ABSTRACT

Title of Thesis: RISK-BASED MULTIOBJECTIVE PATH PLANNING AND DESIGN OPTIMIZATION FOR UNMANNED AERIAL VEHICLES

Eliot Rudnick-Cohen, M.S. 2016

Dissertation directed by: Professor Shapour Azarm,
Professor Jeffrey W. Herrmann,
Department of Mechanical Engineering

Abstract: Safe operation of unmanned aerial vehicles (UAVs) over populated areas requires reducing the risk posed by a UAV if it crashed during its operation. We considered several types of UAV risk-based path planning problems and developed techniques for estimating the risk to third parties on the ground. The path planning problem requires making trade-offs between risk and flight time. Four optimization approaches for solving the problem were tested; a network-based approach that used a greedy algorithm to improve the original solution generated the best solutions with the least computational effort. Additionally, an approach for solving a combined design and path planning problems was developed and tested. This approach was extended to solve robust risk-based path planning problem in which uncertainty about wind conditions would affect the risk posed by a UAV.

RISK-BASED MULTIOBJECTIVE PATH PLANNING AND DESIGN OPTIMIZATION FOR UNMANNED AERIAL VEHICLES

by

Eliot Rudnick-Cohen

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2016

Advisory Committee:
Professor Jeffrey W. Herrmann, Chair, Co-Advisor
Professor Shapour Azarm, Co-Advisor
Professor Balakumar Balachandran

TABLE OF CONTENTS:

CHAPTER 1: INTRODUCTION

1.1 MOTIVATION

In recent years interest in UAV operation over public areas has increased significantly, however for larger UAVs operation over public areas can create a risk of injuring people on the ground in the event of an accident. In the event of a crash UAV systems are generally not capable of controlling the location of where they crash, creating a risk of injuring any third parties present at the crash location. The risk associated with this type of incident can be affected by a variety of factors, such as the population density in the region affected by a crash, local weather conditions or the flight speed of the UAV. Ultimately the risk posed by a UAV to third parties on the ground is heavily influenced by the path that the UAV uses to fly through a region, as avoiding densely populated areas will reduce the risk harming third parties in the event of a crash. However, mitigating risk in this manner can increase flight time, which UAV operators generally seek to minimize as much as possible, as paths that mitigate the risk of injuring third parties are likely to take long, indirect routes in order to avoid densely populated areas. Thus a conflict exists between the objectives of mitigating third party risk and minimizing the flight time for a path. In order to effectively determine how a UAV can be safely flown along a path while still taking account preferences in terms of how long it takes to fly those paths, it is thus necessary for UAV operators to be able to explore the trade-offs between these two objectives when determining what the flight path should be for their vehicle.

Adjusting the path taken by an UAV is not the only way in which the risk posed by an UAV can be minimized. The design of both the physical and operational parameters of an UAV can also affect the risk that it ends up posing to third parties on the ground. However, such parameters may not universally reduce the risk being posed, depending on how the population along the path being flown is distributed. It is thus necessary to consider what the optimal parameters for a UAV should be while also considering what the optimal path should be for those parameters. The problem of solving both of these problems simultaneously can be called combined design optimization and path planning. Solving such problems can be useful for determining operation parameters for UAVs such as flight speed, or for adjusting the design of an UAV in order to reduce the risk the UAV will pose on a typical flight path for it. Similar types of combined design optimization and path planning problems can appear in other autonomous system related applications, such as in optimizing manufacturing processes or determining an optimal design and mission plan for other types of robotic systems.

Another issue in determining risk minimal paths for UAVs is that the risk posed by the UAV might be a function of external variables that might not be well known at flight time. For example, the direction that the wind is blowing in can bias a UAV towards crashing in a certain direction, particularly in the presence of wind gusts. Thus a risk minimal path should be the path that has the lowest possible risk given the worst possible set of external variables that affect risk. Such paths would be robust to the external variables, we thus refer to the task of finding such path as a robust risk-based path planning. It can be observed that this problem has a similar structure to the task of combined design and path planning, however instead of aiming to determine the best possible design to minimize an objective the goal is to instead find the worst possible set of external parameters in order to maximize the objective. Since a maximization problem can be converted into a minimization problem by simply inverting the sign of the objective, it should be expected that by approaches developed for solving the design and path planning optimization problem can thus be converted into approaches for solving robust path planning problems.

1.2 RESEARCH QUESTIONS

In this thesis we will discuss methods for performing risk-based path planning for UAVs and how such methods can be integrated into a design process for designing an UAV. To achieve this risk metrics will be discussed for UAVs for quantifying third-party risk. Most risk metrics for UAVs require some representation of the region in which a UAV can crash in relative to its point of failure, methods for determining this will be discussed and compared. Specific questions to be answered include how to these various methods compare against each other in terms of solution quality and performance. Additionally, we will explore whether the use of graph based path planning in a design and path planning optimization context can produce superior results to using waypoint based path planning optimization techniques. Furthermore, we will also explore how approaches for design and path planning optimization problems can be adapted for solving robust risk-based path planning problems. Thus our goals are to answer the following questions:

- How can we quantify third-party risk for a path taken by a UAV?

- What are the best methods for determining paths for UAVs that trade off the risk posed by the UAV and the time needed to traverse the path?
- How can we solve design and path planning optimization problems and what are the benefits to using graph based planning techniques in these problems?
- How can we account for sources of uncertainty in risk-based path planning problems in order to obtain a robust solution?

1.3 PROBLEM DEFINITIONS

The primary problems discussed in this thesis are those of risk-based path planning, combined design optimization and path planning and robust risk-based path planning. Both problems discussed here either are path planning problems or contain a path planning problem as a subproblem, in all cases the problem consists of determining the path between two points that minimizes a cost function. The problem of risk-based path planning is defined a path planning problem where the cost function to be minimized represents the risk to third parties posed by the path being solved for. In addition to path planning, the problem of combined design optimization and path planning also involves a design optimization subproblem. In this thesis, the design optimization problems considered will always be to minimize a cost function by manipulating the values of a set of design variables defined on continuous intervals. The problem of combined design optimization and planning as discussed in this thesis is the problem of simultaneously solving a path planning problem and a design optimization problem where both problems share and can influence a cost function. For the problem of risk-based combined design optimization and path planning considered in this thesis, the cost function used will represent the risk associated with an UAV flying over a specific path. For the problem of robust path planning, we view the problem as finding the path that minimizes the objective under consideration, while that objective is being maximized with respect to a set of variables defined of continuous intervals. Thus the problem of robust risk-based path planning is a robust path planning problem where the objective is the risk associated with an UAV flying over a specific path.

1.4 ORGANIZATION

The chapters of this thesis are organized as follows. Chapter 2 discusses methods for risk-based path planning and defines the risk metric that will be used throughout this thesis. Chapter 3 discusses methods for risk-based combined design optimization and path planning. Chapter 4 discusses methods and results for robust risk-based path planning approaches based off the algorithm developed in Chapter 3. Chapter 5 discusses the conclusions that can be drawn from the work discussed in the previous chapters.

CHAPTER 2: RISK-BASED PATH PLANNING OPTIMIZATION METHODS FOR UAVs OVER INHABITED AREAS

(The material in this chapter originally appeared as [30])

NOMENCLATURE

$c(e)$	cost (weighted sum of the time and risk) of an edge
d	distance between two adjacent points in the discrete probability distribution
D_k	population density of a census tract
$\bar{D}(i, i+1)$	expected crash location population density along a leg
$f(X)$	cost objective function
f_x	fraction (“tolerance”) for the x-coordinates
f_y	fraction (“tolerance”) for the y-coordinates
$G(n_1, n_2)$	edge between nodes n_1 and n_2
K_1	expected number of crashes per 100,000 flight hours
K_2	expected area in which persons will be killed if the vehicle crashes
n	number of waypoints
N	number of whole intervals in a leg
n_x	number of points in a row in the grid
n_y	number of points in a column in the grid
p_{jk}	probability associated with a point in the bivariate distribution
$r(i,i+1)$	risk of flying a leg
$\bar{r}$	normalization constant for risk
$t(i,i+1)$	time to travel a leg
$\bar{t}$	normalization constant for time
V	vehicle airspeed
w_r	weight on risk
w_t	weight on time
$\left(x^S, y^S\right)$	start point of the flight plan
$\left(x^F, y^F\right)$	finish point of the flight plan
(x_i, y_i)	coordinates of a waypoint
x_L, x_U	lower and upper bounds for waypoint x-coordinates
y_L, y_U	lower and upper bounds for waypoints y-coordinates
X	x- and y-coordinates of a list of waypoints
X^N	list of waypoints in solution obtained from network optimization
Δ_x	horizontal distance between adjacent nodes (vertices) in the grid
Δ_y	vertical distance between adjacent nodes (vertices) in the grid
$\left(\Delta x_{jk}, \Delta y_{jk}\right)$	rotated coordinates of a point in the bivariate distribution
Γ_k	census tract polygon

2.1 INTRODUCTION

In the United States, the use of UAVs by government agencies, commercial enterprises, and others requires mitigating the risk to persons on the ground. A UAV operator must demonstrate that the activity poses little risk; that is, the expected number of persons harmed by the activity must be sufficiently small (less than one fatality per ten million flight hours [4]). The risk depends upon the size and reliability of the UAV, the weather conditions, the number of persons who are on the ground close to the path of the UAV, the shelters that protect these persons, and other factors.

Because the risk to persons on the ground depends upon the UAV flight path, UAV operators are interested in approaches (techniques) that can find low-risk flight paths between the start and finish points of the activity. In some areas, the flight paths with the lowest risk are excessively long and indirect because the least-populated areas are too remote. Thus, UAV operators are concerned about the tradeoff between risk and flight time. In some cases, risk acceptance criteria may set an upper bound on the risk; in other cases, UAV fuel capacity or other operational issues may set upper bounds on the time. In general, it is important to find the tradeoffs between these two objectives (risk versus time).

A wide variety of methods exist for solving path planning problems for UAVs [11]. An important distinction to make amongst these methods is between methods that merely find a feasible path (a path that satisfies all constraints present) and methods that find an optimal path (a path that optimizes some objective in addition to satisfying constraints). In the context of risk-based path planning for UAVs, most methods define some form of cost metric to represent the type of risk being minimized and then formulate the problem as a multiobjective optimization problem where the objectives are the risk metric and another metric representing the length of the path (such as distance traversed along the path or time needed to traverse the path). Examples of types of risk considered in such methods include risk posed due to environmental hazards and terrain [21][31][7], risk posed due to large scale obstacles such as radar or heavily populated areas [3][19], the risk of a mid-air collision [34][16][5] or the risks to persons on the ground [19].

In general most methods for solving UAV path planning optimization problems utilize either discrete graph-based planning approaches or mathematical optimization techniques that optimize a fixed number of waypoints. A discussion of methods for solving graph based planning problems with multiple objectives can be found in [26]. Many mathematical optimization techniques for risk-based planning utilize evolutionary optimization algorithms [13] [8].

The risk posed by a UAV to people on the ground can be described in terms of the expected number of fatalities associated with a given flight, which can be determined by identifying the possible crash locations and multiplying the probability of a UAV crash by the number of people present in the potential crash location [4]. Typically this is quantified as a 2-dimensional probability distribution representing the likelihood of crashing at a certain distance away from the point of the failure. For example, Pikaar *et al.* [24] used data about historical crashes at airports to generate a crash location distribution for the specific scenarios of takeoff and landing. For the more general case of a UAV in flight, Wu and Clothier used worst case assumptions to bound the potential crash area [36], which can be used as a distribution with the assumption of a uniform distribution in those bounds. Ford and McEntee [10] generated a bivariate crash location distribution using simple assumptions about the flight dynamics of an unpowered UAV. Lum *et al.* [15] determined a non-uniform distribution of potential crash locations for a particular UAV by performing Monte-Carlo simulations of that UAV failing and crashing to the ground.

The author of this thesis is unaware of any risk-based path planning approach that has considered the distribution of where a UAV will crash and the population density of the areas in and near the flight path. The problem is computationally difficult, and this work considers approaches that can quickly find high-quality solutions. More specifically, this chapter presents a risk-based optimization approach for exploring the tradeoffs between the risk to persons on the ground and flight time and describes the results of a computational study that evaluated the performance of these optimization algorithms for some specific instances. The approach is a novel combination of multiple elements: (1) a flight dynamics model that predicts the crash location for a UAV that loses power at a given altitude and velocity, (2) a Monte Carlo simulation to generate a probability distribution of crash locations, (3) a risk assessment method that incorporates the crash location distribution (not the worst case) and the population density near the flight path (based on census data), (4) an efficient algorithm for finding a flight path that minimizes both time and risk, (5) two different solution improvement techniques, (6) a bi-objective framework for generating a set of non-dominated solutions, and (7) a set quality metric for evaluating and comparing sets of bi-objective solutions.

The rest of the paper is organized as follows. Section 2 formulates the problem, and Section 3 describes the solution approaches. Section 4 presents the design of the experiments that were conducted, and Section 5 discusses the results. Section 6 is the summary and conclusions.

2.2 PROBLEM DEFINITION

Given a start point A, a finish point B, a planned altitude, and the UAV velocity, the objective is to find the UAV's flight plan from A to B to minimize risk and time. In theory, the flight plan can be any continuous path from A to B. However, here, it is treated as a piecewise linear path passing through *n* waypoints (x_i, y_i). The first waypoint is the start point $(x_0, y_0) = \left(x^S, y^S\right)$, and the last waypoint is the end point $(x_{n+1}, y_{n+1}) = \left(x^F, y^F\right)$.

In theory, there are no constraints on the locations of the waypoints. In practice, of course, flight plans must avoid different types of restricted airspace, which are ignored in this study (but these could easily be added as constraints if needed). For computational purposes, locations of the waypoints were restricted to remain within upper and lower bounds on the *x*- and *y*-coordinates, in order to place a limit on the size of the region being considered.

The total time of a flight path is the sum of the time for each leg. In this study, the time $t(i, i+1)$ equals the distance from (x_i, y_i) to (x_{i+1}, y_{i+1}) divided by the vehicle's airspeed *V*.

In this study, the risk measure is the expected number of deaths. The total risk for a flight plan equals the sum of the risk for each leg. The risk measure depended upon the population density at the potential crash locations, which are determined by the flight path. This study did not consider the influence of shelter.

2.3 OPTIMIZATION APPROACHES

The risk-based path planning optimization problem had two stages: (Stage 1) estimate the probability distribution of the crash location based on planned altitude and velocity of the UAV; and (Stage 2) determine the flight paths that minimize time and risk.

Table 2.1 INITIAL CONDITIONS FOR MONTE CARLO SIMULATIONS.

Velocity (m/s)	Mean	Deviation
$\dot{x}$	50	50
$\dot{y}$	0	10
$\dot{z}$	0	10
Position (m)		
x	0	0
y	0	0
z	1,524	0
Orientation, Euler angles (degrees)		
Φ	0	11.25
Θ	0	11.25
Ψ	0	11.25
Angular Velocity (degrees/s)		
P	0	11.25
Q	0	11.25
R	0	11.25
Control surface deflection (degrees)		
Elevator Deflection (δ_E)	0	11.25
Rudder Deflection (δ_R)	0	11.25
Aileron Deflection (δ_A)	0	11.25
Control surface deflection rates (degrees/s)		
Elevator deflection rate ($\dot{\delta}_E$)	0	0
Rudder deflection rate ($\dot{\delta}_R$)	0	0
Aileron deflection rate ($\dot{\delta}_A$)	0	0

To obtain a crash location distribution, a Monte Carlo simulation of a UAV crashing was used to generate sample crash locations. To model a UAV crashing, a dynamics simulation of an unpowered UAV with freely moving control surfaces (unpowered) was implemented using the non-linear ODE models detailed in [33] and [32]. By solving this non-linear ODE numerically using MATLAB's ode45 solver [18] it was thus possible to simulate the trajectory of how a UAV would crash given specific initial conditions. An example of a crash trajectory from this simulation can be seen in Figure 2.. The final crash location of the UAV was determined to be the point at which the UAV had a height (*z*) of zero from the ground, meaning it had hit the ground. By varying the initial conditions of the UAV randomly about a fixed initial state, it was thus possible to simulate a range of possible crash locations by repeatedly running this simulation from those initial conditions. A list of the state variables used in the model, the baseline case, and the distributions of the random perturbations can be found in Table 2.1. The aerodynamic coefficients and physical properties used for the UAV being simulated were based on those provided for a Cessna 182 aircraft [27]. The crash distribution was then used to compute the risk presented to people on the ground. The crash distribution was discretized into a 2-dimensional grid of bins for computational efficiency. A heat map of this discretized distribution can be found in Figure 2.2. In this distribution, the probabilities of landing in the central cells are much greater than those of other cells, but the small cell size (relative to the lengths of the edges and the size of the census tracts) makes the distribution adequate.

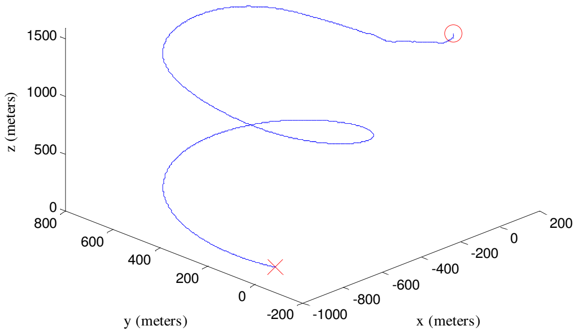


Figure 2.1 EXAMPLE UAV CRASH TRAJECTORY. THE CIRCLE DENOTES THE START POINT AND THE “×” DENOTES THE FINAL CRASH LOCATION.

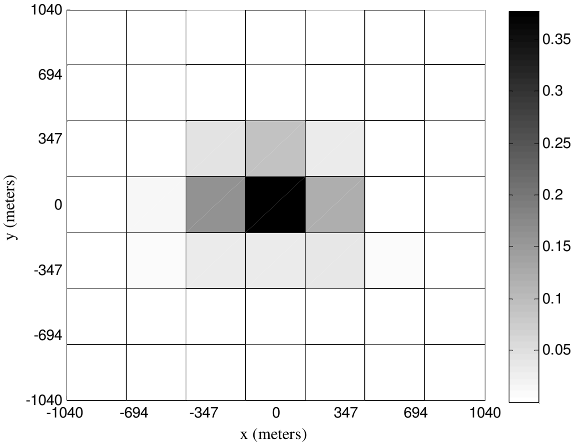


Figure 2.2 DISCRETIZED HEAT MAP OF CRASH DENSITY DISTRIBUTION. THE SCALE CORRESPONDS TO THE PROBABILITY THAT THE VEHICLE WILL LAND IN THAT CELL.

The process of discretizing the crash distribution yielded a two-dimensional discrete probability distribution that specifies, for each discrete point in an *m*-by-*m* grid, the probability that the UAV will land at that spot. By choosing *m* to be an odd number, the center of this discrete probability distribution is the location of the UAV when the failure occurs and it begins to crash.

To compute the risk for a single leg of the flight plan, the risk was sampled at the midpoints of *N* intervals of length *d* along the leg, where *d* was 3 times the length of the bins used to discretize the crash distribution.

Next, the points in the probability distribution are rotated by the bearing along the leg for which the risk is being evaluated. There are *m* rows of points in the bivariate distribution, each with *m* points.

A “cloud” of $(m + N - 1)m$ points is created as follows:

Step 1. For $a = 1, \ldots, m$, do the following:

For $b = 1,\ldots,N$, $\tilde{x}_{ab} = x_b + \Delta x_{a1}$, $\tilde{y}_{ab} = y_b + \Delta y_{a1}$.

For $b = N+1,\ldots,N+m-1$, $\tilde{x}_{ab} = x_N + \Delta x_{a,b-N+1}$, $\tilde{y}_{ab} = y_N + \Delta y_{a,b-N+1}$

Step 2. If $m \le N$, then the probabilities for each point can be determined as follows:

For $b = 1,\ldots,m-1$, $\tilde{p}_{ab} = \dfrac{1}{N}\sum_{k=1}^{b} p_{ak}$.

For $b = m,\ldots,N$, $\tilde{p}_{ab} = \dfrac{1}{N}\sum_{k=1}^{m} p_{ak}$.

For $b = N+1,\ldots,N+m-1$, $\tilde{p}_{ab} = \dfrac{1}{N}\sum_{k=b-N+1}^{m} p_{ak}$.

Step 3. If $m > N$, then the probabilities for each point can be determined as follows:

For $b = 1,\ldots,N-1$, $\tilde{p}_{ab} = \dfrac{1}{N}\sum_{k=1}^{b} p_{ak}$.

For $b = N,\ldots,m$, $\tilde{p}_{ab} = \dfrac{1}{N}\sum_{k=b-N+1}^{b} p_{ak}$.

For $b = m+1,\ldots,N+m-1$, $\tilde{p}_{ab} = \dfrac{1}{N}\sum_{k=b-N+1}^{m} p_{ak}$.

Step 4. Loop over the census tracts. For each census tract *k*, determine which points in the "cloud" are in that tract's polygon Γ_k and, for $(\tilde{x}_{ab}, \tilde{y}_{ab}) \in \Gamma_k$, set $\tilde{d}_{ab} = D_k$. Calculate the likelihood of crashing into census tract *k*:

$$\Pi_k = \sum_{(\tilde{x}_{ab}, \tilde{y}_{ab}) \in \Gamma_k} \tilde{p}_{ab} \tag{2}$$

Step 5. Determine the expected population density along this leg:

$$\bar{D} = \sum_{a=1}^{m} \sum_{b=1}^{N+m-1} \tilde{p}_{ab} \tilde{d}_{ab} = \sum_{k} \Pi_k D_k \tag{3}$$

The risk of flying from (x_i, y_i) to (x_{i+1}, y_{i+1}) can thus be determined as shown in Equation 4.

$$r(i,i+1) = t(i,i+1)\left(\frac{K_1}{100,000}\right)K_2\bar{D}(i,i+1) \tag{4}$$

Multiple optimization approaches were used for Stage 2, which generated the flight paths, but all of them involved the same procedure for calculating $\bar{D}$ for a leg. The approaches used for Stage 2 generated a set of flight paths by solving a set of path-planning problems. Biobjective optimization was performed using a weighting method, in which the overall objective function ("cost") is defined to be the weighted sum of the scaled risk and time objectives, as detailed in Equation 5. Thus two weighting constants are defined, the time weighting constant w_t and the risk weighting constant w_r. The quantities w_t and w_r must be non-negative and satisfy $w_t + w_r = 1$:

$$f(X) = w_t \sum_{i=0}^{n} \frac{t(i,i+1)}{\bar{t}} + w_r \sum_{i=0}^{n} \frac{r(i,i+1)}{\bar{r}} \tag{5}$$

By varying the weights w_t and w_r and minimizing the value of Equation 5 it was possible to generate a set of different flight paths with the optimization approaches discussed in this chapter.

As detailed next, the optimization approaches included network-based approaches and a non-network approach that used only continuous variable optimization methods.

2.3.1 NETWORK OPTIMIZATION APPROACH

The network optimization step created a network with a grid of nodes and the start and finish points, evaluated the time and risk of every edge in the graph, and then found the minimum-cost path from the start to the finish point. The network consisted of a uniformly spaced grid of nodes with horizontal spacing $\Delta_x = (x^U - x^L)/(n_x - 1)$ and vertical spacing $\Delta_y = (y^U - y^L)/(n_y - 1)$ and the points (x^S, y^S) and (x^F, y^F). Nodes outside the census tracts of states being considered in the optimization were deleted. This type of network was chosen for its simplicity, which makes it easy to create.

Each node in the grid was connected with edges going to the eight nodes neighboring it in the grid. In addition, for the points $\left(x^S, y^S\right)$ and $\left(x^F, y^F\right)$, edges were added from each point to the four corners of the grid element that contained that point. A visual representation of this can be seen in Figure 2.3.

Next, the time and risk of each edge (i, j) was determined followed by the calculation of the cost (weighted sum of the time and risk) of an edge:

$$c(G((x_i, y_i), (x_j, y_j))) = w_t t(i, j) / \bar{t} + w_r r(i, j) / \bar{r} \qquad (6)$$

The network optimization approach found the minimal cost path X^N using Dijkstra's algorithm [9]. Changing the values of the weights w_t and w_r required only recalculating the edge costs and optimizing; it was not necessary to build the network and evaluate the time and risk of every edge every time.

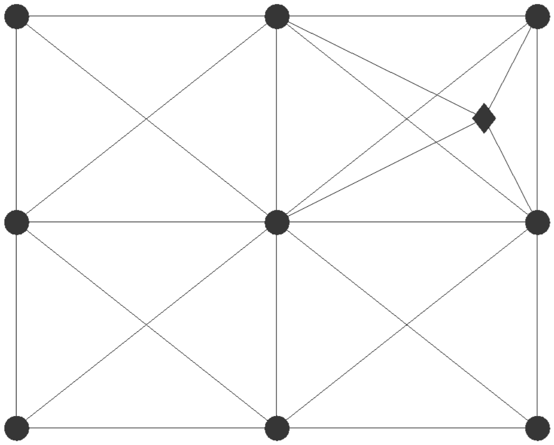


Figure 2.3 IN THIS SECTION OF THE GRID, THE SOLID CIRCLES ARE NODES IN THE GRID, THE DIAMOND IS THE START (OR FINISH POINT), AND THE ADDITIONAL EDGES SHOW HOW THAT POINT IS CONNECTED TO THE NODES IN THE GRID.

2.3.2 LOCAL IMPROVEMENT APPROACH

The local improvement approach used the output of the network optimization step as its initial solution and then found a solution near that solution by solving a continuous variable optimization problem with Equation 5 as its objective function and subject to the additional constraints defined by Equation 7 that kept each waypoint close to a waypoint of the initial solution. The constraints are determined by the tolerances f_x and f_y:

$$\begin{aligned} x_i^N - f_x\Delta_x \le x_i \le x_i^N + f_x\Delta_x \\ y_i^N - f_y\Delta_y \le y_i \le y_i^N + f_y\Delta_y \end{aligned} \tag{7}$$

Pseudocode:

```
$X_0 = X^N$
$X_{sol} = Minimize\left(f(X), X_0, Eq.\ 7\right)$
```

2.3.3 GREEDY IMPROVEMENT APPROACH

The greedy improvement approach also used the output of the network optimization step as its initial solution and then searched for a solution near that solution using a continuous variable optimization method subject to the constraints imposed by Equation 7. However, the greedy improvement approach solved a sequence of *n* subproblems, one for each waypoint in turn. This way each subproblem that was solved had only two variables (the coordinates for one waypoint) which was solved relatively quickly (compared with the time needed to optimize all of the waypoints at the same time). Additionally, since only one waypoint was being optimized at a time, the objective function defined in Equation 5 only needed to be evaluated for two legs: the ones immediately before and after the waypoint being optimized.

Pseudocode:

```
$X_0 = X^N$
$For\ i = 1 \dots n$
        $X_{sol}[i] = Minimize\left(f\left(X[i]\right), X_0, Eq.7\right)$
        $X_0[i] = X_{sol}[i]$
$End$
```

2.3.4 NON-NETWORK APPROACH

The non-network approach did not require the network optimization step because it used a straight-line path between the start and finish points as the initial solution. The number of waypoints was fixed (at 5, 10, 14, or 20), and their coordinates were constrained by the lower and upper bounds (not the nodes of the network). In the initial solution, the waypoints divided the straight-line path into legs with the same distance.

2.4 EXPERIMENTAL DESIGN

Multiple studies were conducted to compare the performance characteristics of the methods described above. In particular, the computational experiments were designed to provide insights into the tradeoffs between the quality of the solutions that were generated and the computational effort required. Throughout these studies two different scenarios were considered, a flight traveling from Patuxent River Naval Air Station, Maryland, to Camp David, Maryland (the “Pax River case”), and a flight traveling from College Park Airport in College Park, Maryland, to Virginia Tech Executive Airport in Blacksburg, Virginia (the “College Park case”).

A set of solutions was generated by solving the problem with different combinations of weights, with w_t = 0, 0.1, 0.2, …, 1.0, and $w_r = 1 - w_t$. For the network optimization step the dimensions of the grid (the number of points in each direction) were varied between several sizes: 30×12, 40×16, 50×20. (For example, the 30×12 grid began with 360 nodes arranged in 30 columns and 12 rows.) Examples of the types of grids used can be seen in Figure 2.4 and **Error! Reference source not found.**. Solutions for the greedy and local improvement approaches were computed for each grid size and for three different values of the tolerance parameters f_x and f_y: 0.25, 0.5 and 0.75 times the size of each grid element. See Table 2.2 for a comparison of sizes. The non-network-based approach was used to generate solutions with 5, 10, 14, and 20 waypoints. MATLAB’s fmincon [17] function was used to solve the continuous optimization problems in the local improvement, greedy improvement, and non-network-based approaches.

Table 2.2 HORIZONTAL AND VERTICAL EDGE LENGTHS FOR DIFFERENT CASES CONSIDERED.

Case	Grid size	Horizontal edge length (°longitude) (Δ_x)	Vertical edge length (°latitude) (Δ_y)
College Park	30x12	0.2989	0.3707
	40x16	0.2223	0.2875
	50x20	0.1769	0.2146
Pax River	30x12	0.3092	0.5414
	40x16	0.2299	0.3970
	50x20	0.1830	0.3134

Each approach generated a set of solutions (one for each value of the weights, see Equation 6). In order to quantify and compare the quality of a set of solutions, a closeness metric based on the method detailed in [35] was developed. To calculate this metric, the time and risk of every solution generated was scaled so that the scaled time and risk of all of the solutions generated for that case ranged from 0 to 1. The metric can be defined as the left handed Riemann sum of the points comprising a Pareto frontier with two additional points added to the frontier at (max objective 1, min objective 2) and (min objective 1, max objective 2) (where the min and max objective function values are relative to all Pareto frontiers being compared), these two additional points represent the worst case values for any regions not covered by the Pareto frontier being evaluated. Note that if the values of each objective function are scaled onto [0,1] using a min-max scaling these two added points become (1,0) and (0,1). Figure 2.5 shows a visual example of this metric. A lower value for this closeness metric will represent a higher quality solution as the solution set will be closer to the ideal point of (0,0).

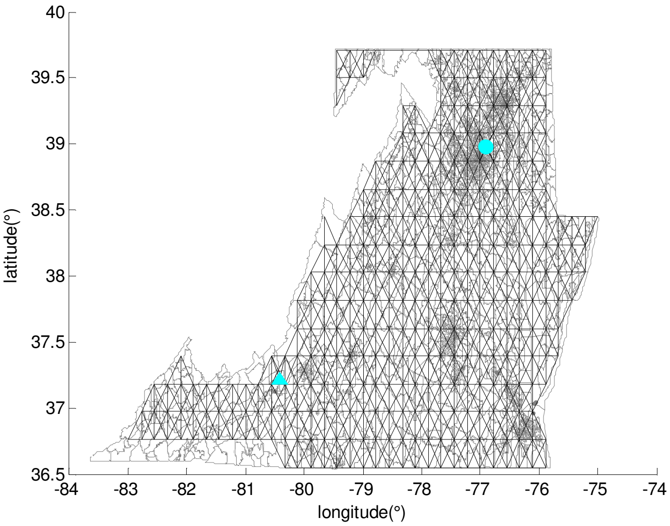


Figure 2.4 EXAMPLE GRID (40x16 NETWOR FOR THE COLLEGE PARK CASE). THE GRAY LINES SHOW THE CENSUS TRACTS IN VIRGINIA, MARYLAND, AND THE DISTRICT OF COLUMBIA. THE BLACK LINES SHOW THE EDGES IN THE GRID. THE CYAN CIRCLE SHOWS THE START POINT (COLLEGE PARK, MD), THE CYAN TRIANGLE IS THE END POINT (BLACKSBURG, VA).

2.5 RESULTS

The results were generated using a computer equipped with an Intel i5 2400 processor and 4 GB RAM. MATLAB's fmincon was used with its default tolerances and the active set method as its optimization algorithm. To generate the crash distribution, all of the relevant error tolerances in MATLAB's ode45 solver were set as 10^{-3}. For each case, three grids were generated. For each grid, the network optimization and the local and greedy improvement approaches were used, each with three different values for the tolerances (which yielded seven sets of solutions per grid and 21 network-based sets of solutions). The non-network approach was also used with four different values for the number of waypoints, which generated four more sets of solutions. Thus, there were 25 sets of solutions for each case. Figures 2.9 and 2.10 show the average computation time required for each approach (the average is taken over the different values for the weights) and the closeness of the sets of solutions that were generated.

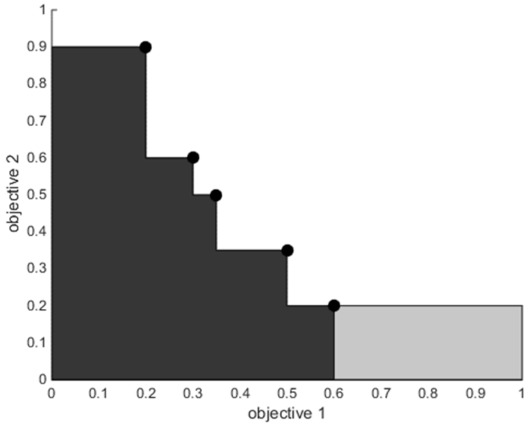


Figure 2.5. EXAMPLE OF CLOSENESS METRIC, THE BLACK POINTS ARE THE PARETO FRONTIER, THE BLACK AREA SHOWS THE AREA CONSIDERED IN THE METRIC FROM [35], AND THE GRAY AREA SHOWS THE ADDITIONAL AREA THAT IS CONSIDERED BY THE METRIC DETAILED HERE.

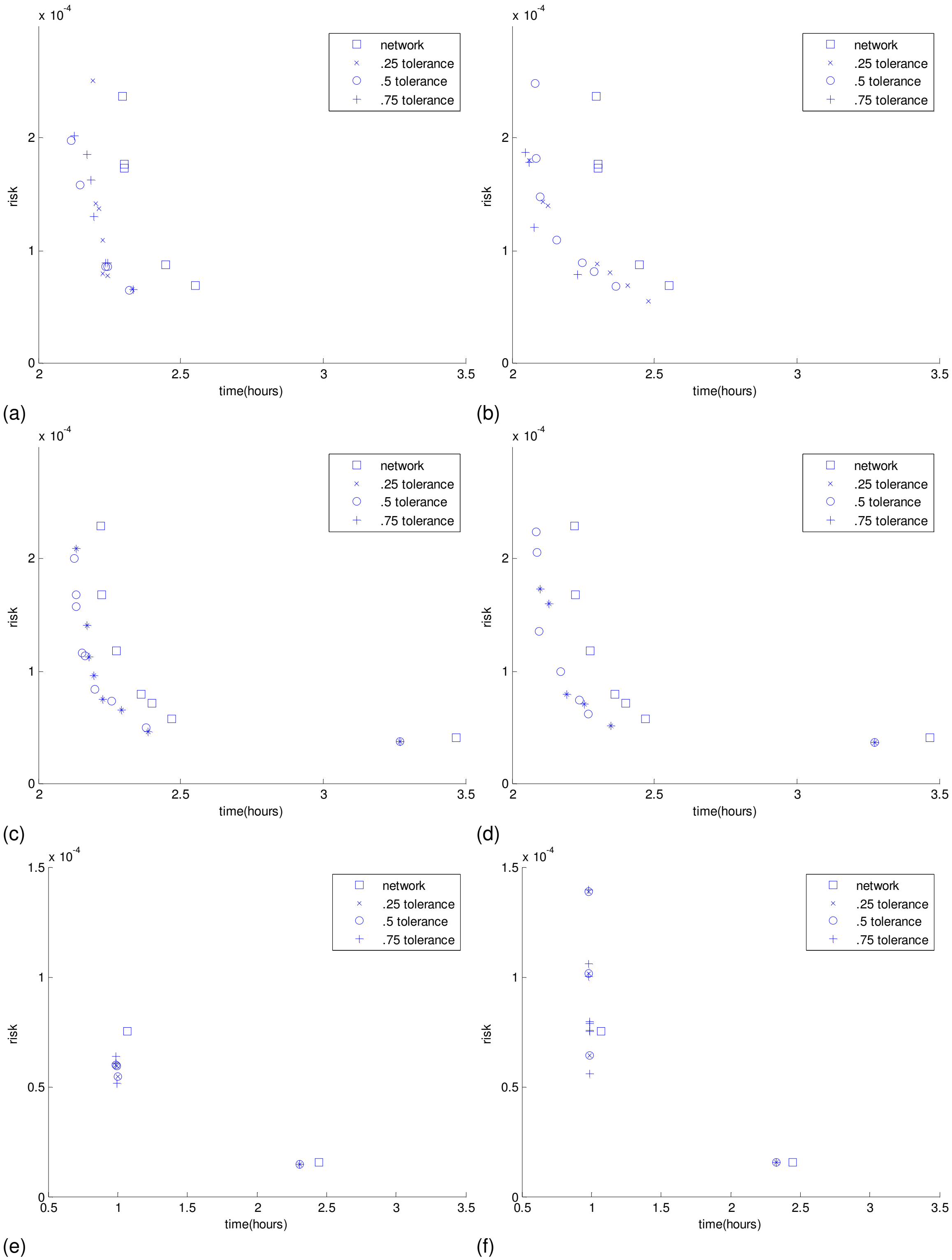


Figure 2.6. SELECTED PARETO FRONTIER RESULTS FOR THE COLLEGE PARK CASE: (a) GREEDY APPROACH, 30x12 GRID, (b) LOCAL APPROACH, 30X12 GRID, (c) GREEDY APPROACH, 40x16 GRID, (d) LOCAL APPROACH, 40x16 GRID, AND FOR PAX RIVER CASE (e) GREEDY APPROACH, 40x16 GRID, (f) LOCAL APPROACH, 40x16 GRID.

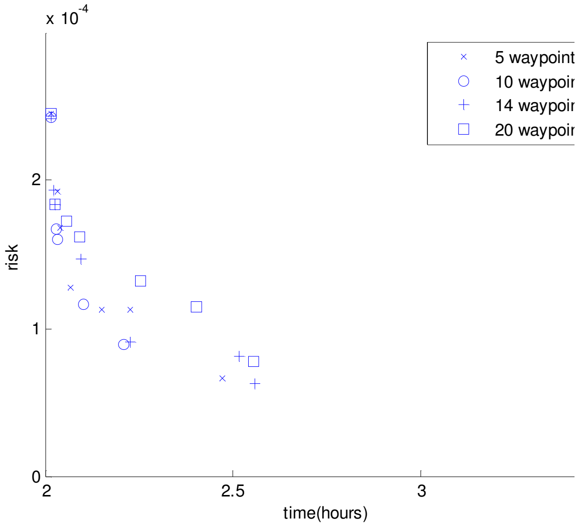


Figure 2.7. PARETO FRONTIER RESULTS FOR THE COLLEGE PARK CASE USING THE NON-NETWORK METHOD FOR DIFFERENT NUMBERS OF WAYPOINTS.

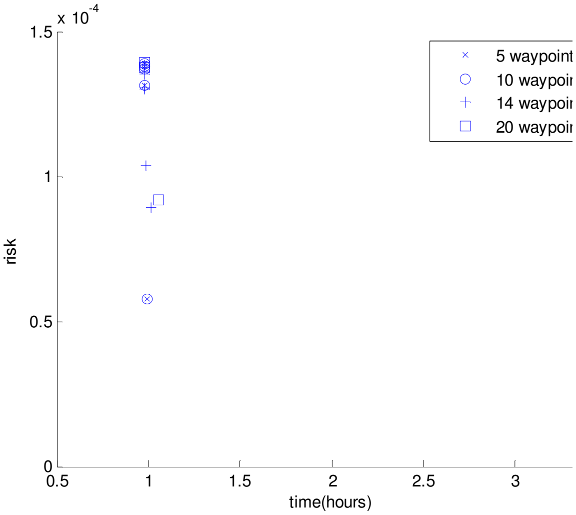


Figure 2.8. PARETO FRONTIER RESULTS FOR THE PAX RIVER CASE USING THE NON-NETWORK METHOD FOR DIFFERENT NUMBERS OF WAYPOINTS.

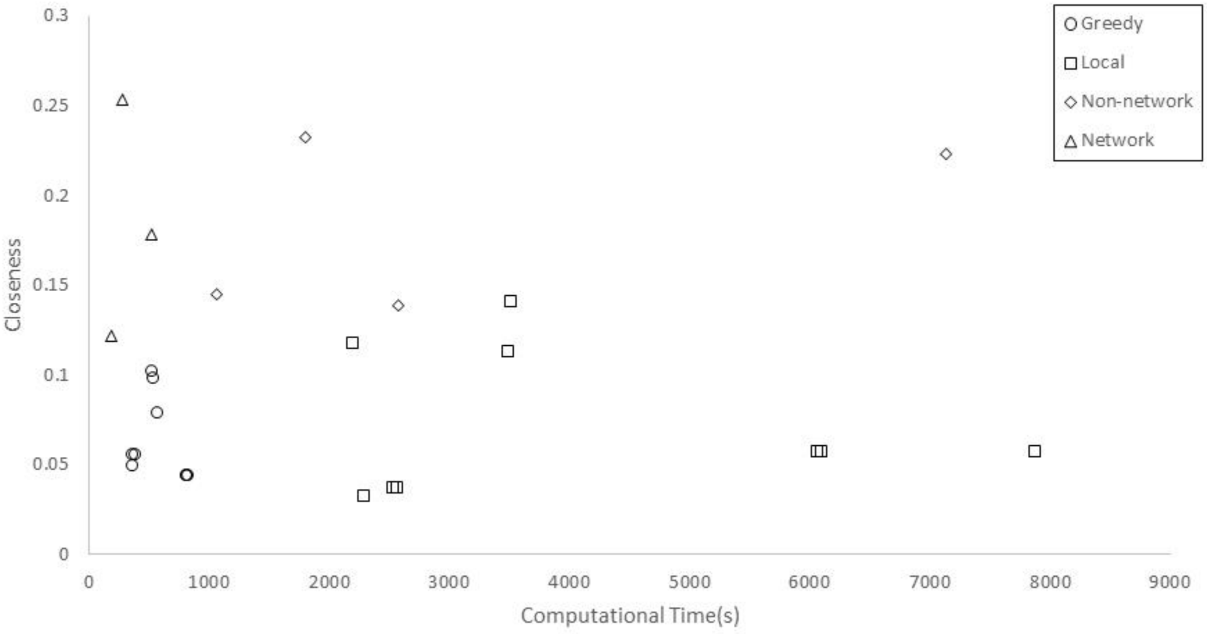


Figure 2.9. CLOSENESS AGAINST COMPUTATION TIME FOR THE COLLEGE PARK CASE.

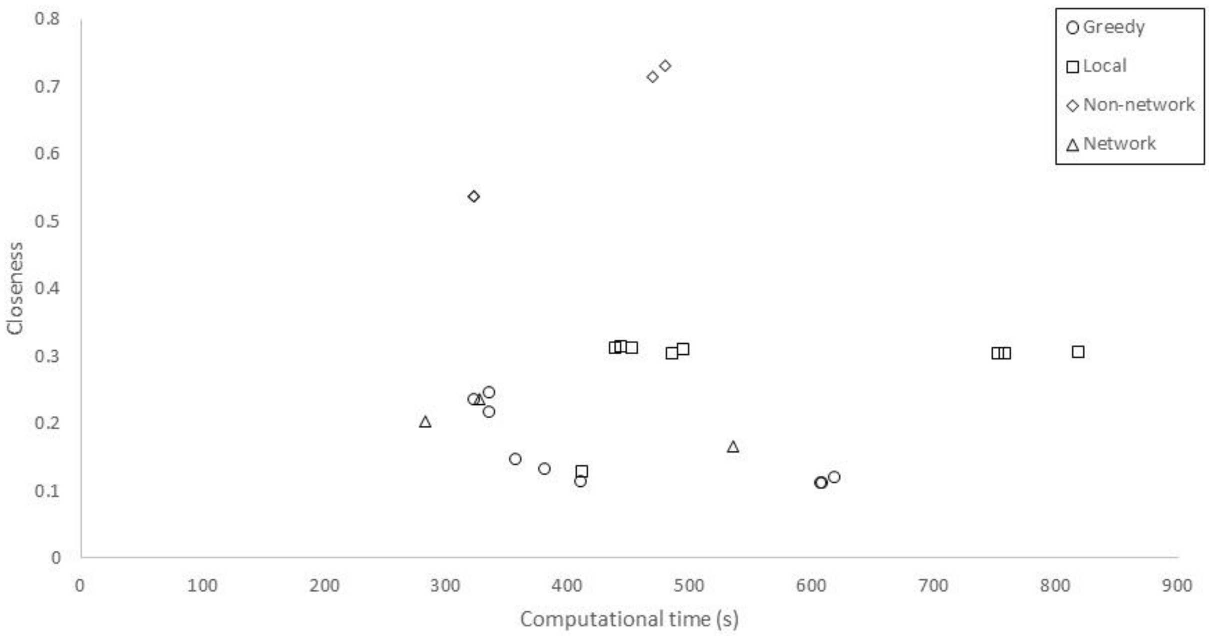


Figure 2.10. CLOSENESS AGAINST COMPUTATION TIME FOR THE PAX RIVER CASE.

The results displayed in Figure 2.6 show that the different approaches generate very different sets of solutions. For the College Park case, the network optimization approach generated a variety of solutions, including some with moderate values of both time and risk, as shown in Figure 2.6. The local improvement and greedy improvement approaches similarly generated a variety of solutions that improved upon those generated by the network approach. The non-network approach also generated a variety of solutions, as shown in Figure 2.7.

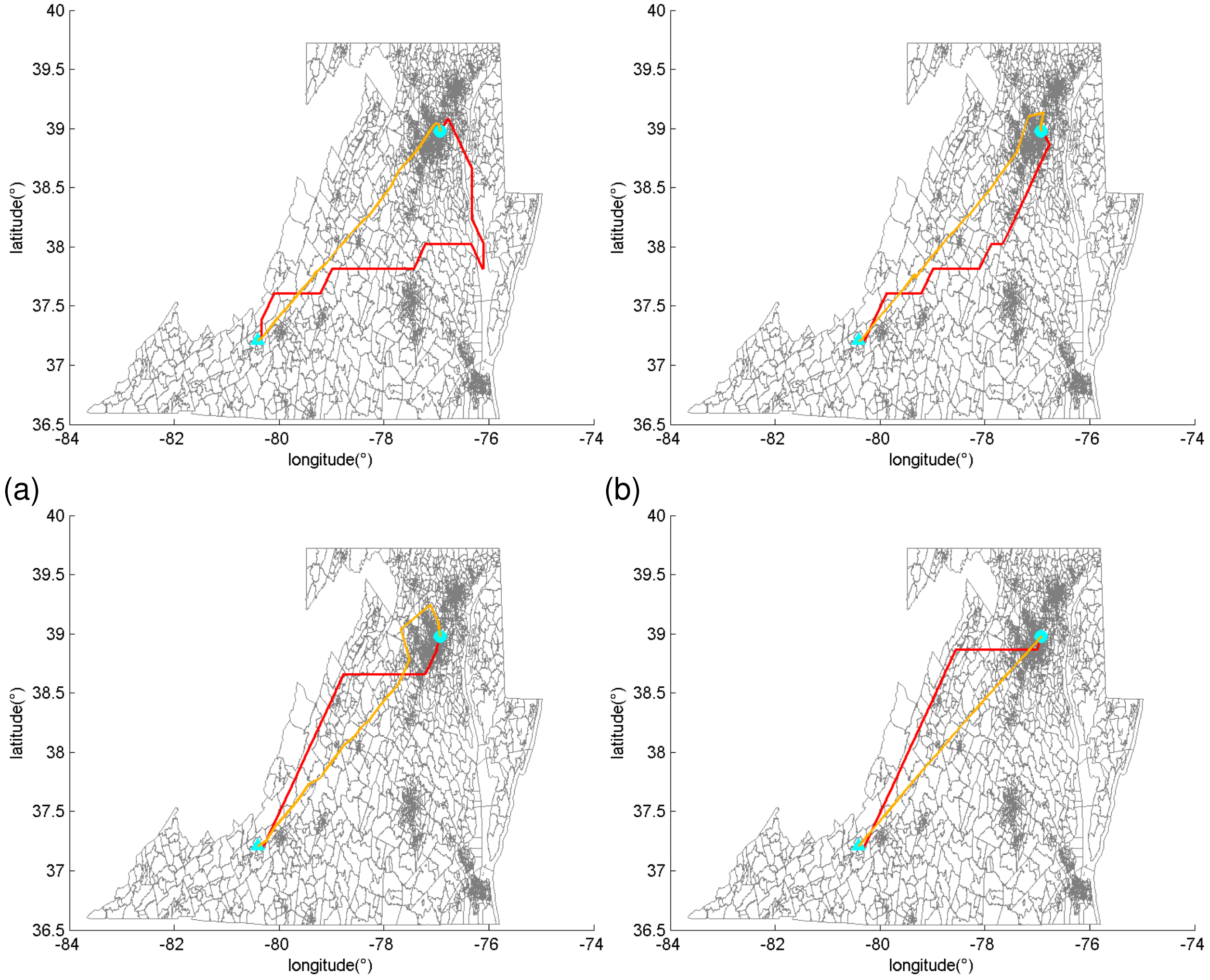
The network optimization approach for the Pax River case generated only two distinct solutions (a nearly straight, minimum-time solution and a wandering minimum-risk solution). As a result, the local improvement and greedy improvement approaches generated sets of solutions that had many solutions near the minimum-time solution and one solution near the minimum-risk solution (as shown in Figure 2.6). The non-network approach was unable to find a low-risk solution; it generated solutions near the initial straight-line solution, as shown in Figure 2.7 and Figure 2.8.

The closeness metric shows that the quality of the solutions generated by the local improvement and greedy improvement approaches were superior to the quality of the solutions that the network optimization step generated. This was true for both approaches in the College Park case. In the Pax River case, the greedy approach with the 40×16 and 50×20 grids generated solutions that reduced closeness. The tolerance value did not show any consistent trend in how it affected the closeness of the solutions. As can be seen in **Error! Reference source not found.** and **Error! Reference source not found.**, the Pareto frontiers generated by these approaches either dominate or are non-dominated by those produced by only using the network approach. The greedy and local approaches both produce superior results to using only the network optimization approach. The Pareto frontiers in Figure 2.7 and Figure 2.8 show that the non-network approach was unable to construct long, low-risk solutions like those that the network approaches found. The lack of low-risk solutions is due to the non-network approach converging to local optima that are near the initial straight-line solution, which prevents the approach from finding solutions near the better solutions that the network-based approaches find. Several examples of the differences between these two types of solutions can be seen in Figure 2.11. The greedy and local approaches appear to be the best of the approaches that were considered in this chapter (that is, they produced the best Pareto frontiers of solutions).

As can be seen in Figure 2.9 and Figure 2.10, neither the local improvement approach nor the greedy improvement approach was substantially better than the other in terms of solution quality; the computational effort, however, was quite different: the local improvement approach required more effort than the greedy improvement approach (the computational effort for both includes the computational effort for the network optimization step). The computational effort of the non-network approach increased as the number of waypoints increased, which is expected given that an increase in waypoints means that the optimizer has more variables that it needs to manipulate. Additionally, as the grid becomes finer (includes more nodes), the computation time required for the greedy improvement approach does not grow at the same rate as the computation time required for the local improvement approach does, which suggests that the difference in the computation time for the two methods would likely increase for larger problems.

The quality of the solutions and the computational effort of the network optimization step varied as the grid size varied, but no trend was evident. In general the solution quality should improve as the grid resolution becomes finer (and the network has more points), though it should be noted that exceptions to this can exist if the nodes at a certain resolution allow for a solution that does not exist for nearby grid resolutions. This issue can be avoided by using a significantly finer grid, though it should be noted that doing so will increase the computational effort of obtaining solutions accordingly. Although the computation time needed to construct the network is low compared to the computation time needed for the non-network approach, the network optimization step does require sufficient memory to store the network and the time and risk of every edge. The number of edges is proportional to the number of nodes, which will increase as the resolution of the grid increases. Methods that use mathematical optimization techniques (such as the non-network approach used in these experiments) do not store the graph and do not require the associated memory.

In terms of both computation time and solution quality, the greedy approach produces the best results of the methods considered. While the local approach does also provide a similar level of improvement in quality over the network solution, the substantially lower time required for the greedy approach would make it more useful in practice.



(a) (b) (c) (d)

Figure 2.11. EXAMPLES OF THE SOLUTIONS GENERATED BY THE NETWORK APPROACH WITH THE 40X16 GRID (RED) AND THE NON-NETWORK APPROACH WITH 20 WAYPOINTS (ORANGE) FOR THE COLLEGE PARK CASE: (a) $w_t = 0, w_r = 1$ (b) $w_t = 0.3, w_r = 0.7$ (c) $w_t = 0.6, w_r = 0.4$ (d) $w_t = 1, w_r = 0$.

2.6 CONCLUDING REMARKS

This chapter presented a bi-objective path planning optimization framework for exploring the tradeoffs between risk and flight time for UAVs. A risk assessment technique and bi-objective optimization methods were developed to find low-risk and time (flight path) solutions. Computational experiments were performed to evaluate the relative performance of the proposed optimization methods. The optimization methods considered were based on a network optimization approach, followed by improvements by a local approach and a greedy approach that used the network optimization results. A fourth approach did not use the network results but locally optimized the coordinates of a fixed number of waypoints.

The results from the computational experiments described the relative performance of the four methods and illustrated the tradeoffs involved. These results indicate that in terms of both computation time and solution quality, the greedy improvement approach produces the best results of the methods considered.

The proposed framework can be extended to incorporate factors such as the shelter provided by buildings that would affect the risk calculations. It can also be extended to incorporate other types of risks (including the risk of mid-air collisions). Future work will consider testing other approaches for generating the initial solutions for the non-network approach, using approximations to evaluate solutions faster, using higher resolution population data for takeoff and landing patterns, using time-dependent population data (time of day, seasonality, special events), developing consistent heuristics for risk for use in an A* search and incorporating shelter data. The problem formulation can be expanded to include selecting the altitude and velocity of each leg (which affects crash location distribution) and avoiding no-fly zones.

CHAPTER 3: MULTI-OBJECTIVE DESIGN AND PATH PLANNING OPTIMIZATION OF UNMANNED AERIAL VEHICLES (UAVS)

29
(The material in this chapter originally appeared as [29])

3.1 INTRODUCTION

There are several potential benefits to combining UAV's design and path planning optimization problems. Optimizing a path for a fixed design or optimizing a design for a fixed path can yield solutions that are inferior to those found by optimizing the design and the path together. Additionally, the ability to adjust the design as well as the path during optimization can allow for a wider variety of solutions to multi-objective UAV optimization problems as changes to a design can allow for a much wider possible range of values for objective functions than would be possible with changing only the path.

Previous works have considered the issue of path planning for the purposes of minimizing risk to third parties or other safety related objectives. Medeiros and Da Silva [20] constructed a visibility graph around highly populated areas and then used discrete path planning algorithms such as Dijkstra's algorithm to generate paths by solving the shortest path problem. Narayan *et al.* [22] considered a trajectory optimization approach for the problem of optimizing multiple objectives using weighted sum methods and dynamic programming. Rudnick-Cohen et al. [28] compared multiple methods for risk-based path planning for a given UAV. They used Monte Carlo simulations of a flight dynamics model to determine a crash location distribution, which was used to estimate the expected fatalities of a flight over inhabited areas. Their results showed that a network approach, combined with a greedy algorithm to improve the best path on the network, generated high-quality solutions more quickly than other approaches tested. For a more in depth survey of methods for UAV path planning in general, consult [11].

Although there has been little work looking into the specific problem of combined UAV design and path planning optimization, there have been several works that have considered issues related to this problem. Kallrath [12] discussed mixed integer programming solutions to production planning and network design problems in the process industry, which have some similarities to design and path planning problems. There are some similarities also to the work done on design and control optimization [25], in that both the design of a system and how the system moves were both optimized. Nigam and Kroo [23] studied the optimization of both the design and mission of a UAV (or multiple UAVs) for surveillance tasks using an approach based around coordinating two optimizers to solve two subproblems that decompose the primary problem being considered with a third coordinating “system-level” optimizer. The approach described by Rastegar *et al.* [5] avoids having the two optimizers directly interact; instead, the third optimizer mediates any conflicts between them. In this chapter we also consider a similar decomposition scheme using two separate subproblems. That is, instead of solving the problems independently, we will present a new algorithm which takes advantage of the coupled nature of the subproblems in order to solve the path planning subproblem while keeping it independent from the aspects of the design optimization subproblem that would normally compromise the optimality of typical methods for path planning. We will also discuss what these aspects of the design optimization subproblem are and the situations where they can cause a conventional path planning method to arrive at suboptimal solution for the combined path planning and design optimization problem.

The organization of this chapter is as follows: Section II defines the problem, including the objectives considered and the specific scenario under consideration. Section III discusses the two models that are considered and compared. Section IV details the approach used to implement these models. This includes a description of the algorithm proposed for solving the path planning subproblem while the UAV design is considered and also a description of the issues the can arise in attempting to use a conventional path planning method. Section V presents the scenario we used for testing the two models, results from applying each of the models to that scenario and a discussion of the results. Section VI presents some concluding remarks and directions for future research.

3.2 PROBLEM DEFINITION

The problem instance considered in this work is to find the optimal design and path for a UAV travelling from College Park Airport in College Park, Maryland, to Virginia Tech Executive Airport in Blacksburg, Virginia. The UAV design consists of the speed that the UAV will be flying along the path and the wing reference area of the UAV. Two objectives were optimized using a weighted sum method to convert the optimization problem into a single objective optimization form in order to generate a Pareto frontier. The two objectives considered were the time needed to traverse the path from starting point to the destination and the risk to the human population on the ground associated with that path. The time objective was based off the assumption that speed of the UAV would remain constant throughout the flight, thus defining it as simply being the length of the path divided by the speed of the vehicle. The risk objective was defined using the risk metric defined in [28]. The crash location probability distribution was parametrized in terms of the design variables under considerations, which was used to optimize the risk objective. The crash location distributions used in the model were generated by running Monte Carlo simulations of an unpowered UAV crashing such as the ones done in [28]. Multiple distributions were generated for different design variable combinations, the parametrization used these distributions to estimate the distribution for arbitrary design variable combinations within the bounds of the design variables using Delaunay triangulation. For the exact formulae for the objectives used see Section 2.3 of Chapter 2.

3.3 PROPOSED MODEL

We solved and compared results from two different models, which are described in Sections 3.3.1 and 3.3.2. In both models we considered the bi-objective problem of minimizing the risk associated with a path and a design and the time needed to traverse the path.

3.3.1 INTEGRATED MODEL

The first model considers design and path planning optimization problem in an all-at-once (AAO) model. This model defines the flight path as a path that travels through a finite number of waypoints and then treat the coordinates of those waypoints as design variables. This model also considers the design of the UAV. This approach allows for the use of standard optimization techniques, making this a fairly conventional approach to attempt to solve the problem as a single (integrated) optimization problem.

3.3.2 DECOMPOSED MODEL

The second model entails separating the problem into two subproblems, a design subproblem and a path planning subproblem, both of which have the same objective functions. There are two sets of design variables: X_D, for the UAV design variables, and X_P, for the UAV path design variables (i.e., coordinate of waypoints) located between the start and end points. The objective function is $f(X_D, X_P)$. In this model, the two subproblems interact by solving the design optimization subproblem for partial and complete solutions to the path planning subproblem and then feeding those solutions back into the algorithm we have developed for solving problems of this type. The structure of this decomposed model is depicted in Figure 3.1.

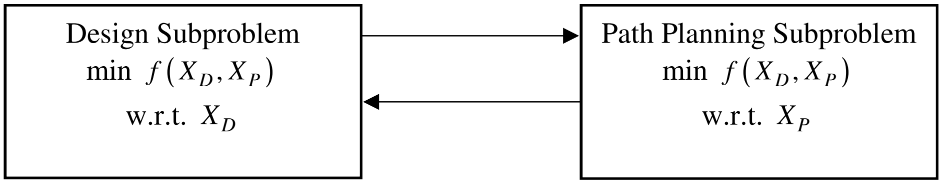


Figure 3.1: The two subproblems considered for the decomposed model

3.4 APPROACH

With the proposed decomposed approach, it is not possible to use a typical graph-based method such as Dijkstra’s algorithm [9] for solving the path planning sub-problem. This is because it is possible that the optimal design and path for reaching a location on the optimal path from the starting point to the goal will not be part of the optimal path to reach the goal. This situation can occur when the optimal design to reach the goal may need to compensate for some part of the optimal path to reach the goal that is not part of the optimal path to a location on the optimal path to the goal. Consequently the optimal path to reach that location on the optimal path may not be a Section of the optimal path to reach the goal.

To get around this issue we developed a new algorithm for handling the path planning subproblem that was capable of accounting for the issues introduced by the capability of the design to change. We can provide a brief high level description of this algorithm as follows. The algorithm works by first using Dijkstra's algorithm, with each edge cost being determined for the optimal path to that edge as determined by solving the design optimization problem. Once a path is found that connects the start node to the goal node, alternate paths from the start to the goal are generated by merging the paths that Dijkstra's algorithm has found with a path from the start to the goal and solving a design optimization subproblem for the resulting path from the start to the goal. As this is done for nodes, they are removed from the part of the graph being considered by the Dijkstra's algorithm portion of this approach. This causes the algorithm to search alternate paths to nodes, which accounts for the issue of past edges becoming sub-optimal choices due to edges located further ahead in the path.

To describe the specific algorithm used for the decomposed approach the following definitions and step by step process are provided:

Node: Represents a location that a path can travel through, keeps track of the best partial or complete solution that involves a path traveling through it.

Solutions Lists: Lists of nodes that depend on other nodes in the solution list for their best solutions, the algorithm will not consider edges that connect to nodes in the same solution list when generating candidate solutions

Edge: Represents that it is possible to travel from the start node of the edge to the destination node of the edge.

Partial Solution: A solution that consists of an incomplete solution to the path planning problem (in the form of an incomplete path that starts from either the start or end node and does not reach the other node) and a solution to the design problem.

Complete Solution: A solution that consists of a complete solution to the path planning problem (a path that travels from the start node to the end node) and a solution to the design optimization problem.

Steps:

1. Initialize two partial solution lists, 1 for the start node and 1 for the goal node, each list should only consist of a single node at this point.
2. For each edge of each node in each solution list that does not go to another node in that solution list, compute the partial or complete optimal solution by solving the design problem for that edge. For complete solutions, determine the path by taking the start nodes path up until that node occurs in its path and then go to the destination node's path starting from where the destination node is in its path. Add solutions that are better than a node's current best known solution to this list of candidate solutions.
3. Select the best solution from the list of candidate solutions to add to the solution lists, if no solutions remain in the list, terminate.
4. With the selected best solution so far two different actions can be taken depending on whether the solution is a complete solution or not. Add partial solutions to the

partial solution list of the start node of the edge associated with the selected solution. For complete solutions, if the destination node of the edge associated with the selected solution is the last node in its solution list, add it to the edge's start node's solution list, otherwise add the complete solution to a new solution list.

5. Set all edges leading to and from the edge associated with the selected solution destination node to be recomputed and remove any nodes in the partial solution list that may have depended on the node associated with the solution added if that node was originally in a partial solution list. Go back to step 2.

3.5 RESULTS

A version of the AAO model was implemented for a small number of waypoints. In the model, risk is determined by determining a distribution of possible UAV crash locations and evaluating the risk level, as defined in [4]. For the population density data needed for the model, data from the US census was used. The risk was evaluated along with path segments, which are straight lines between waypoints. To evaluate the risk along the path being followed, the risk was calculated at multiple points along each segment of the flight path in order to approximate the value of the risk along the entire segment. Flight time was calculated by dividing the length of the path by the airspeed.

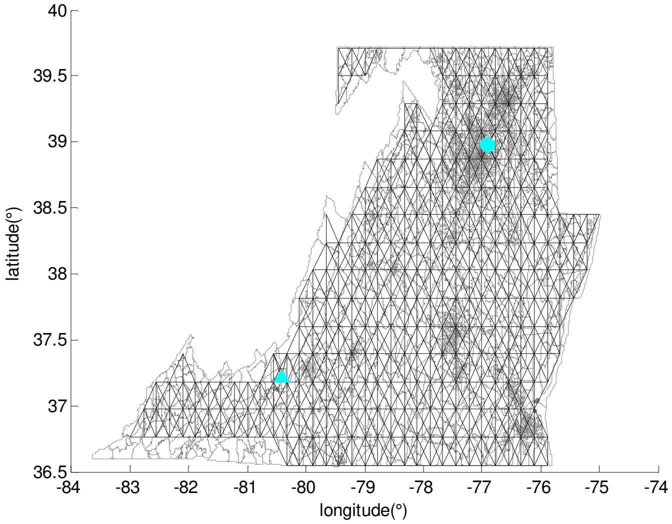


Figure 3.2: The grid used by the decomposed approach for the region under consideration. The cyan circle indicates the start point at the College Park, Maryland airport and the cyan triangle indicates the end point at Virginia Tech Executive Airport in Blacksburg, Virginia [3]

For the integrated model, three waypoints were considered along with the airspeed (V) and the wing reference area (S) of the vehicle. The start point was the College Park, Maryland airport and the destination was Virginia Tech Executive Airport in Blacksburg, Virginia. The locations of the two points relative to the region under consideration can be seen in Figure 3.2. Figure 3.2 also depicts the grid that was used for the decomposed approach. The upper and lower bounds were set to keep the waypoints within a box enclosing the region being considered and to keep design variables within the range of values used for Monte Carlo simulation. The initial conditions used for optimization were a straight line path for the waypoint variables, an airspeed of 50 m/s, and a wing reference area of 16.17 m^2. Optimization was done using MATLAB's fmincon [1] function for the integrated model.

For the decomposed approach a 40×16 node grid was created and then pruned to remove any nodes outside the states that were being considered in the grid. The initial solution for the design problem was the same as the one that was used for the integrated approach.

For both models, the integrated and decomposed, 11 equally spaced weights between a 100% and 0% weighting on the risk objective and the time objective were used to generate a range of solutions for different objectives. The results from this are provided in Figure 3.3.

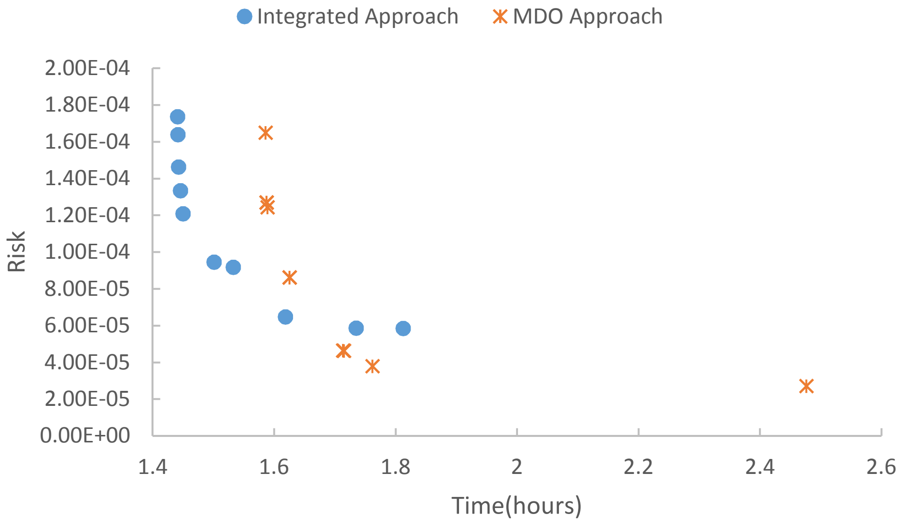


Figure 3.3: Pareto front of risk vs. time for both approaches

Design variables		Waypoint coordinates					
V(m/s)	S(m)^2	x1(deg)	x2(deg)	x3(deg)	y1(deg)	y2(deg)	y3(deg)
70.00	16.13	-76.79	-77.60	-79.64	39.12	39.31	37.18
70.00	16.48	-77.15	-78.36	-79.53	39.20	39.25	38.71
62.62	16.28	-77.14	-77.91	-79.13	39.28	38.82	37.80
66.43	16.34	-76.88	-77.63	-79.09	38.77	38.48	38.18
70.00	16.10	-76.91	-76.90	-79.41	38.76	38.77	37.75
70.00	16.18	-77.10	-77.52	-79.83	38.80	38.61	37.57
70.00	16.66	-76.95	-77.09	-78.43	38.94	38.85	38.29
70.00	16.15	-77.03	-78.07	-78.69	38.96	38.42	38.13
70.00	16.14	-77.22	-78.25	-79.62	38.85	38.30	37.64
70.00	16.17	-78.34	-79.06	-79.49	38.28	37.91	37.69

Design variables		Objectives	
V(m/s)	S(m)^2	t(hours)	risk
70.00	16.13	1.81	5.86×10^{-5}
70.00	16.48	1.74	5.88×10^{-5}
62.62	16.28	1.62	6.48×10^{-5}
66.43	16.34	1.53	9.18×10^{-5}
70.00	16.10	1.50	9.46×10^{-5}
70.00	16.18	1.45	1.21×10^{-4}
70.00	16.66	1.45	1.34×10^{-4}
70.00	16.15	1.44	1.46×10^{-4}
70.00	16.14	1.44	1.64×10^{-4}
70.00	16.17	1.44	1.74×10^{-4}

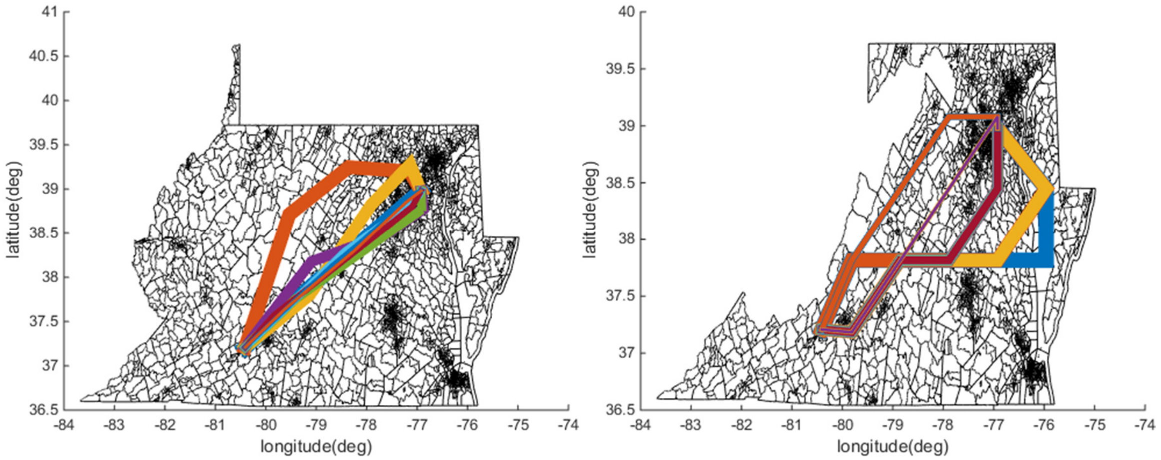
Table 3.1: Solutions from Figure 3.3 for integrated approach

Table 3.1 depicts the design variable values for the solutions shown in Figure 3.3 for the integrated approach. The variables x1, x2, x3 and y1, y2, y3 describe the 3 waypoints in terms of their longitudes (x) and latitudes (y). As expected most solutions are close to the upper bound for speed (70 m/s) as currently no constraints are being enforced that can limit the speed based off any of the other design variables and the objective function used for risk can be reduced by reducing the time needed to traverse a path, which is minimized by maximizing the speed of the UAV. Several solutions did deviate from the maximum speed allowed, indicating that the optimal solution is not necessarily always to travel at the maximum possible speed. Variations in the optimal wing reference area indicate that while there may not exist a specific trend, it was possible to reduce the risk for a given path by adjusting the wing reference area. The lack of a trend here is expected, as adjusting the wing reference area will change the shape of the crash location distribution, but the optimal shape of the distribution will depend on the population distribution along the path taken, meaning that the optimal wing reference area for a specific path could be significantly different depending on what sort of distribution is best for that path. These results demonstrate the capability for this type of optimization for generating a variety of solutions for multiobjective problems as the variation in the wing reference area in conjunction with the variation of the waypoint locations create a clear trade-off between time and risk.

Design variables		Objectives	
V(m/s)	S(m^2)	t(hours)	risk
70.00	16.17	1.59	1.65×10^{-4}
70.00	16.17	1.59	1.27×10^{-4}
70.00	16.17	1.59	1.27×10^{-4}
70.00	16.17	1.59	1.27×10^{-4}
70.00	16.17	1.59	1.24×10^{-4}
70.00	16.17	1.63	8.63×10^{-5}
70.00	16.17	1.63	8.63×10^{-5}
70.00	16.17	1.71	4.65×10^{-5}
70.00	16.18	1.71	4.64×10^{-5}
70.00	16.18	1.76	3.80×10^{-5}
70.00	16.17	2.48	2.73×10^{-5}

Table 3.2: Solutions from Figure 3.3 for decomposed approach

Table 3.2 depicts the design variable values for the solutions shown in Figure 3.3 for the decomposed approach. The decomposed approach is capable of generating superior solutions in terms of the risk objective as unlike the integrated approach the decomposed approach will not get stuck and local optima and will thus actually reach globally optimal solutions for both objectives. However, the solutions generated are restricted to remaining on the grid, which prevents the solutions from taking the most direct possible routes to the goal, consequently the integrated approach is able to generate solutions that take more direct paths to the goal and thus solutions that are more optimal with respect to the time objective. This can be seen in Figure 3.4, as the solutions for the decomposed approach are forced to make small detours instead of following a direct straight line path. The results from Table 3.2 also indicate different trends in terms of the optimal design variables in that the results for the decomposed approach indicate the existence of an ideal point for the design optimization problem. The optimal wing reference areas all tended to be near the initial condition chosen for that variable, which is likely due to the presence of a local optimal located there. A possible cause for this may be that the optimal wing reference area may be very sensitive to the locations of the waypoints that make up the path, since the decomposed approach does not have the ability to make small adjustments to where it places waypoints since its path is defined between elements located on a grid.



(a) **(b)**

Figure 3.4: Plots of solution paths for: (a) integrated approach, (b) decomposed approach, with thickness of a path corresponding to the amount of risk for that path, the thicker the line path the lower the risk

Figure 3.4 also provides a comparison of the paths generated by the different approaches. As can be seen, the more globally optimal solutions that the decomposed approach generates for the cases with a heavier weight on risk follow fundamentally different paths than the solutions generated by the integrated approach for those weights. As discussed before, the cause for this is due to the integrated approach getting stuck at local optima, which keeps it from finding more globally optimal solutions such as the ones found by the decomposed approach. A major difference between the paths generated by the two methods is that the paths generated by the integrated approach cross over into

West Virginia, while the paths generated by the decomposed approach remain inside the 2 states (Maryland, Virginia) plus Washington D.C. under consideration for this scenario. This is due to the fact that the integrated approach has to use a bound constraint to limit the region where waypoints can be placed, which leads to a Section of West Virginia being present in the feasible region for waypoint placement. However, with the decomposed approach, it is possible to always restrict the paths under consideration to stay within the 3 states being considered as by not placing nodes outside those states the paths found will never be able to go outside them.

3.6 CONCLUDING REMARKS

A new algorithm was presented for solving path planning subproblem within a decomposed approach for solving path planning and design optimization subproblems. The decomposed approach was able to find more globally optimal solutions than an integrated approach, though it did suffer some restrictions that resulted from the discrete nature of how it determined the optimal path. Future work on this topic will likely include the development of approaches that can handle constraints that relate the design variables of both subproblems, methods for refining the size of the grid used for the path planning subproblem inside the algorithm to reduce computational cost and consideration of additional design variables relating the physical design of a UAV and its operating parameters.

CHAPTER 4: ROBUST RISK-BASED PATH PLANNING FOR UNMANNED AERIAL VEHICLES

4.1 MOTIVATION

Although a UAV operator controls some of the variables that affect the risk of UAV operations, such as the design of the UAV or operational parameters such as flight speed, there are also variables affecting the risk which cannot be controlled or even known in advance. The uncertainty in these variables creates uncertainty about the risk associated with a path. In practice this uncertainty can lead to the optimal path changing depending on the values taken by uncertain variables. In order to determine a path for a UAV that is optimal with respect to uncertainty, it is necessary to determine a path that is optimal under any possible combination of uncertain variables, thus we need to find an optimal path that is robust with respect to uncertainty.

4.2 LITERATURE REVIEW

Uncertainty can enter path planning problems in a variety of ways, such as uncertainty about the effects of actions being taken, uncertainty about the location or state of a vehicle, uncertainty about a vehicles sensor reading and uncertainty about the cost of a path being taken [14]. Robust path planning approaches attempt to determine either an optimal or feasible path while accounting for sources of uncertainty such as the ones discussed.

Much of the existing work on robust path planning for robotic systems primarily focuses on dealing with either uncertainty affecting constraints or uncertainty about the effects of actions taken. Dadkhah and Mettler [6] provides a survey of how these types of methods have been applied for use in UAV systems. These approaches do not extend to be able to minimize an objective subject to uncertainty, as their primary goal is to determine a feasible path with respect to uncertainty, rather than an optimal one. When dealing with robust risk-based path planning, uncertainty affects the risk associated with a given path, meaning that a path planning technique that takes into account uncertainty in the objective function is necessary.

In terms of the more general area of robust optimization, the problem of handling uncertainty in an objective function can be more easily described. Beyer and Sendhoff [2] provides an overview of the general concept of robust optimization and the forms that robust optimization problems can take. In the case of robust risk-based path planning, using a min-max structure makes the most sense, where the inner problem consists of determining the worst case risk of a path while the outer problem consists of minimizing the cost of the path with respect to the objective in use. An important concept about robust optimization that needs to be considered here is that the worst case being found for a path needs to be the largest possible risk for that path, locally maximal risk values will not result in a truly robust solution. Thus we need a method for solving our robust risk-based path planning problem globally, rather than locally.

It can be observed that determining the worst case cost for a path where the cost of the path is parameterized by some function is a similar problem to the problem of design and path planning discussed in Chapter 3. Instead of solving a design optimization problem which consists of minimizing the cost of an objective function, determining the worst case cost would consist maximizing the cost of an objective function instead. These two problems are equivalent, as negating the sign of the objective function in a maximization problem converts that problem into a minimization problem. In this chapter we discuss how an approach based off this strategy can be developed for solving a robust risk-based path planning problem.

4.3 PROBLEM FORMULATION

In order to perform robust risk-based path planning our goal is to determine the optimal path under worst case for possible wind conditions that can occur along the path. Let the wind conditions along the path be represented by associating sets of wind conditions with specific spatial region making up the area being traversed. The solution to the problem should consist of an optimal path and the worst case wind conditions for all spatial regions that the optimal path passes through.

Formally written, our problem is:

Let $p =$ a path from point A to point B, which consists of multiple legs

$p =$ a sequence of legs making up the *path:* $e_1, e_2, \ldots, e_n$

Let $P =$ the set of all possible paths from point A to point B

Let $e =$ a leg of a path, representing an edge in the graph used for path planning

$e =$ A pair of points denoting the startpoint and endpoint of the leg

Let $T(p) =$ time to travel along path p

Let $T(e)$ = time to travel across leg e

Let $R_{max}(p)$ = worst-case expected number of fatalities associated with using path p

Let $S_{worst}(p)$ = Wind conditions that produce $R_{max}(p)$for a given path p

Let $R(e,s)$ = expected number of fatalities (risk) for leg e under wind conditions s

Let s = a combination of wind conditions that can affect the risk of a path

s = a mapping of vectors of windspeeds to specific regions where they occur

Let S = the set of all possible wind conditions being considered in this problem

w = weighting coefficient used to combine objectives to get multiobjective solutions

We can formally write the problem that we need to solve as:

Given:

$$T(p) = T(e_1) + T(e_2) + \dots + T(e_n)$$

$$R_{max}(p) = R(e_1, S_{worst}(p)) + R(e_2, S_{worst}(p)) + \dots R(e_n, S_{worst}(p))$$

$$S_{worst}(p) = \underset{s \in S}{max} \sum_p R(e_i, s)$$

Solve:

For a given weight $w \in [0,1]$

Find the path p_{robust}, such that

$$p_{robust} = \underset{p \in P}{argmin} \; w\, T(p) + (1\text{-}w)\, R_{max}(p)$$

Figure 4.1 depicts the representation chosen for the wind conditions across the area being traversed. Each region is represented a box with an associated windspeed vector representing the wind in that region. For a given path, the risk of that path can only be affected by the wind conditions for regions that the path actually passes over, thus when determining Sworst(p) for a path p, we only need to maximize risk with respect to the wind conditions in regions that the path actually passes through. This is important, as the area being traversed can easily consist of hundreds of regions, attempting to maximize risk with respect to the wind conditions in all of these regions simultaneously would require a significant amount of computational time.

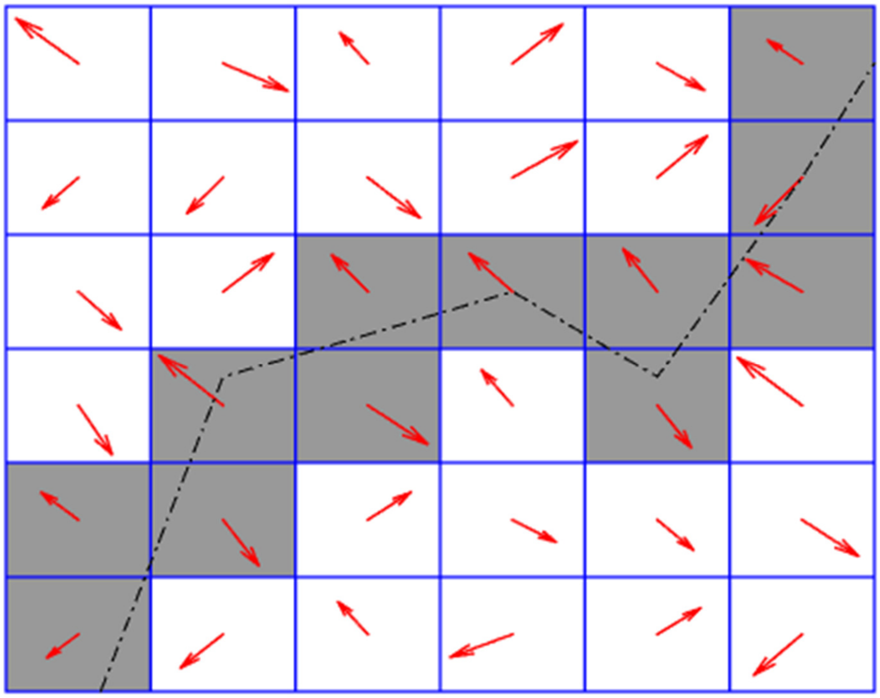


Figure 4.1: Visual representation of how wind is represented for the model used for robust path planning. Each square represents a region where the wind speed in that region is defined as a 2 dimensional vector. The path taken through the regions is indicated by the dashed line and the highlighted wind regions indicate the regions which the path passes through.

To model the effects of how the wind conditions along a path can affect the risk associated with that path, we model how different wind conditions can change the crash distribution of the UAV, by using this modified crash distribution we can determine what the risk should be for a path under a specific set of wind conditions. The current model in use for this is based off linearly shifting the crash distribution of the UAV in the presence of no wind, this model was calibrated by generating crash distributions for multiple possible wind conditions through the use of monte-carlo simulation approach detailed in Chapter 2. The slopes used in the linear shift model were a slope of 0.0728 m per m/s windspeed in the X direction and 0.1207 m per m/s windspeed in the Y direction, the maximum windspeed considered in both the X and Y directions was +-15.1994 m/s (corresponding to 34 mph). The linear shift model is used due it being relatively accurate as a method of approximating the changes in the crash distribution and also being extremely fast to evaluate. The linear shift model's main limitation is that it does not represent changes that can occur in the shape of the crash distribution due to wind, however such effects have a much smaller impact on the risk associated with a path relative to the effects of the crash distribution shifting. However, the approach detailed in this document will work for any model for the crash distribution as a function of wind conditions, thus more sophisticated models can still be used with the robust planning approach discussed.

4.4 SOLUTION APPROACH

As shown in Chapter 2, risk-based path planning approaches for minimizing the risk to third parties on the ground can suffer from issues due to local optima if a graph based approach is not used for representing the path. However, the wind conditions in each region are continuous variables, which cannot be efficiently represented using a graph. In order to account for this, we employ the design and path planning approach developed in Chapter 3, where the design problem considered is the problem of determining the wind conditions that maximize the risk of a path. This allows us to obtain the global optimality benefits in terms of the path planning aspects of this problem. However, for robust optimization global optimality is needed for both the variables being optimized and also the uncertain variables being used to maximize risk. We thus need to extend the approach from Chapter 3 in order to be able also perform a global search on the uncertain variables in this problem, which are the wind conditions that occur along the path.

Recall that we defined the risk of a path as $R(p, s) = R(e_1, s) + R(e_2, s) + \ldots R(e_n, s)$.

We perform robust path planning in two steps: (1) a local search step that solves the problem formulated in Section 4.3 to find a locally optimal solution and (2) a global search step that checks the global optimality of the Step 1 solution by attempting to find a new worst case for it. A graphical representation of this algorithm is provided in Figure 4.2.

Step 1 utilizes the combined design and path planning approach presented in Chapter 3 to solve a discrete path planning optimization problem on a graph to determine the optimal path p^* under worst case wind conditions and a worst case local maximization problem to determine the worst case wind conditions s^*.

Step 2 performs a global search on possible wind conditions in order to determine if the wind conditions $s*$ found in Step 1 lead to the global maximum risk for the path $p*$ found in Step 1. This is done using a random search on the uncertain variables (the wind conditions) in order to try to find a new worst case scenario. If a new worst case scenario s' is found such that $R(p*,s') > R(p*,s*)$, a local maximization is done (using the new worst case scenario s' as the initial solution) to find the worst case scenario s'' that maximizes $R(p*, s)$.

After finding the worst case scenario s'', the algorithm calculates $R(p*, s'')$ and then returns to Step 1 to search for a better path p that minimizes $wT(p) + (1\text{-}w)R(p, s'')$. If no path better than $p*$ can be found, then the algorithm returns to Step 2 in order to continue the global search. Otherwise, $p*$ is updated (it is a new, potentially better path), and the algorithm returns to Step 2 to perform a global search on the wind conditions for $p*$. This causes the algorithm to eventually converge to the globally optimal solution to the problem formulated in section 4.3. When the algorithm has converged to the globally optimal solution, Step 2 should not be able to find a new worst case for the path found. Thus the algorithm should be terminated when the optimal path $p*$ and its worst case scenario $s*$ remain constant for a large number of iterations.

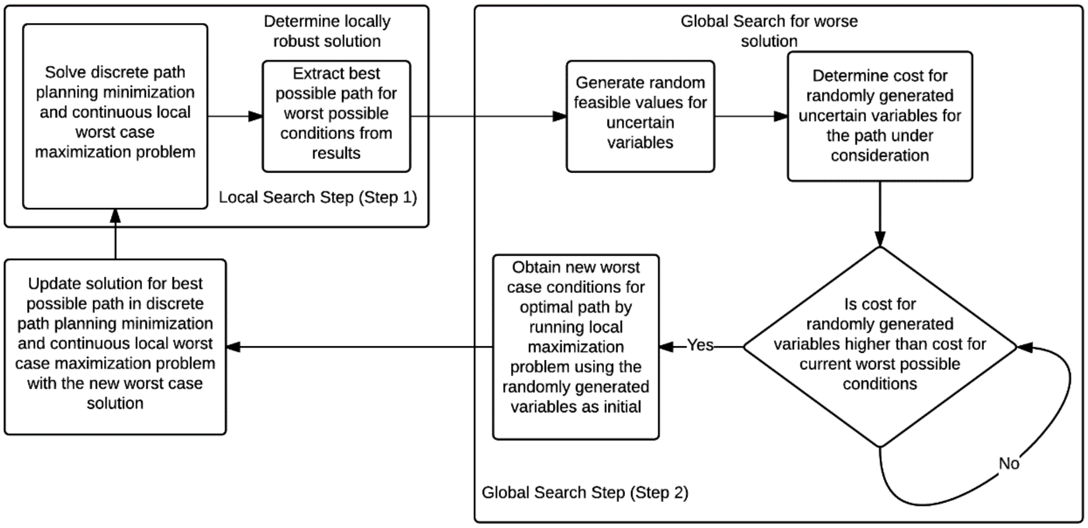

Determine locally robust solution
Solve discrete path planning minimization and continuous local worst case maximization problem
Extract best possible path for worst possible conditions from results
Local Search Step (Step 1)
Global Search for worse solution
Generate random feasible values for uncertain variables
Determine cost for randomly generated uncertain variables for the path under consideration
Is cost for randomly generated variables higher than cost for current worst possible conditions
Yes
No
Obtain new worst case conditions for optimal path by running local maximization problem using the randomly generated variables as initial
Update solution for best possible path in discrete path planning minimization and continuous local worst case maximization problem with the new worst case solution
Global Search Step (Step 2)

Figure 4.2: High level flowchart for the robust path planning approach described

4.5 EXPERIMENTAL SETUP

The robust path planning approach was tested in a single specifically chosen scenario which had a noticeable variation between the risk not accounting for the effects of wind on the crash distribution and the worst case risk accounting for the effects of wind. The case considered was determining a route going through the region consisting of Maryland, Virginia and Washington D.C, with a startpoint at (-76.00154, 36.8792) and an endpoint at (-79.95189, 37.52974). The robust path planning approach was run until a global solution appeared to be observed, meaning that the approach was unable to find a worse solution for the current optimal path for a large number of iterations. In order to obtain multiobjective solutions, the same approach was used as in Chapter 2, with the robust path planning approach being run for 11 equally spaced weights between a purely risk weighted case and a purely time weighted case. Unlike the approach in Chapter 2, US census block data was used instead of census tract data, this was done in order to take advantage of the higher resolution population information available in the census block data. This allows for a more accurate representation of the risk of a path traveling through an area, which is necessary for being able to consider the effects of shifts in the crash distribution on the risk. In order to test the validity of the solutions being generated by the robust path planning approach, a deterministic risk-based path planner similar to the non-network approach discussed in Chapter 2 was also tested with the same objective as the robust path planning approach except that the windspeeds were all held to be zero.

4.6 RESULTS

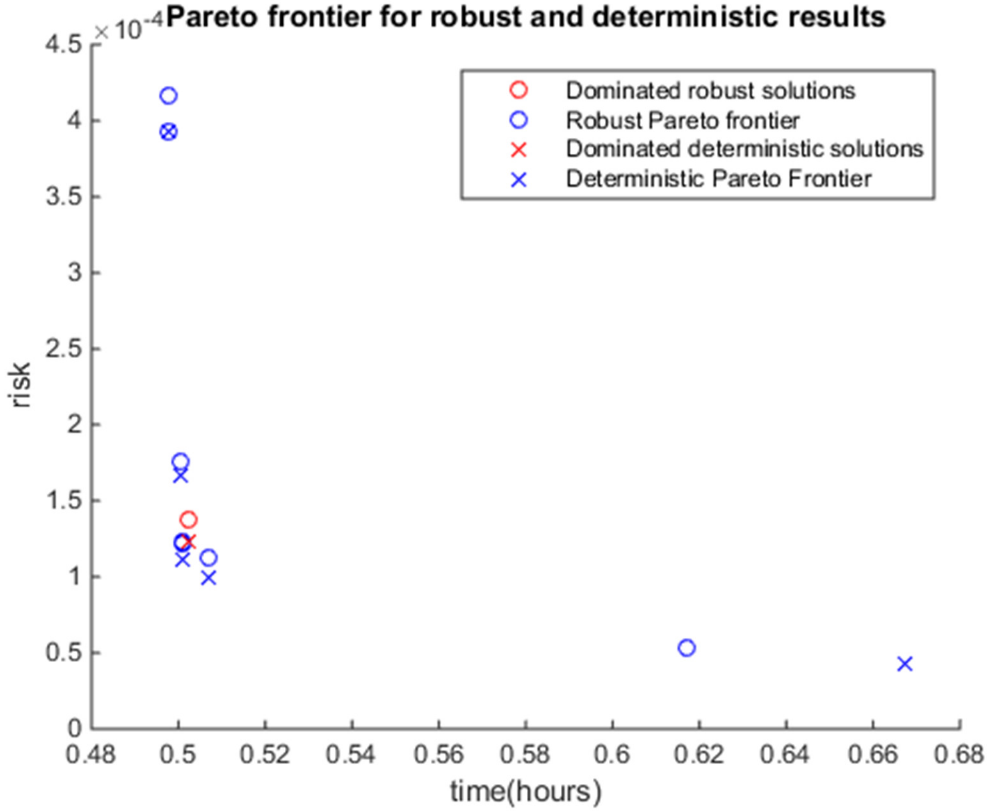


Figure 4.3: Plot of Pareto frontier and dominated solutions for case considered, note that all risk values provided for the Robust Pareto frontier are the worst case risk values as determined by the robust path planning approach

Figure 4.3 depicts the Pareto frontiers for the results from both the robust path planning approach and from the deterministic path planner. It can be seen that outside of the solutions that only minimize risk, the solutions found by the robust path planning approach are similar to those found by the deterministic path planner. This is expected as the uncertainty considered here only affects the risk objective, thus as more weight is put on the time objective, less weight is put on accounting for uncertainty in the risk objective, leading to solutions that are similar to what would be found when not considering uncertainty. In particular, the time optimal case actually leads to what would appear to be the exact same solution for both time and risk as found by the deterministic path planner. However, it should be noted that the reason for this is due to the fact that in the time optimal case, the robust path planning approach cannot actually do anything to account for uncertainty in the risk objective, since the risk objective will have no contribution to the objective function being considered. This is why there is also a robust solution above the deterministic time optimal solution, the robust solution above that solution corresponds to the case where the most of the weight is put on the time objective, but some weight is still put on the risk objective, which allows the robust path planning approach to determine what the worst case risk should actually be for the path in question. For the pure risk case the robust path planning approach obtained a significantly different solution when compared with the deterministic path planner. In this case the robust path planning approach was able to identify a solution with a slightly higher worst case risk than deterministic solution’s risk. In this case the path found by the robust path planning approach is a different path than the path found by the deterministic path planner, this indicates that the worst case risk for the path found the deterministic path planner is actually higher than the robust path planning approach's solution's risk, as the robust path planning approach would have searched through this solution in the process of reaching the solution which it obtained.

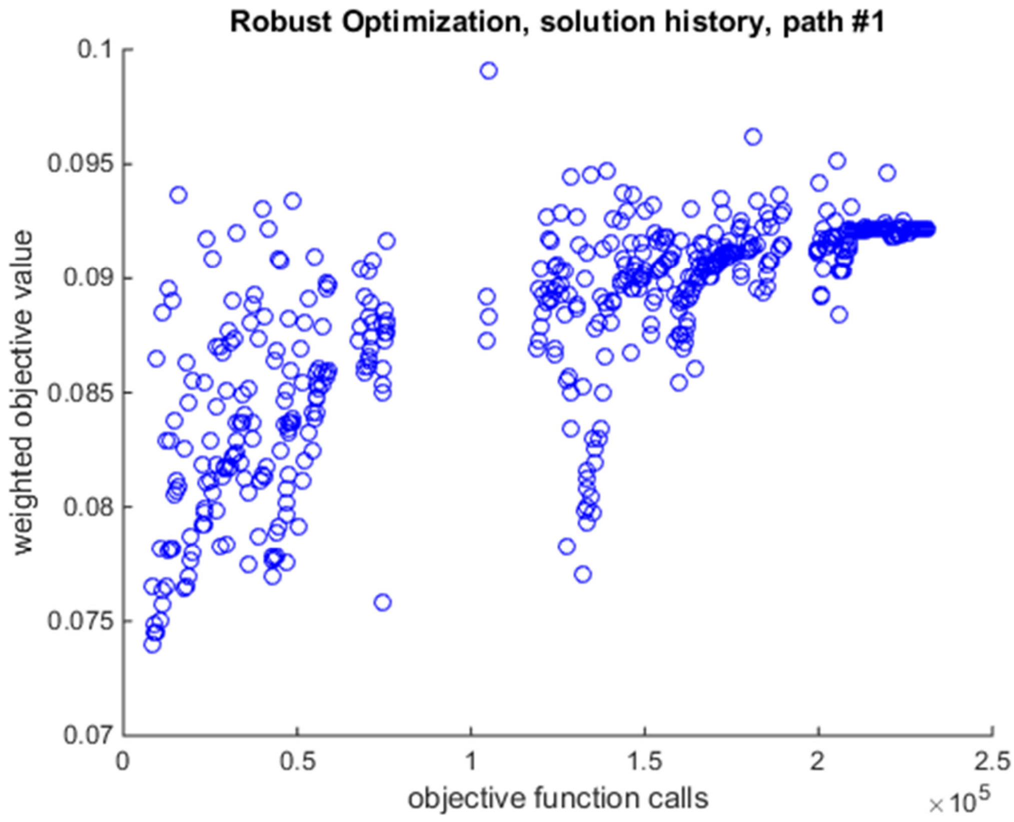


Figure 4.4: Solution history of the global search step of the robust path planning approach, pure risk case, gaps indicate multiple objective function calls being needed for optimization

The process the robust path planning approach went through in order to obtain a solution in the pure risk case is depicted in Figure 4.4. Each time the solution increases in value indicates that the robust path planning approach identified a new worst case for a solution, each time the solution decreases in value indicates that the robust path planning approach determined that an alternate path has a lower objective function value than the current one. A total of 110 different paths are searched in the pure risk case, however many of these paths are searched multiple times due to the robust path planning approach not finding the true worst case for these paths the first time that they are considered. As a result of this, the pure risk case requires an extremely large number of objective function calls before finally converging to its final solution.

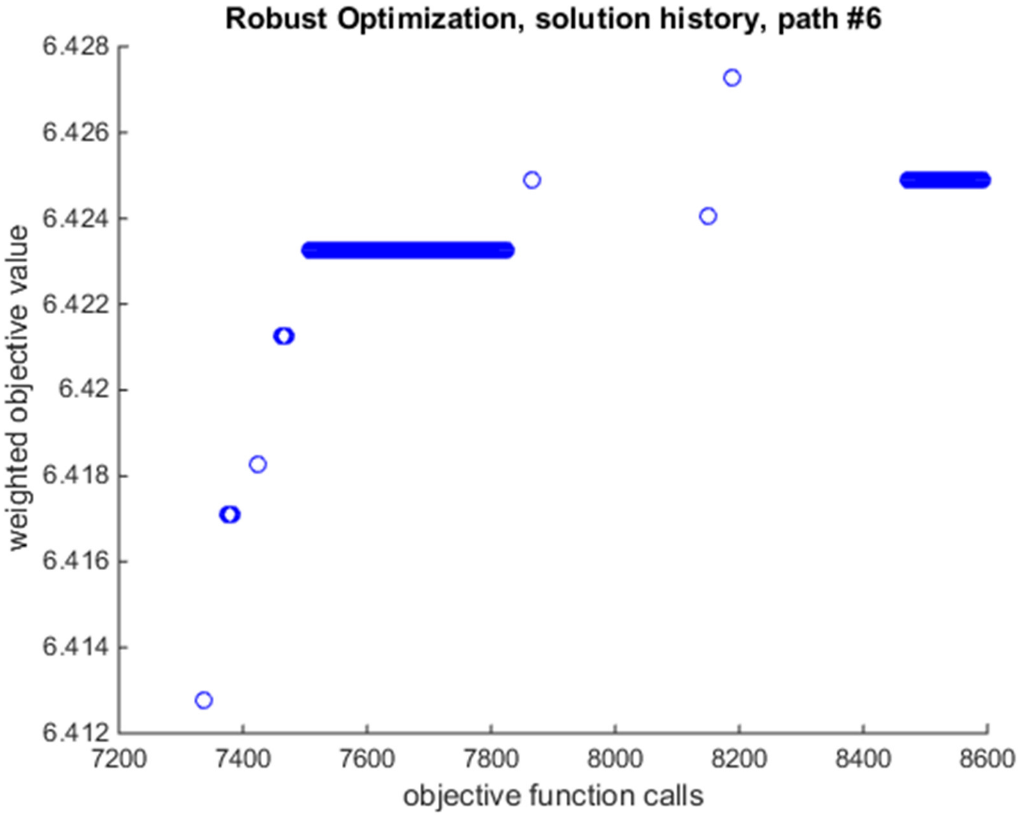


Figure 4.5: Solution history of the global search step of the robust path planning approach, 60% time weight, 40% risk weight, gaps indicate multiple objective function calls being needed for optimization

Figure 4.5 shows the process the robust path planning approach goes through for a different combination of objective weights, to provide a reference for how much longer the pure risk case took to compute. As can be seen, a significantly smaller number of objective function calls are needed in this case. A total of only 2 different solutions are considered in this case, the final solution arrived at is the same as the initial solution that the global search step starts from. The second solution is initially considered because its initial worst case appears to be better the first solutions, however after a single iteration of the global search step a worse solution is identified for it, causing the first solution to become the minimal objective value solution again.

4.7 DISCUSSION

4.7.1 COMBINED RISK AND TIME

In the case considered the primary benefit of using a robust path planning approach over one that doesn't account for uncertainty is that the resulting paths will not increase in risk due to the effects of wind. Consequently a side effect of this is that the solutions found by the robust approach will have a higher risk values than would be found by a deterministic approach. In the case considered, for the pure risk weighting on the objective this actually leads to a solution that does better on the time objective. The reason for this is that the robust solution avoids the cost of small detours in the deterministic solution that make the path longer, but lower the risk of the path when not accounting for uncertainty. It is expected that cases should also exist where the time objective should increase due to detours being needed for a more robust solution, but we did not consider such a case in the study conducted here.

It can be observed that the robust path planning solutions follow a similar shape in their Pareto frontier compared to the deterministic solutions. This structure is also fairly similar to the Pareto frontiers observed in Chapters 2 and 3, with a risk optimal solution spaced out away from the rest of the solutions, which then quickly move towards a vertical asymptote at the time optimal solution. The solutions near the time optimal solution tend to be paths that are fairly close to the time optimal solution, so it makes sense that these solutions would not be significantly changed by introducing uncertainty into the risk objective.

4.7.2 PURE RISK

The large number of objective function calls needed for the pure risk case is one of the main limitations of the approach discussed here, as it causes the approach to take an extremely long time to optimize that particular case. A possible method for better handling this issue in cases like in the pure risk case would be to make the maximization problem solved in the global search step of the approach a global maximization problem rather than a local one. While global optimization methods can typically not truly guarantee a globally optimal solution, they can get near one when run for a finite number of iterations, by running such a method for a fixed number of iterations the global worst case for a path could be found more quickly, minimizing the number of times a path might get considered again due to not having reached its global worst case. Additionally the implementation of the approach used in order to generate the results seen here has a number of inefficiencies built into it that existed for the purposes of being able to effectively test out code. Future implementations of this approach should be able to run significantly more quickly, which will affect not only the speed of slow cases such as the pure risk case but also all of the other cases as well.

CHAPTER 5: CONCLUSION

5.1 SUMMARY

We have explored the problem of risk-based path planning for UAVs and developed a number of approaches for solving different types of risk-based path planning problems. In order to quantify risk we simulated the dynamics of a UAV crashing under a variety of conditions in order to determine the distribution of possible crash locations of the UAV, which we used to define a metric for risk-based off the expected number of fatalities along a flight path. Using this risk metric we detailed a weighting method which allows for multiobjective studies comparing the risk posed by a UAV flying over a path against the time needed to fly along the path. The results of several different path planning approaches were compared in terms of solution quality and computational performance, the approaches considered were a network based approach, a non-network approach based off optimizing waypoint locations and two hybrid approaches which considered different strategies for optimizing the locations on waypoints generated by the network based approach. It was found that an approach based off greedily refining the path produced by the network approach had the best overall performance in terms of solution quality and computational time.

The problem of combined design and path planning was also explored using similarly structured problem. A model correlating a limited number of design variables to the crash distribution of a UAV was developed using a Delaunay triangulation. An approach was detailed for solving combined design and path planning problems using a network based path planning approach while still allowing for a continuously valued representation to be used for the design. The results from this approach were compared to an integrated approach using continuous variables to represent both the design variables and the path being optimized. It was found that the proposed approach for solving combined design and path planning problems was more effective than the integrated approach considered.

The approach developed for solving combined design and path planning problems was then adapted and extended in order to solve the problem of robust risk-based path planning, taking into account the effects of uncertainty about the wind speed when determining the optimal paths. Results for the robust risk-based path planning approach were presented and compared with the results produced by a deterministic path planning approach that did not consider any uncertainty.

5.2 CONTRIBUTIONS

1. We developed a risk metric for representing the risk posed by UAVs to third parties on the ground that can be used to quantify the risk associated with a specific flight path.
2. Methods were developed for determining the distribution of possible crash locations of a UAV in order to aid in computation of this metric.
3. Techniques were developed for determining not only risk optimal paths subject to the risk metric developed, but also for determining the multiobjective trade-offs between the risk of a path and the time needed to traverse it.

4. The topic of combined design and path planning was also explored in the context of multiobjective risk-based path planning for UAVs.
5. A MDO based approach was developed for solving this problem by decomposing the problem into a graph based path planning optimization subproblem and a design optimization subproblem.
6. The results of the MDO approach developed were compared with an all-in-one approach that considered both path planning and design optimization as a single problem.
7. An approach for solving robust risk-based path planning problems was developed based off the approach developed for solving combined design and path planning problems.

5.3 CONCLUSIONS

Throughout this thesis, there has been a consistent trend of graph based techniques for path planning related problems outperforming waypoint optimization based techniques in terms of global optimality. Risk-based path planning problems contain a large number of local optima due to the nature of how the population in a region is distributed, making it very important that risk-based path planning approaches be able to avoid getting trapped in local optima. In both the work discussed here on risk-based path planning and combined design and path planning, these local optima typically compromise the performance of approaches that rely purely on local optimization, such as the non-network approach for risk-based path planning or the integrated approach for combined design and path planning. This loss in performance is most noticeable for when these approaches are being used for purely risk-based path planning, as the globally optimal solution often needs to go significantly out of the way in order to reach a destination, leaving local optimization based approaches to get trapped at local optima before getting near that solution. Use of local optimization as a technique for refining the solutions of graph based path planners, which are limited to a discrete set of edges for where they can plan paths demonstrated the ability to improve the quality of the solutions found. Furthermore, using strategies such as the one employed by the greedy approach for risk-based path planning, it was possible to obtain these benefits without significantly increasing the time needed to solve the planning problem. In the context of combined design and path planning and the approach developed for robust path planning, the combination of an optimization problem being solved via local search with a graph based path planner allowed for complex problems to be represented in a relatively simple way, without incurring a large number of local optima. These approaches utilize a stronger concept of optimality compared to the results from the greedy and local approaches for risk-based path planning, as the approaches based off the algorithm for combined design and path planning are based around solving both the local optimization and the graph planning problem optimally at the same time. This is not an issue for risk-based path planning, as both the graph search and local search are solving the same problem, however this optimality is more important when the graph search and local search aren't solving the same problem, such as in the combined design and path planning and robust path planning approaches. The advantages of taking this into account are demonstrated in the experiments conducted on the robust path planning approach, as the problem considered would have an enormous number of local optima if considered using an approach similar to the integrated approach used for combined design and planning in Chapter 3. By using an approach that could actually avoid dealing with the local optima from the path planning aspects of the problem it was possible to limit the global searching needed to just the uncertain variables, allowing for a computationally feasible approach to solving the problem.

5.4 FUTURE WORK

A number of possible extensions to the work detailed in this thesis exist. While risk-based path planning considered here primarily focuses on the task of determining routes that are low risk, there are a number of different actions that can be taken while actually traversing routes which can further mitigate risk. An example of this would be identifying areas where large numbers of people are gathered in real time so as to avoid flying over crowds. By incorporating the possibility of cases like this occurring into the task of route planning, it becomes possible to take into account the need to avoid such risks in real time when initially determining a route. Another type of uncertainty that can cause a similar type of issue is inclement weather, which can force a UAV to have to change routes. Planning while taking these types of factors into account would fall under the topic of feedback planning, determining not only an optimal route, but also what should be done if certain scenarios crop up when actually following the route. Many of these factors also intersect with the goals of robust risk-based path planning, thus it would likely be practical to pursue these issues from the perspective of developing a robust feedback planning approach. Unlike a conventional feedback planning approach that typically uses probabilities in order to determine the expected costs of a plan, a robust feedback planning approach would need to determine a method for optimizing the costs across the wide variety of scenarios that could occur without assuming any form of probability distribution for the scenarios.

The approach developed here for the problem of combined path planning and design optimization can likely be extended to the more general problem of combined planning and design optimization. A number of problems of considerable interest can fall under this topic, such as optimizing both the design of a part and how the part is being manufactured, or integrating task planning problems with motion planning problems. While the approach discussed here can be applied to these problems to an extent, it is limited by the fact that the constraints present in these problems must be independent of any planning decisions that are made. There a large number of planning problems where the decisions that make up the plan affect the constraints that apply to the problem, thus this is a key limitation of the approach discussed here, and consequently an important area for future research on this type of problem.

REFERENCES

[1] Matlab and optimization toolbox release 2014b. Technical report, The MathWorks Inc., 2014.

[2] Hans Georg Beyer and Bernhard Sendhoff. Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.

[3] Scott A Bortoff. Path planning for uavs. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 364–368. IEEE, 2000.

[4] David Burke. *System Level Airworthiness Tool: A Comprehensive Approach to Small Unmanned Aircraft System Airworthiness*. PhD thesis, North Carolina State University, 2010.

[5] Jose A Cobano, Roberto Conde, David Alejo, and Anbal Ollero. Path planning based on genetic algorithms and the monte-carlo method to avoid aerial vehicle collisions under uncertainties. Ín *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4429–4434. IEEE, 2011.

[6] Navid Dadkhah and Bérénice Mettler. Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 65(1-4):233–246, 2012.

[7] Luca De Filippis, Giorgio Guglieri, and Fulvia Quagliotti. A minimum risk approach for path planning of uavs. *Journal of Intelligent & Robotic Systems*, 61(1-4):203–219, 2011.

[8] Jesus Manuel de la Cruz, Eva Besada-Portas, Luis Torre-Cubillo, Bonifacio Andres-Toro, and Jose Antonio Lopez-Orozco. Evolutionary path planner for uavs in

realistic environments. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1477–1484. ACM, 2008.

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

[10] Andrew Ford and Kevin McEntee. Assessment of the risk to ground population due to an unmanned aircraft in-flight failure. In *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Fort Worth, TX, 2010.

[11] C Goerzen, Zhaodan Kong, and Bernard Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65–100, 2010.

[12] Josef Kallrath. Solving planning and design problems in the process industry using mixed integer and global optimization. *Special Edition of Annals of Operations Research*, 2004.

[13] G.B. Lamont, J.N. Slear, and K. Melendez. Uav swarm mission planning and routing using multi-objective evolutionary algorithms. In *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, pages 10–20, April 2007.

[14] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[15] Christopher Lum, Kristoffer Gauksheimy, Chris Deseure, Juris Vagnersx, and Tad McGeer. Assessing and estimating risk of operating unmanned aerial systems in populated areas. Virginia Beach, VA, 2011.

[16] Christopher W Lum and Blake Waggoner. A risk based paradigm and model for unmanned aerial vehicles in the national airspace. *Infotech@ Aerospace 2011*, pages 2011–1424, 2011.

[17] MATLAB. *Optimization Toolbox User's Guide, Version R2012b*. Natick, MA, 2012.

[18] MATLAB. *version R2012b*. The MathWorks Inc., Natick, MA, 2012.

[19] Felipe Leonardo Lôbo Medeiros and José Demisio Simoes Da Silva. Computational modeling for automatic path planning based on evaluations of the effects of impacts of uavs on the ground. *Journal of Intelligent & Robotic Systems*, 61(1-4):181–202, 2011.

[20] Felipe Leonardo Lôbo Medeiros and José Demisio Simoes Da Silva. Computational modeling for automatic path planning based on evaluations of the effects of impacts of uavs on the ground. *Journal of Intelligent & Robotic Systems*, 61(1-4):181–202, 2011.

[21] Shashi Mittal and Kalyanmoy Deb. Three-dimensional offline path planning for uavs using multiobjective evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation (CEC-2007),(Singapore)*, pages 3195–3202. Proceedings of the Congress on Evolutionary Computation (CEC-2007),(Singapore), 2007.

[22] Pritesh Narayan, Duncan Campbell, and Rodney Walker. Computationally adaptive multi-objective trajectory optimization for uas with variable planning deadlines. *IEEEAC*, 2009.

[23] Nikhil Nigam and Ilan Kroo. Control and design of multiple unmanned air vehicles for a persistent surveillance task. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference. Victoria, British Columbia, AIAA-2008-5913*, 2008.

[24] AJ Pikaar, CJM De Jong, and J Weijts. *An Enhanced Method for the Calculation of Third Party Risk Around Large Airports: With Application to Schiphol.* Nationaal Lucht-en Ruimtevaartlaboratorium, 2000.

[25] Jahangir S Rastegar, Lidong Liu, and Dan Yin. Task-specific optimal simultaneous kinematic, dynamic, and control design of high-performance robotic systems. *Mechatronics, IEEE/ASME Transactions on*, 4(4):387–395, 1999.

[26] Line Blander Reinhardt and David Pisinger. Multi-objective and multi-constrained non-additive shortest path problems. *Computers & Operations Research*, 38(3):605–616, 2011.

[27] Jan Roskam. *Airplane flight dynamics and automatic flight controls*, chapter AIRPLANE DATA. DARcorporation, Lawrence, KS, 1995.

[28] E Rudnick-Cohen, J Herrmann, and S Azarm. Risk-based path planning optimization methods for uavs over inhabited area. In *To appear in Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2015*, 2015.

[29] Eliot Rudnick-Cohen, Shapour Azarm, and Jeffrey W Herrmann. Multi-objective design and path planning optimization of unmanned aerial vehicles (uavs). In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 2322, 2015.

[30] Eliot Rudnick-Cohen, Jeffrey W. Herrmann, and Shapour Azarm. Risk-based path planning optimization methods for unmanned aerial vehicles over inhabited areas1. *Journal of Computing and Information Science in Engineering*, 16(2):021004–021004, Apr 2016.

[31] Glenn Sanders and Tapabrata Ray. Optimal offline path planning of a fixed wing unmanned aerial vehicle (uav) using an evolutionary algorithm. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 4410–4416. IEEE, 2007.

[32] Robert Stengel. *Flight dynamics*, chapter 3. Princeton University Press, Princeton, NJ, 2004.

[33] Brian Stevens. *Aircraft control and simulation*, chapter 3. J. Wiley, Hoboken, N.J, 2003.

[34] Roland Everett Weibel. Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system. Master's thesis, Massachusetts Institute of Technology, 2005.

[35] Jin Wu and Shapour Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Journal of Mechanical Design*, 123(1):18–25, 2001.

[36] Paul P Wu and Reece A Clothier. The development of ground impact models for the analysis of the risks associated with unmanned aircraft operations over inhabited areas. In *Proceedings of the 11th Probabilistic Safety Assessment and Management Conference (PSAM11) and the Annual European Safety and Reliability Conference (ESREL 2012)*, 2012.