore it is assumed that the number of possible paths, and thereby the number of labels for the algorithm to investigate, rises with a rise in \bar{D} . It is evident, from table 4.10, that the time spent to find the robust path, in general, is higher than for the deterministic setting, nonetheless the fact that the minimax setting finds the robust path across all scenarios and not just an optimal path for an arbitrary scenario should be emphasised. Finding a robust path with only utilising the deterministic setup means one would have to run the implementation for each distinct scenario to reassure that the path is at least moderately robust, and even then it might be a possibility to minimise the maximum regret further. However it is also seen in table 4.10 that the algorithm cannot terminate within the given timeframe when the density of the grid is too high combined with an increasing number of scenarios.

CHAPTER 5

Discussion and Perspectives

This thesis has investigated path planning of a UAV traversing a map where certain areas are desired to avoid. Furthermore, uncertainty is introduced to these areas of avoidance. In this chapter the various methods and setups are discussed, suggestions on future work are made, and the setup is put into perspective.

5.1 Discussion

Probability of Detection

As already indicated and visualised in the experiments from chapter 4, the probability of detection is in general very high with all paths displayed. One of the reasons this is evident is since there is a conflict in the desire of getting a detailed path and keeping the risk at a level where the Poisson distribution returns a lower probability of detection than the ones displayed. This could have been altered by normalising the total amount of risk within a PRM, however, another conflict then arises, as introducing additional areas of avoidance would then diminish the risk of the other areas, which also is deemed inappropriate. To make the probability of detection work as desired, a thorough analysis of the reason to avoid a specified area and risk taken by approaching the area is needed.

Uncertainties

The introduction of uncertainties and the transformation of the Gaussian bivariate normal distributions to bivariate Gaussian-tailed uniform distributions also introduce a higher amount of risk in general, as intended. The introduction of further risk affects the paths in some instances, nonetheless, the paths in most PRMs are unaffected by the introduction of uncertainty, thus the necessity of the uncertainties could be questioned. However it is evident from the sensitivity analysis that the path is non-trivial in 25 cases when describing the uncertainty with 3 scenarios, and in 27 cases when utilising a segmentation of uncertainty yielding 5 scenarios, and thus the utilisation of algorithm 1 with the minimax setup becomes relevant.

Probability Risk Map and Paths

The different paths within every PRM are in general obtaining almost the same amount of risk despite the paths being different. Thereby the opportunity of finding a path which minimises the regret further than the optimal paths from each distinct scenario is slim. However, the fact that the paths in some of the randomly sampled instances alter with the various scenarios, and that it in 7 out of 25 instances with 3 scenarios, and 6 out of 27

instances with 5 scenarios, is a possibility to minimise the maximum regret even further than the optimal path for every distinct scenario is notable. Furthermore, the methodology of the minimax regret and the Label Setting Algorithm applies to many other fields than path planning of a UAV.

5.2 Future Work

In this section, suggestions are made on what to elaborate, and investigate further, for the path planning to be flyable and more detailed for the UAV.

Independent Areas of Avoidance

The areas of avoidance in this thesis represent all sorts of objects, and to assume that they all follow the exact same pattern all the time is a limitation towards the possible scenarios that could unfold. Making each area of avoidance independent of each other would depict an environment more true towards the real world. Furthermore having multiple independent areas to avoid within a certain region, means that the number of scenarios for that region would increase massively. As a single change to one of the areas of avoidance would give a new scenario to consider, for the path planning of the entire region desired to traverse. Thereby the total number of scenarios would be a^s , where a is the number of areas to avoid, and s is the number of uncertainty segmentations for the specific area of avoidance. To depict areas of avoidance properly a detailed distribution analysis on the movement of specific objects, within a region desired to traverse, is suggested. Furthermore, this would make scenario reduction advisable. A large number of scenarios and a better understanding of the movement of specific objects could lead to a possible reduction of the number of scenarios while keeping the information on the movement of objects close to intact. By reducing the number of scenarios the representativity of sampling 5 scenarios to investigate with the minimax approach of algorithm 1 would be better, compared to sampling 5 scenarios without performing scenario reduction.

Modelling

To execute the path planning proposed in this thesis to a UAV various factors need to be considered and integrated to the modeling. As mentioned in the limitations, the turning radius and general maneuverability of the UAV are not considered, and therefore need to be addressed, furthermore investigation of minimum step size for a path to be flyable is suggested (Zhan et al., 2014). Lastly, the various additional factors affecting the decision-making process and thereby the path planning should be integrated, this could be factors such as interactions between fuel consumption and weather, or general environment the UAV is pathing through, this should be considered as a part of the preprocessing of the path plan visualised in the flowchart of figure 3.1.

5.3 Perspectives

The setup of the path planning for a UAV in a PRM with uncertainty is relatable to other fields. That is, the desire to make the best choice under uncertainty is eminent in many different problems in operations research. Furthermore, the setup of utilising a graph highlights the decision making, as every step is controlled and an optimal path is found, this is also applicable in other fields. The optimal path through a PRM is one way of showing the decision making which is evident, and in general, is a big part of the utilisation of combinatorial optimisation, other decision-making problems could have been optimal production planning or road transportation of goods (Paschos, 2014).

The ability to minimise the maximum regret with some given scenarios could also be desired in other fields than path planning. That is, e.g., in a situation where an arbitrary company might be forced to consider alterations to their production or supply chain, the certainty of taking a robust decision might be requested. The robust decision is not always the most desired decision to carry out, as the reward is oftentimes less than other decisions. However, if a company is in a situation where confusion to which scenario transpires is apparent, then the robust decision made on the basis of minimising the maximum regret might be desired. A setting where robust decisions are prominent could be with the introduction of the COVID-19 virus, as many normal motives and intentions are changed.

UAVs are also utilised for other tasks than avoiding certain areas, and with alterations, to the objective functions of this thesis, the problem can be transformed into a setting where one wants to obtain the most probability possible, in this thesis seen as risk. However, if the theme of the path planning was *search and rescue* instead of safe path planning, one could utilise the probability map to find a path maximising the probability obtained, and thereby maximising the possibility of finding a missing person (Sebbane, 2018). Furthermore, the uncertainty towards the actual location of, e.g., a threat, as in this thesis, could be altered to be seen as lack of information regarding the location of a missing person in a forest or by the sea.

CHAPTER 6

Conclusion

Based on the content of this thesis it can be concluded that:

The risk of a path plan for a UAV traveling through a region with enemies, endangered species, or other moving objects, can be minimised by utilising a bivariate Gaussian-tailed uniform distribution to describe the areas of avoidance and thereby set up a probability risk map, followed by discretising the map into a grid-based graph. The optimal path of the graph in regard to minimising the risk is thus the optimal path of the UAV. In the event of having uncertainty towards the location of the reason to avoid a specific area, this uncertainty can be taken into consideration by introducing scenarios expanding the areas of avoidance, and thereby increasing the amount of risk in said areas. This modifies the path planning as an optimal path now needs to handle all scenarios well. Therefore a robust path plan is found by application of the changed definition of domination and which labels to address first in the Label Setting Algorithm.

The robust path planning can be obtained by utilising the Label Setting Algorithm with the cost of each distinct scenario being a part of all labels, thereby the need of utilising the algorithm for each distinct scenario can be seen as redundant, as the robust path found by investigating for each distinct scenario is outperformed by the method applied when looking into all scenarios at the same time. This is evident from table 4.1 and table 4.2. When the Label Setting Algorithm looks into multiple scenarios at once, the time it takes to find the robust path is increased compared to the same algorithm in a deterministic setting. Nonetheless, a robust path is found and is deemed a better outcome than a path that might only perform well in one of multiple scenarios. If execution time is the main factor and the computational burden of utilising the minimax approach is deemed too high, then it is recommended to utilise the sample average approach instead of a deterministic setting.

APPENDIX A

Plots, Algorithms and Tables

A.1 Modelling Plot

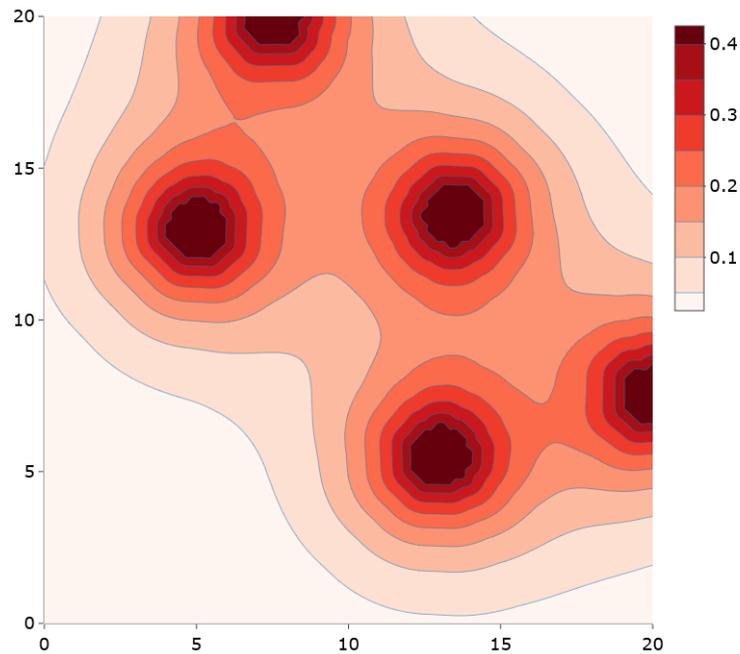


Figure A.1: Illustration of a risk map with areas of avoidance centred at the positions: (13.5, 13.5), (13.0, 5.5), (5.0, 13.0), (20.0, 7.5), and (7.5, 20.0) with $\xi = 1$.

A.2 5 Scenarios Robust Paths

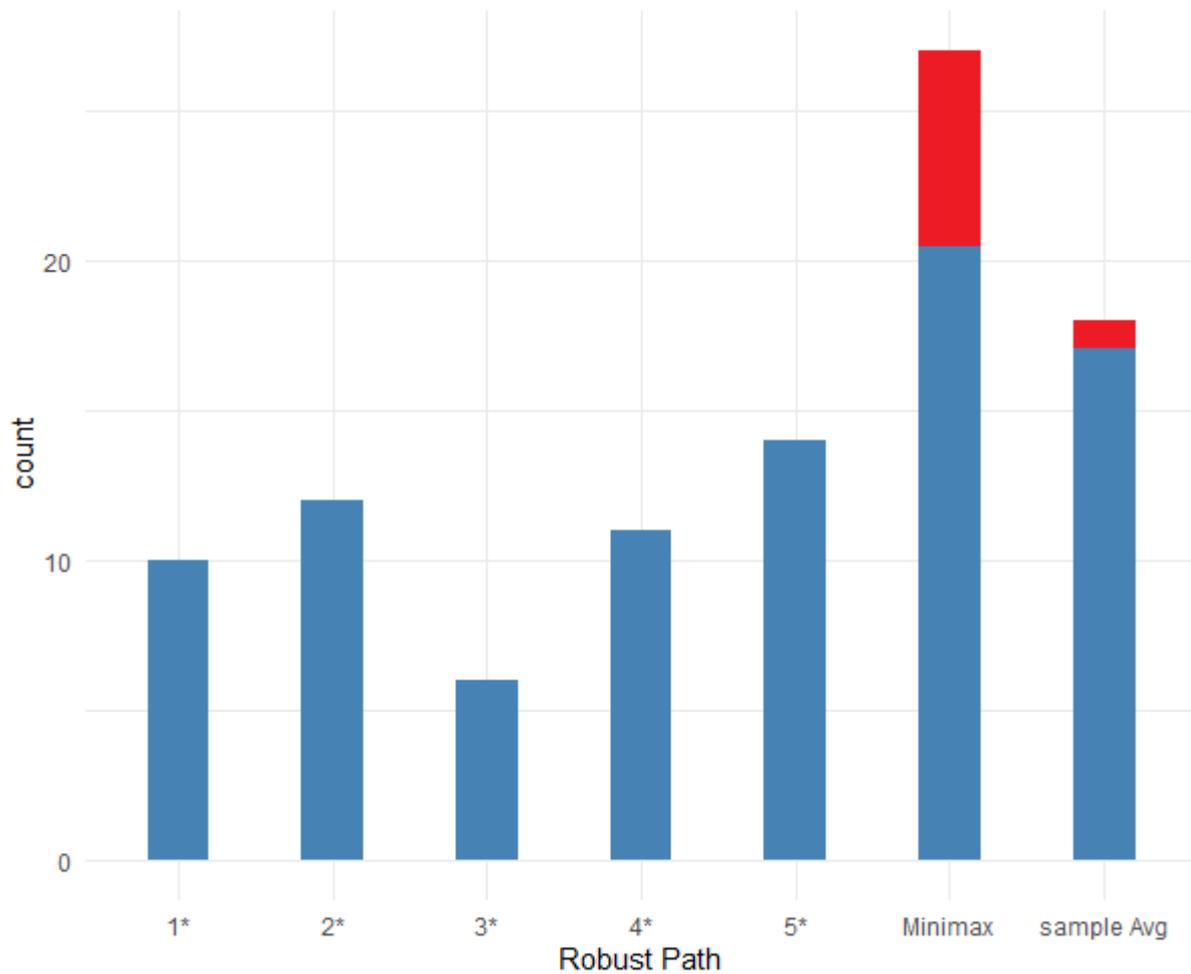


Figure A.2: Illustration of counts on how often the various paths are also the robust path, in the 27 instances where all paths are not equal when looking at 5 different scenarios. Note that it is a possibility that multiple paths are equal and robust with every PRM.

A.3 Robust paths From Deterministic path

Low Uncertainty Scenario Robust

From figure A.3 an example is visualised where the path of the first scenario is also the robust path. The risk and PD is seen in table A.1 and the regret is seen in table A.2. The paths in the other scenarios are seen in figure A.7. As that the paths overlap on most of the map it is a challenge to tell the difference, to compensate for this the robust path is wider than the others. Furthermore as all paths are close to one another the same can be said about the risk of the paths, nonetheless the robust path obtains a path close to the optimal one for each scenario, this is evident from the low regret in table A.2.

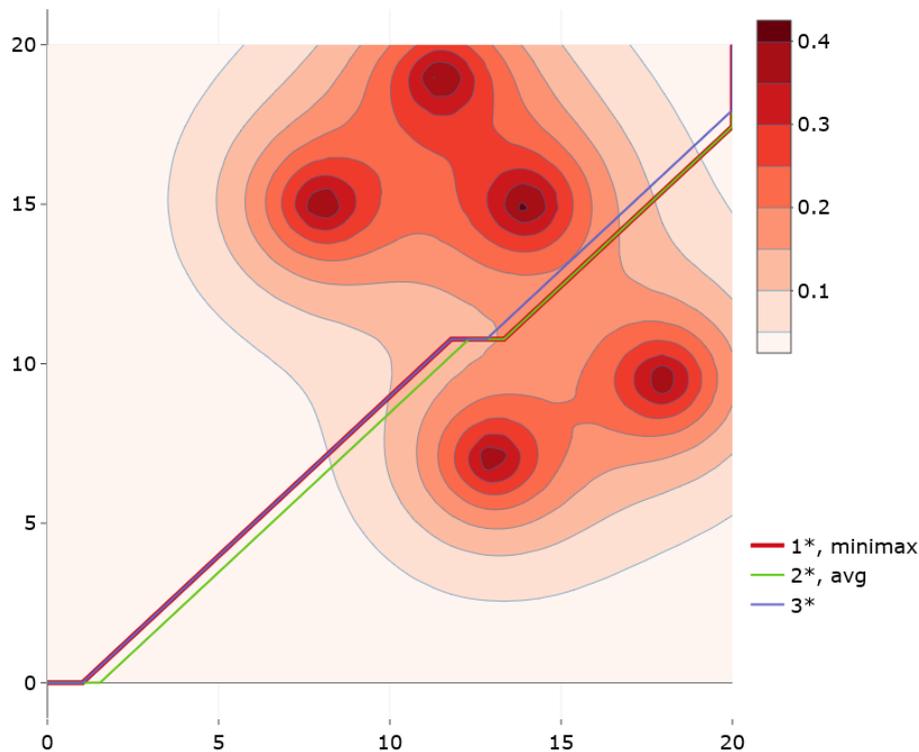


Figure A.3: Illustration of all paths, where the red path represents both 1^* , and Minimax. The green path represents 2^* , and Sample Avg, the blue path represents 3^* . All paths in a PRM with areas of avoidance centred at the positions: $(8, 15)$, $(13, 7)$, $(14, 15)$, $(18, 9.5)$, and $(11.5, 19)$ with $\xi = 0$.

| Path | 1^* | | 2^* | | 3^* | | Minimax | | Sample Avg | |
|---------|-------|-------|-------|-------|-------|-------|---------|-------|------------|-------|
| | R | PD | R | PD | R | PD | R | PD | R | PD |
| ξ_1 | 3.01 | 95.1% | 3.01 | 95.1% | 3.02 | 95.1% | 3.01 | 95.1% | 3.01 | 95.1% |
| ξ_2 | 4.43 | 98.8% | 4.42 | 97.8% | 4.44 | 98.8% | 4.43 | 98.8% | 4.42 | 98.8% |
| ξ_3 | 6.92 | 99.9% | 6.94 | 99.9% | 6.91 | 99.9% | 6.92 | 99.9% | 6.94 | 99.8% |
| Dist | 29.78 | | 29.78 | | 29.48 | | 29.78 | | 29.78 | |

Table A.1: The risk, R, of the various paths together with the probability of detection, PD for the pathing visualised in figure A.3.

| Path | 1* | 2* | 3* | Minimax | Sample Avg |
|---------|---------------|--------|--------|---------------|------------|
| ξ_1 | 0.0000 | 0.0005 | 0.0055 | 0.0000 | 0.0005 |
| ξ_2 | 0.0005 | 0.0000 | 0.0142 | 0.0005 | 0.0000 |
| ξ_3 | 0.0068 | 0.0240 | 0.0000 | 0.0068 | 0.0240 |
| Max | 0.0068 | 0.0240 | 0.0142 | 0.0068 | 0.0240 |

Table A.2: The regret of the various paths visualised in figure A.3 with risk from table A.1.

Medium Uncertainty Scenario Robust

From figure A.4 an example is visualised, here the path of the medium uncertainty scenario is also the robust path together with the minimax path. The risk and PD is seen in table A.3 and the regret is seen in table A.4. The paths in the other scenarios are seen in figure A.8. Note that the robust path is again wider than the other.

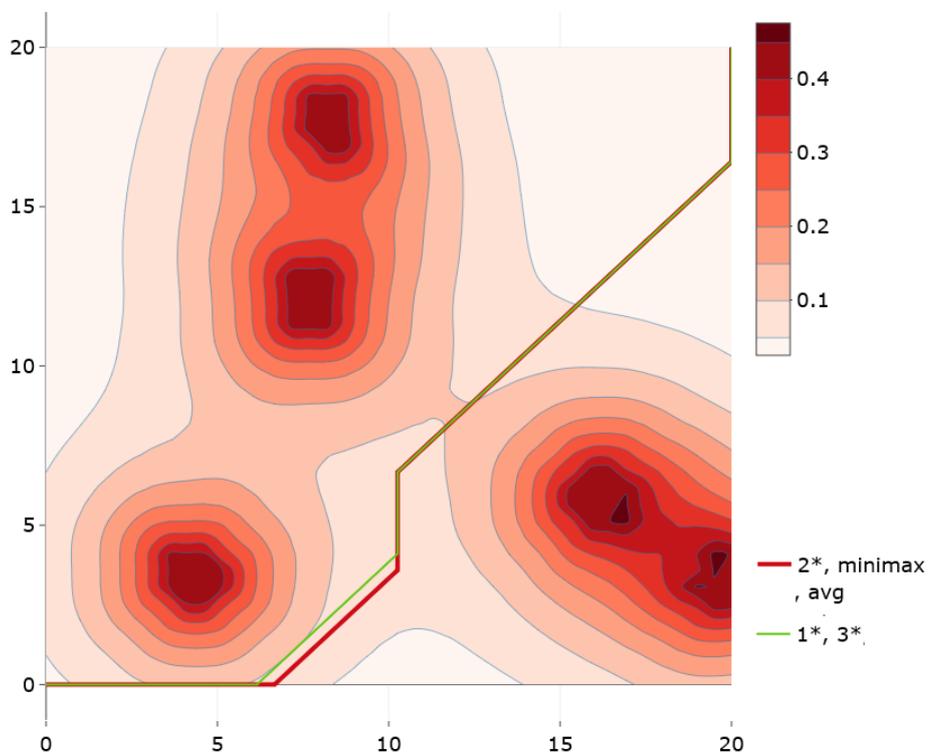


Figure A.4: Illustration of all paths, where the red path represents both 2*, Sample Avg, and Minimax. The green path represents 1*, and 3*, the blue path represents 3*. All paths in a PRM with areas of avoidance centred at the positions: (2.8, 3.9), (15.1, 17.3), (6.1, 11), (18.5, 13.3), and (11, 11) with $\xi = 1$.

| Path | 1* | | 2* | | 3* | | Minimax | | Sample Avg | |
|---------|--------|-------|--------|-------|--------|-------|---------|-------|------------|-------|
| | R | PD | R | PD | R | PD | R | PD | R | PD |
| ξ_1 | 2.1040 | 87.8% | 2.1048 | 87.8% | 2.1040 | 87.8% | 2.1048 | 87.8% | 2.1048 | 87.8% |
| ξ_2 | 3.2125 | 95.9% | 3.2109 | 95.9% | 3.2125 | 95.9% | 3.2109 | 95.9% | 3.2109 | 95.9% |
| ξ_3 | 5.2519 | 99.5% | 5.2523 | 99.5% | 5.2519 | 99.5% | 5.2523 | 99.5% | 5.2523 | 99.5% |
| Dist | 31.88 | | 32.19 | | 31.88 | | 32.19 | | 32.19 | |

Table A.3: The risk, R, of the various paths together with the probability of detection, PD for the pathing visualised in figure A.4.

| Path | 1* | 2* | 3* | Minimax | Sample Avg |
|---------|--------|---------------|--------|---------------|---------------|
| ξ_1 | 0.0000 | 0.0008 | 0.0000 | 0.0008 | 0.0008 |
| ξ_2 | 0.0016 | 0.0000 | 0.0016 | 0.0000 | 0.0000 |
| ξ_3 | 0.0000 | 0.0004 | 0.0000 | 0.0004 | 0.0004 |
| Max | 0.0016 | 0.0008 | 0.0016 | 0.0008 | 0.0008 |

Table A.4: The regret of the various paths visualised in figure A.4 with risk from table A.3.

A.4 Experiments Plots

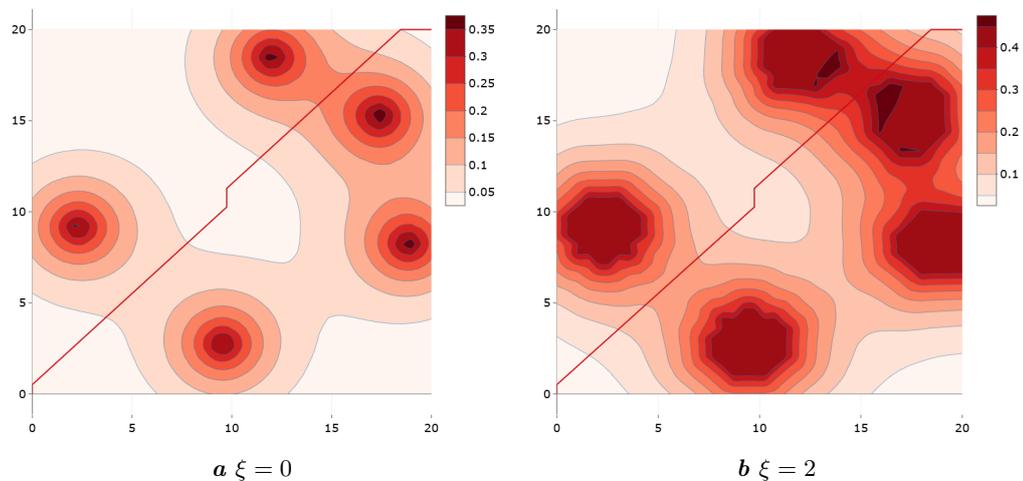


Figure A.5: Illustration of path with areas of avoidance centred at the positions: (18.8, 8.2), (17.4, 15.3), (2.3, 9.1), (12.0, 18.5), and (9.5, 2.7).

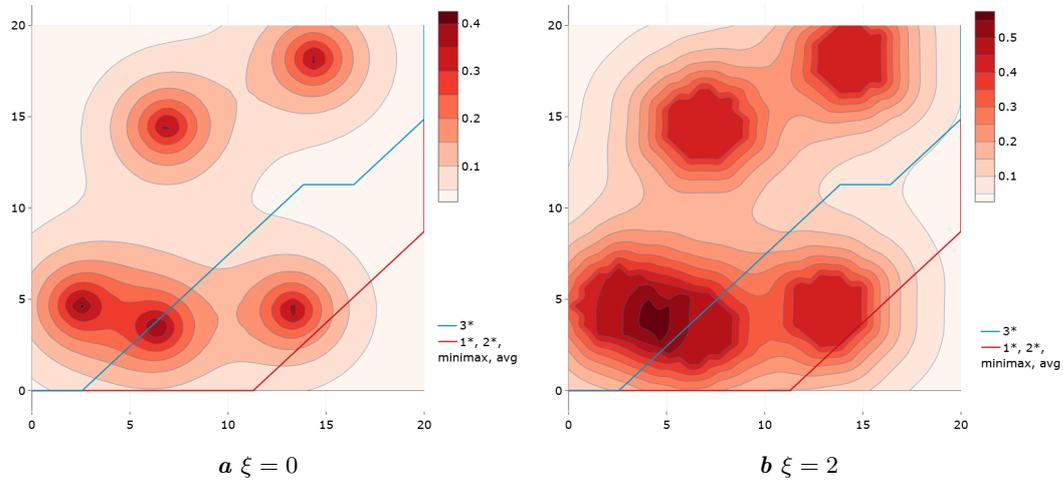


Figure A.6: Illustration of paths with areas of avoidance centred at the positions: (2.5, 4.5), (6.4, 3.4), (7, 14.5), (13.3, 4.4), and (14.4, 18.2).

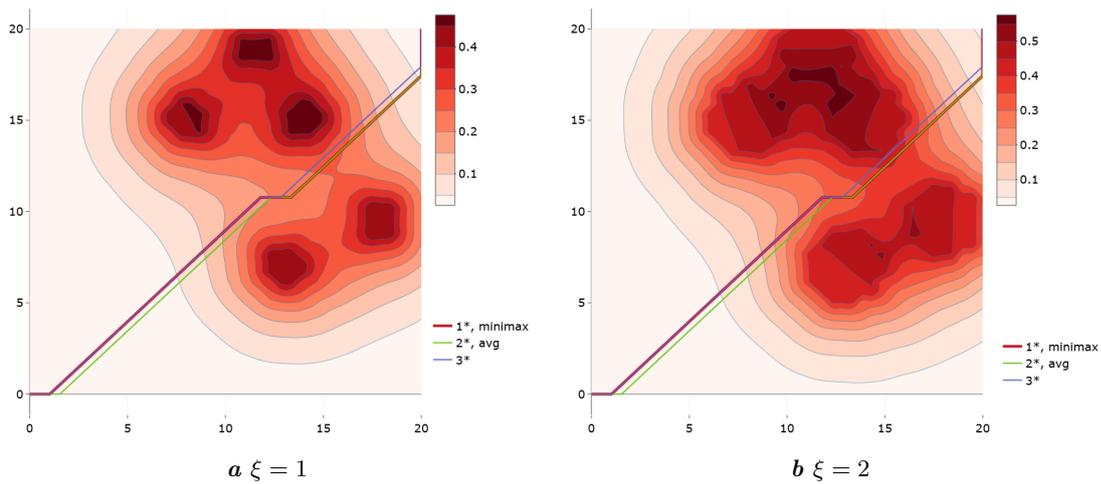


Figure A.7: Illustration of paths with areas of avoidance centred at the positions: (8, 15), (13, 7), (14, 15), (18, 9.5), and (11.5, 19).

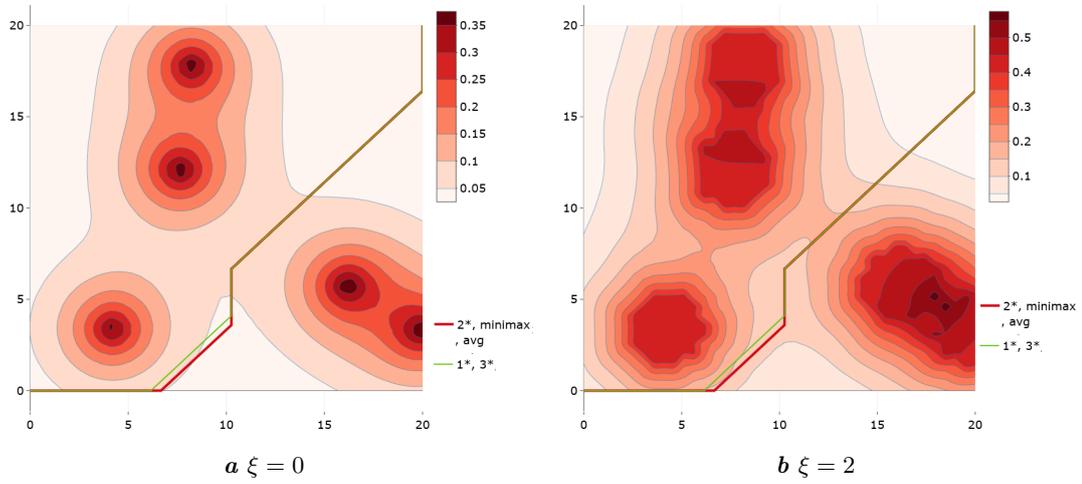


Figure A.8: Illustration of path with areas of avoidance centred at the positions: (2.8, 3.9), (15.1, 17.3), (6.1, 11), (18.5, 13.3), and (11, 11).

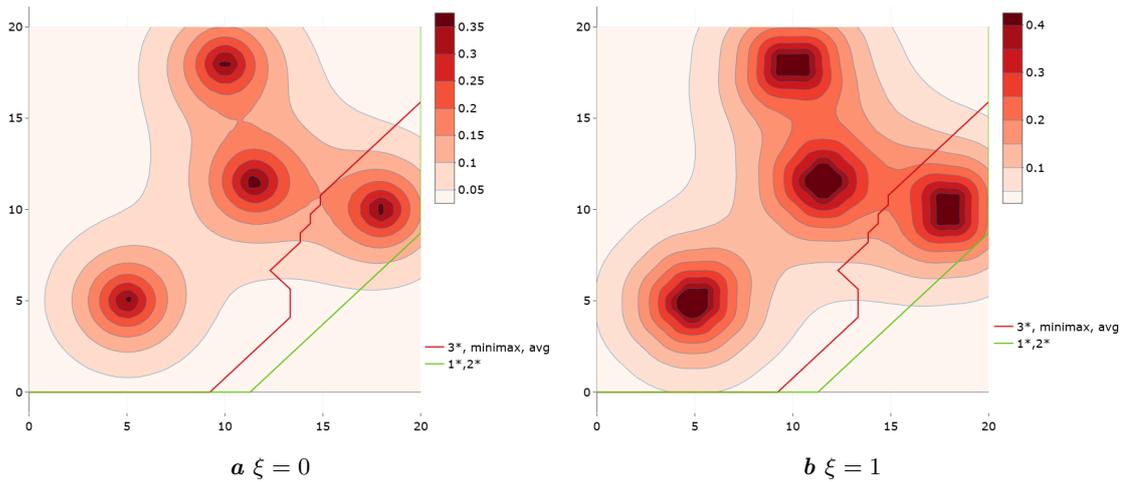


Figure A.9: Illustration of path with areas of avoidance centred at the positions: (5,5), (11.5, 11.5), (10, 18), (18, 10), and (9.5, 2.7).

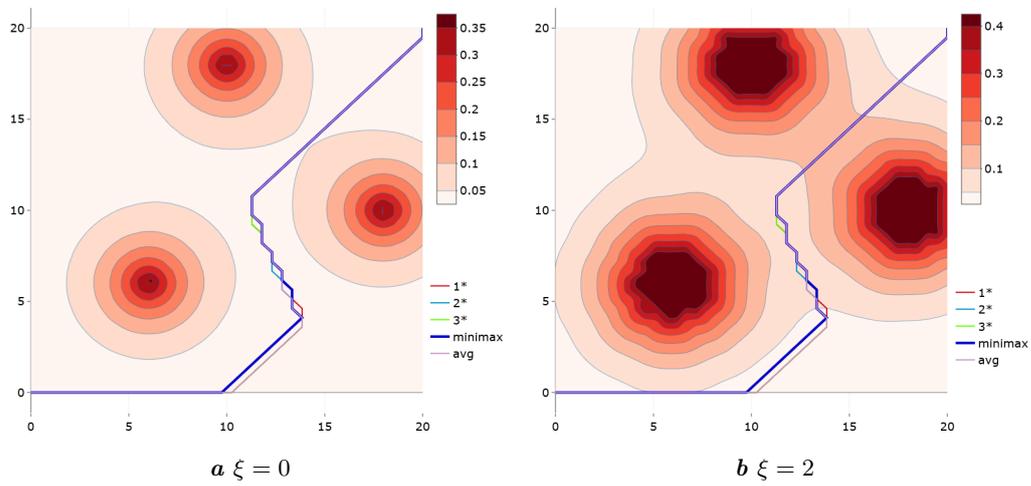


Figure A.10: Illustration of paths with areas of avoidance centred at the positions: (6,6), (10, 18), (18, 10).

A.5 Tables

Tables of regret from the 7 instances the minimax path outperforms the other paths while having 3 different scenarios to consider.

| Paths | 1 | 2 | 3 | Minimax | Sample Avg |
|---------|---------|---------|---------|---------|------------|
| ξ_1 | 0.00000 | 0.00028 | 0.00028 | 0.00002 | 0.00002 |
| ξ_2 | 0.00029 | 0.00000 | 0.00000 | 0.00008 | 0.00008 |
| ξ_3 | 0.00096 | 0.00000 | 0.00000 | 0.00017 | 0.00017 |
| max | 0.00096 | 0.00028 | 0.00028 | 0.00017 | 0.00017 |
| ξ_1 | 0.00000 | 0.00202 | 0.00190 | 0.00114 | 0.00059 |
| ξ_2 | 0.00115 | 0.00000 | 0.00274 | 0.00100 | 0.00097 |
| ξ_3 | 0.00335 | 0.00478 | 0.00000 | 0.00239 | 0.00362 |
| max | 0.00335 | 0.00478 | 0.00274 | 0.00239 | 0.00362 |
| ξ_1 | 0.00000 | 0.00143 | 0.00090 | 0.00014 | 0.00049 |
| ξ_2 | 0.00015 | 0.00000 | 0.00294 | 0.00100 | 0.00005 |
| ξ_3 | 0.00299 | 0.00678 | 0.00000 | 0.00139 | 0.00537 |
| max | 0.00299 | 0.00678 | 0.00294 | 0.00139 | 0.00537 |
| ξ_1 | 0.00000 | 0.01508 | 0.00649 | 0.00195 | 0.01367 |
| ξ_2 | 0.01365 | 0.00000 | 0.01079 | 0.00204 | 0.00131 |
| ξ_3 | 0.00689 | 0.01074 | 0.00000 | 0.00569 | 0.00363 |
| max | 0.01365 | 0.01508 | 0.01079 | 0.00569 | 0.01367 |
| ξ_1 | 0.00000 | 0.00041 | 0.00586 | 0.00019 | 0.00290 |
| ξ_2 | 0.00137 | 0.00000 | 0.00364 | 0.00041 | 0.00187 |
| ξ_3 | 0.00511 | 0.00145 | 0.00000 | 0.00111 | 0.00038 |
| max | 0.00511 | 0.00145 | 0.00586 | 0.00111 | 0.00290 |
| ξ_1 | 0.00000 | 0.00237 | 0.00237 | 0.00079 | 0.00079 |
| ξ_2 | 0.00115 | 0.00000 | 0.00000 | 0.00041 | 0.00041 |
| ξ_3 | 0.00134 | 0.00000 | 0.00000 | 0.00025 | 0.00025 |
| max | 0.00134 | 0.00237 | 0.00237 | 0.00079 | 0.00079 |
| ξ_1 | 0.00000 | 0.00133 | 0.00133 | 0.00055 | 0.000152 |
| ξ_2 | 0.00142 | 0.00000 | 0.00000 | 0.00078 | 0.00098 |
| ξ_3 | 0.00192 | 0.00000 | 0.00000 | 0.00097 | 0.00181 |
| max | 0.00192 | 0.00133 | 0.00133 | 0.00097 | 0.00181 |

Table A.5: Regret for the 7 instances where minimax outperforms the deterministically found paths.

A.6 Floyd-Warshall Shortest Path

This is build upon (Floyd, 1962).

Algorithm 2 Floyd-Warshall Algorithm

```
1:
let dist be a  $K \times K$  matrix of minimum distances at first set to  $\infty$ .
2:
for each edge (i, j) do
    dist [i,j] = w(i, j) The weight of the edge (i, j)
3:
for each point i do
    dist[i,i] = 0
4:
for k from 1 to |K|
    for i from 1 to |K|
        for j from 1 to |K|
            if dist[i,j] > dist[i,k] + dist[k,j] then
                dist[i,j] = dist[i,k] + dist[k,j]
            end if
```

APPENDIX B

R scripts

Implementation of the various parts to find solutions to the objective functions.

B.1 Label Setting Algorithm

Label Setting Algorithm for deterministic solution

```
1 GLA <- function(graphmap, max_dist){
2
3   #early buildup, mapsetup, distance between all pair of nodes etc.
4   g1 <- graphmap[[1]]
5   resource <- graphmap[[2]][,3]
6   g2 <- g1 %>% set.edge.attribute("weight", value = resource)
7   adjmat <- as_adjacency_matrix(g1, attr = "weight") #for probability
8   adjmat2 <- as_adjacency_matrix(g2, attr = "weight") #for distance
9   adj <- as.matrix(adjmat)
10  adj2 <- as.matrix(adjmat2)
11
12  adj[adj == 0] <- NA
13  adj2[adj2 == 0] <- NA
14
15  par_prob <- floyd(adj)
16  par_dist <- floyd(adj2)
17
18  pairs_res_min <- list(par_dist, par_prob)
19
20  node_succ <- lapply(1:grid^2, function(x) neighbors(g1, x, "out"))
21  origin <- 1
22  goal <- grid^2
23
24  rm(adj, adj2, adjmat, adjmat2, par_dist, par_prob)
25  if( origin == goal){ stop("Error: not applicable, origin and goal is at
26    same node")}
27  if(pairs_res_min[[1]][1, goal] > max_dist){stop("Error: No feasible path,
28    distance condition too strict")}
29  #initialisation of labels and indexing ----
30  labels <- rep(list(c()), grid^2)
31  labels[[1]] <- matrix(c(pairs_res_min[[1]][1], pairs_res_min[[2]][1], 0),
32    nrow = 1, ncol = 3)
33  index <- lapply(1:grid^2, FUN = function(x) nrow(labels[[x]]))
34  treated <- rep(list(c()), grid^2)
35  stopping_criteria = 1L
36  tmp_rmv = c()
37  source("dominated_func.R")
38}
```

```

38 while(!is_empty(stopping_criteria)){
39   #finding label to treat -- step 1 in algorithm ----
40   searchable <- which(!compare.list(index, treated))
41   label_search <- mapply(setdiff, index, treated)
42   label_search <- lapply(1:length(label_search), function(x) if(length(
      label_search[[x]]) == 0
43                                     {label_search[[x]] <-
                                         NULL} else{label_
                                         search[[x]]})
44
45   tmp <- suppressWarnings(lapply(1:length(searchable), FUN = function(x)
      min(labels[[searchable[x]]][,1])))
46
47   i <- searchable[which.min(tmp)]
48   multi_labels <- c()
49   k_1 <- c()
50
51   if(nrow(labels[[i]]) == 1 | is_empty(treated[[i]])){k <- which.min(
      labels[[i]][,1])
52   }else if(length(label_search[[i]]) == 1){
53     k <- label_search[[i]]
54   }else{multi_labels <- data.frame(labels[[i]])
55     k_1 <- which.min(cbind(multi_labels[-treated[[i]],1))
56     k <- label_search[[i]][k_1]}
57
58   label_treat <-labels[[i]][k,]
59   # treating label, step 2 in algoithm ----
60   labels_tmp <- which(label_treat[1] + pairs_res_min[[1]][i,node_succ[[i
      ]]] + pairs_res_min[[1]][node_succ[[i]],goal] <= max_dist)
61   labels_acceptable <- as_ids(node_succ[[i]][labels_tmp])
62
63   tmp_dist <- label_treat[1] + pairs_res_min[[1]][i, labels_acceptable]
64   tmp_prob <- label_treat[2] + pairs_res_min[[2]][i, labels_acceptable]
65   tmp_labels <- cbind(tmp_dist, tmp_prob, i)
66   tmp_labels <- unname (tmp_labels)
67
68   # checking for nondominated labels ----
69   dominated <- list()
70   if(length(labels_acceptable) > 0){
71     dominated <- lapply(1:length(labels_acceptable), function(x)
72       is_dominated(tmp_labels[x,], labels[[labels_acceptable[x]]]))
73   }else{
74     dominated[[1]] = TRUE
75   }
76
77   #listing all nondominated labels appropriately ----
78   rmv_add <- lapply(1: length(dominated), function(x) if(!dominated[[x]])
79     {
80       labels[[labels_acceptable[x]]] <- matrix(tmp_labels[x,], nrow = 1,
81         ncol = 3)
82     })
83
84   #adding non dominated labels to list of labels ----
85   for(i in 1:length(rmv_add)){
86     if(dominated[[i]] == FALSE){
87       labels[[labels_acceptable[i]]] <-rbind(labels[[labels_
88         acceptable[i]]],rmv_add[[i]][1,])}

```

```

86     }
87
88     i = searchable[which.min(tmp)]
89     index <- lapply(1:grid^2, FUN = function(x) if(!is_empty(labels[[x]])){
90         seq(1,nrow(labels[[x]]),1)} else{NULL})
91
92     treated[[i]] <- sort(c(treated[[i]],as.numeric(k)))
93
94     tmp_rmv = c(tmp_rmv, i)
95     if(all(compare.list(index, treated))){
96         stopping_criteria <- NULL
97     }
98 }

```

Label Setting Algorithm for multiple scenarios simultaneously.

```

1 GLA_Multi <- function(scenario_maps, max_dist, domination){
2
3     #early buildup, mapsetup, distance between all pair of nodes etc.
4
5     graphmap <- scenario_maps[[1]]
6     g1 <- graphmap[[1]]
7     resource <- graphmap[[2]][,3]
8     g2 <- g1 %>% set.edge.attribute("weight", value = resource)
9     dummy <- length(uncertainties)
10    adjmat <- lapply(1:dummy, function(x) as_adjacency_matrix(scenario_maps[[
11        x]][[1]], attr = "weight")) #for probabilities
12    adjmat2 <- as_adjacency_matrix(g2, attr = "weight") #for distance
13
14    adj <- lapply(1:dummy, function(x) as.matrix(adjmat[[x]]))
15    adj2 <- as.matrix(adjmat2)
16
17    for(i in 1:dummy){adj[[i]][adj[[i]]==0] <- NA}
18    adj2[adj2 == 0] <- NA
19
20    par_prob <- lapply(1:dummy, function(x) floyd(adj[[x]]))
21    par_dist <- floyd(adj2)
22
23    pairs_res_min <- list(par_dist, par_prob)
24
25    node_succ <- lapply(1:grid^2, function(x) neighbors(g1, x, "out"))
26    origin <- 1
27    goal <- grid^2
28    q <- 1L
29
30    #cleaning
31
32    rm(adj,adj2, adjmat, adjmat2, par_prob, par_dist)
33    if( origin == goal){ stop("Error: not applicable, origin and goal is at
34        same node")
35    }
36    if(pairs_res_min[[1]][1, goal] > max_dist){stop("Error: No feasible path,
37        distance condition too strict")
38    }
39    #initialisation of labels and indexing ----
40
41 }

```

```

38 label <- rep(list(c()), grid^2)
39 prob_min <- c()
40 for(i in 1:dummy){prob_min[i] <- pairs_res_min[[2]][[i]][1,1]}
41
42 label[[1]] <- matrix(c(pairs_res_min[[1]][1],prob_min, 0), nrow = 1, ncol
  = 2+length(prob_min))
43 index <- lapply(1:grid^2, FUN = function(x) nrow(label[[x]]))
44 treated <- rep(list(c()),grid^2)
45 stopping_criteria <- 1L
46 tmp_rmv <- c(0)
47
48 while(!is_empty(stopping_criteria)){
49   #finding label to treat -- step 1 in algorithm ----
50
51   searchable <- which(!compare.list(index, treated))
52   label_search <- mapply(setdiff,index,treated)
53   label_search <- lapply(1:length(label_search), function(x) if(length(
  label_search[[x]]) == 0)
54   {label_search[[x]] <- NULL} else{label_search[[x]})}
55
56   tmp <- suppressWarnings(lapply(1:length(searchable), FUN = function(x)
  label[[searchable[x]][,2:(length(uncertainties)+1)]))
57
58
59   i_pos <- searchable[minimax_label(tmp, label, dummy,q)]
60   i <- i_pos[which.min(lapply(1:length(i_pos), function(x) sum(label[[i_
  pos[x]]),2:(1+length(uncertainties))))))]
61   q <- i
62   multi_labels <- c()
63   k_1 <- c()
64   k <- find_k(label, i, label_search, treated)
65
66   label_treat <-label[[i]][k,]
67
68   # treating label, step 2 in algorithm ----
69
70   labels_tmp <- which(label_treat[1] + pairs_res_min[[1]][i,node_succ[[i
  ]]] + pairs_res_min[[1]][node_succ[[i]],goal] <= max_dist)
71   label_acceptable <- as_ids(node_succ[[i]][labels_tmp])
72
73   tmp_dist <- label_treat[1] + pairs_res_min[[1]][i, label_acceptable]
74   tmp_prob <- unlist(lapply(1:dummy, function(x) label_treat[1+x] + pairs
  _res_min[[2]][[x]][i, label_acceptable]))
75
76   if(!is_empty(tmp_prob)){tmp_prob <- matrix(tmp_prob, nrow = length(
  label_acceptable), ncol = length(prob_min))}
77
78   tmp_labels <- cbind(tmp_dist,tmp_prob, i)
79   tmp_labels <- unname (tmp_labels)
80
81   # checking for nondominated labels ----
82
83   dominated <- list()
84   if(length(label_acceptable) > 0){
85     dominated <- lapply(1:length(label_acceptable), function(x)
86     domination(tmp_labels[x,], label[[label_acceptable[x]]]))
87   }else{

```

```

88     dominated[[1]] = TRUE
89   }
90
91   #listing all nondominated labels appropriately ----
92
93   new_label <- lapply(1: length(dominated), function(x) if(!dominated[[x
94     ]]){
95     label[[label_acceptable[x]]] <- matrix(tmp_labels[x,], nrow = 1, ncol
96       = (2+length(uncertainties)))
97   })
98   # adding non dominated labels to list of labels ----
99
100  for(i in 1: length(new_label)){
101    if(dominated[[i]] == FALSE){
102      label[[label_acceptable[i]]] <- rbind(label[[label_acceptable[i]]],
103        new_label[[i]])}
104  }
105
106  index <- lapply(1:grid^2, FUN = function(x) if(!is_empty(label[[x]])){
107    seq(as.numeric(1),nrow(label[[x]]),1)} else{NULL})
108  treated[[q]] <- sort(c(treated[[q]],as.numeric(k)))
109  tmp_rmv = c(tmp_rmv, q)
110
111  if(all(compare.list(index, treated))){
112    stopping_criteria <- NULL
113  }
114 }

```

Domination Criterion and Label Selection

Domination criterion for the single scenario optimisation.

```

1 is_dominated <- function(a,b){
2   if(is.null(b)){label_dominated = FALSE}
3   else{
4     label_dominated <- any(unlist(lapply(1: nrow(b), function(x) ifelse((
5       a[1] == b[x,1]) & (a[2] == b[x,2]),FALSE, TRUE))))
6     label_dominated <- any(unlist(lapply(1: nrow(b), function(x)
7       ifelse(any(a[1:(1+length(uncertainty))] < b[x,1:(1+length(
8         uncertainty)]), FALSE, TRUE))))
9   }
10  return(label_dominated)
11 }

```

Domination criterion for multiple scenarios.

```

1 is_dominated_multiple <- function(a,b){
2   if(is.null(b)){label_dominated = FALSE}
3   else{
4     label_dominated <- any(unlist(lapply(1: nrow(b), function(x) ifelse((a
5       [1] == b[x,1]) & (a[2:(1+length(uncertainties))] == b[x,2:(1+length(
6         uncertainties)]),FALSE, TRUE))))
7     label_dominated <- any(unlist(lapply(1: nrow(b), function(x)
8       ifelse(any(a[1:(1+length(uncertainties))] < b[x,1:(1+length(
9         uncertainties)]), FALSE, TRUE))))

```

```

7   }
8   return(label_dominated)
9 }

```

Parts of the Label Selection for the minimax regret labels.

```

1 minimax_label <- function(tmp, label, dummy, q){
2   df <- as.data.frame(do.call(rbind, tmp))
3   df1 <- as.data.frame(matrix(label[[q]][,2:(1+dummy)], nrow = length(label
4     [[q]])/(dummy+2), byrow = T))
5   min_vals <- as.vector(apply(df1, MARGIN=2, min))
6   df1 <- data.frame(matrix(unlist(lapply(1:nrow(df), function(x) df[x,]-min
7     _vals)), nrow =nrow(df), byrow = T))
8   min <- which.min(apply(df1, MARGIN=1, max))
9   best <- matrix(t(df[min,]), nrow = 1, byrow = T)
10  d <- c()
11  i <- which(unlist(lapply(1:length(tmp), function(x) if(length(tmp[[x]])/
12    dummy == 1){all(best == tmp[[x]])
13  }else{for(i in 1:nrow(tmp[[x]]){d[i] <- all(tmp[[x]][i,] == best)}
14    any(d)})))
15  return(i)
16 }

```

```

1 find_k <- function(label, i, label_search, treated){
2   if(nrow(label[[i]]) == 1 || is_empty(label[[i]])}{k <- 1
3   }else if(length(label_search[[i]]) == 1){k <- label_search[[i]]
4   }else if(is_empty(treated[[i]])){
5     multi_labels <- data.frame(label[[i]])
6     multi_labels <- multi_labels[,2:(1+length(uncertainties))]
7     min_vals_k <- as.vector(apply(multi_labels, MARGIN=2, min))
8     df1_k <- data.frame(matrix(unlist(lapply(1:nrow(multi_labels), function(x)
9       multi_labels[x,]-min_vals_k)), nrow =nrow(multi_labels), byrow = T)
10    )
11    minmax <- which.min(apply(df1_k, MARGIN=1, max))
12    k <- minmax}else{
13     multi_labels <- data.frame(label[[i]])
14     multi_labels <- multi_labels[-treated[[i]],2:(1+length(uncertainties))]
15     min_vals_k <- as.vector(apply(multi_labels, MARGIN=2, min))
16     df1_k <- data.frame(matrix(unlist(lapply(1:nrow(multi_labels), function
17       (x) multi_labels[x,]-min_vals_k)), nrow =nrow(multi_labels), byrow
18       = T))
19     minmax <- which.min(apply(df1_k, MARGIN=1, max))
20     k <- minmax
21   }
22   return(k)
23 }

```

Bibliography

- Alotaibia, Kamil A., Rosenbergerb, M. Jay, Mattinglyc, Stephen P., Punugub, K. Raghavendra, and Visoldilokpund, Siriwat. Unmanned aerial vehicle routing in the presence of threats. *Computers & Industrial Engineering*, 115:190–205, 2018.
- Banderier, Cyril and Schwer, Sylviane. Why Delannoy numbers? *Journal of Statistical Planning and Inference*, 135(1):40–43, 2005.
- Blackmore, L., Hui, Li, and Williams, B. *A probabilistic approach to optimal robust path planning with obstacles*, pages 1–7. IEEE, 2006.
- Blackmore, L., Ono, M., and Williams, B. C. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011.
- Boccardo, Piero, Chiabrando, Filiberto, Dutto, Furio, Giulio Tonolo, Fabio, and LINGUA, Andrea. Uav deployment exercise for mapping purposes: Evaluation of emergency response applications. *Sensors (Basel, Switzerland)*, 15:15717–15737, 2015.
- Bry, Adam and Roy, Nicholas. Rapidly-exploring random belief trees for motion planning under uncertainty. *2011 IEEE International Conference on Robotics and Automation*, pages 723–730, 2011.
- Cassel, David. Remembering shakey, the first intelligent robot. <https://thenewstack.io/remembering-shakey-first-intelligent-robot/>, 2017. Accessed 2020-05-25.
- Desrosiers, Jacques, Dumas, Yvan, Solomon, Marius, and Soumis, François. *Time Constrained Routing and Scheduling*, pages 75–80. Elsevier Science, 01 1995.
- Drawil, N. M. and Amar, O. A. H. M. and Basir. Gps localization accuracy classification: A context-based approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):262–273, 2013.
- Dumitrescu, Irina and Boland, Natashaia. Algorithms for the weight constrained shortest path problem. *International Transactions in Operational Research*, 8(1):15–29, 2001.
- Floyd, Robert W. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.
- Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 100–107, 1968.
- Jouandeau, Nicolas, Touati, Youcef, and Ali Cherif, Arab. Incremental motion planning with las vegas algorithms. In Lazinica, Aleksandar, editor, *New Developments in Robotics Automation and Control*, chapter 16. IntechOpen, 2008.

- Kim, Jongrae and Hespanha, Joao. Discrete approximations to continuous shortest-path: Application to minimum-risk path planning for groups of uavs. In *42nd IEEE International Conference on Decision and Control*, volume 2, pages 1734 – 1740, 2004.
- Kothari, Mangal and Postlethwaite, Ian. A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees. *Journal of Intelligent & Robotic Systems*, 71:1–23, 2013.
- Ladányi, Laszlo, Ralphs, Ted, and Jr, Leslie. Branch, cut, and price: Sequential and parallel. In *Lecture Notes in Computer Science*, volume 2241, pages 223–260, 01 2001.
- Lew, A. and Mauch, H. *Introduction to Dynamic Programming*, pages 3–10. Springer Berlin Heidelberg, 2007.
- Luedtke, Jim. The branch-and-cut algorithm for solving mixed-integer optimization problems, August 2016.
- Masehian, Ellips and Amin-Naseri, M. R. A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning. *J. Field Robotics*, 21:275–300, 2004. doi: 10.1002/rob.20014.
- Miralles, Álvaro and Sanz-Bobi, Miguel. Global path planning in gaussian probabilistic maps. *Journal of Intelligent and Robotic Systems*, 40:89–102, 2004.
- Mulero-Pázmány, Margarita, Jenni-Eiermann, Susanne, Strebel, Nicolas, Sattler, Thomas, Negro, Juan, and Tablado, Zulima. Unmanned aircraft systems as a new source of disturbance for wildlife: A systematic review. *PLOS ONE*, pages 1–4, 2017.
- Paschos, Vangelis Th. *Applications of Combinatorial Optimization*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2014.
- Pfeiffer, Barbara, Batta, Rajan, Klamroth, Kathrin, and Nagi, Rakesh. Path planning for uavs in the presence of threat zones using probabilistic modeling. *IEEE*, 43:1–36, 2007.
- Phung, Manh Duong, Quach, Cong Hoang, Dinh, Tran Hiep, and Ha, Quang. Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection. *Automation in Construction*, 81:25 – 33, 2017. URL <http://www.sciencedirect.com/science/article/pii/S0926580517303825>.
- Pinedo, Michael. *Scheduling. Theory, algorithms, and systems. 4th ed.* Springer, 01 2012.
- Rathbun, David and Capozzi, Brian. *Evolutionary Approaches to Path Planning Through Uncertain Environments*, pages 1–9. AIAA, 2002.
- Righini, Giovanni and Salani, Matteo. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51(3):155–170, 2008.
- Schrijver, Alexander. On the history of combinatorial optimization (till 1960). *Handbooks in Operations Research and Management Science*, 12:31–35, 2001.
- Schrijver, Alexander. On the history of the shortest path problem, 2012.

- Sebbane, Yasmina. *Intelligent autonomy of UAVs : advanced missions and future use*. Chapman & Hall/CRC artificial intelligence and robotics series. CRC Press, Taylor & Francis Group, 2018.
- Shapiro, Alexander and Philpott, A. A tutorial on stochastic programming. pages 1–10, 2007.
- Shapiro, Alexander, Tekaya, Wajdi, Soares, Murilo Pereira, and da Costa, Joari Paulo. Worst-case-expectation approach to optimization under uncertainty. *Operations Research*, 61(6):1435–1449, 2013.
- Souissi, Omar, Benatitallah, R., Duvivier, David, Artiba, Abdelhakim, Belanger, N., and Feyzeau, Pierre. Path planning: A 2013 survey. *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management, IEEE - IESM 2013*, pages 1–8, 2013.
- Szadeczky-Kardoss, E. and Kiss, B. Extension of the rapidly exploring random tree algorithm with key configurations for nonholonomic motion planning. *IEEE International Conference on Mechatronics*, pages 363–368, 2006.
- Toth, C., O'Rourke, J., and Goodman, J. *Handbook of Discrete and Computational Geometry*. Chapman & Hall/crc, 2017.
- Tsardoulis, Emmanouil, Iliakopoulou, Katerina, Kargakos, Andreas, and Petrou, L. A review of global path planning methods for occupancy grid maps regardless of obstacle density. *Journal of Intelligent & Robotic Systems*, 84:830–831, 05 2016.
- Walker, Jon. The self-driving car timeline – predictions from the top 11 global automakers. <https://emerj.com/ai-adoption-timelines/self-driving-car-timeline-themselves-top-11-automakers/>, 2020. Accessed 2020-05-25.
- Wu, Jian, Zhang, Donghao, and Pei, Denghong. Autonomous route planning for uav when threats are uncertain. *IEEE*, pages 19–22, 2014.
- Yang, L., Qi, J., X., Jiang, Song, D., Han, J., and Xiao, J. Guiding attraction based random tree path planning under uncertainty: Dedicate for uav. In *2014 IEEE International Conference on Mechatronics and Automation*, pages 1182–1187, 2014.
- Zhan, Weiwei, Wang, Wei, Chen, Nengcheng, and Wang, Chao. Efficient uav path planning with multiconstraints in a 3d large battlefield environment. *Mathematical Problems in Engineering*, pages 1–5, 2014. ISSN 1024-123X.
- Zhang, Baochang, Mao, Zhili, Liu, Wanquan, and Liu, Jianzhuang. Geometric reinforcement learning for path planning of uavs. *Journal of Intelligent & Robotic Systems*, 77:391–409, 2015.