

## RISK-BASED PATH PLANNING OPTIMIZATION METHODS FOR UAVs OVER INHABITED AREAS

**Eliot Rudnick-Cohen**  
University of Maryland  
College Park, Maryland 20742  
Email: erudnick@umd.edu

**Jeffrey W. Herrmann**  
University of Maryland  
College Park, Maryland 20742  
Email: jwh2@umd.edu

**Shapour Azarm**  
University of Maryland  
College Park, Maryland 20742  
Email: azarm@umd.edu

### ABSTRACT

*Operating unmanned aerial vehicles (UAVs) over inhabited areas requires mitigating the risk to persons on the ground. Because the risk depends upon the flight path, UAV operators need approaches (techniques) that can find low-risk flight paths between the mission's start and finish points. 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. Although there exist approaches for assessing the risks associated with UAV operations, existing risk-based path planning approaches have considered other risk measures (besides the risk to persons on the ground) or simplified the risk assessment calculation. This paper presents a risk assessment technique and bi-objective optimization methods to find low-risk and time (flight path) solutions and computational experiments to evaluate the relative performance of the methods (their computation time and solution quality). The methods were a network optimization approach that constructed a graph for the problem and used that to generate initial solutions that were then improved by a local approach and a greedy approach and a fourth method that did not use the network solutions. The approaches that improved the solutions generated by the network optimization step performed better than the optimization approach that did not use the network solutions.*

### 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  
 $(x^S, y^S)$  start point of the flight plan  
 $(x^F, y^F)$  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  
 $\Delta_x$  horizontal distance between adjacent nodes

	(vertices) in the grid
$\Delta_y$	vertical distance between adjacent nodes
	(vertices) in the grid
$(\Delta x_{jk}, \Delta y_{jk})$	rotated coordinates of a point
	in the bivariate distribution
$\Gamma_k$	census tract polygon

## 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 [1]). 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 [2]. 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 [3][4][5], risk posed due to large scale obstacles such as radar or heavily populated areas [6][7], the risk of a mid-air collision [8][9][10] or the risks to persons on the ground [7].

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 [11]. Many mathematical optimization techniques for risk-based planning utilize evolutionary optimization algorithms [12] [13].

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 [1]. 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. [14] 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 [15], which can be used as a distribution with the assumption of a uniform distribution in those bounds. Ford and McEntee [16] generated a bivariate crash location distribution using simple assumptions about the flight dynamics of an unpowered UAV. Lum et al. [17] 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 authors of this paper are 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 paper 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. 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  $(x_0, y_0) = (x^S, y^S)$ , and the last waypoint is the end point  $(x_{n+1}, y_{n+1}) = (x^F, y^F)$ .

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.

### 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.

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 [18] and [19]. By solving this non-linear ODE numerically using MATLAB's ode45 solver [20] 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 1. 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 1. The aerodynamic coefficients and physical properties used for the UAV being simulated were based on those provided for a Cessna 182 aircraft [21]. For the purposes of generating the crash distribution used in the results presented in this paper, 10,000 simulation runs were conducted. 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. 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.

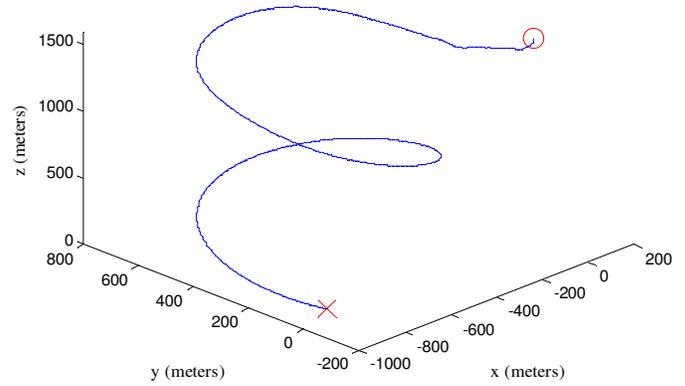


Figure 1. EXAMPLE UAV CRASH TRAJECTORY. THE CIRCLE DENOTES THE START POINT AND THE "x" DENOTES THE FINAL CRASH LOCATION.

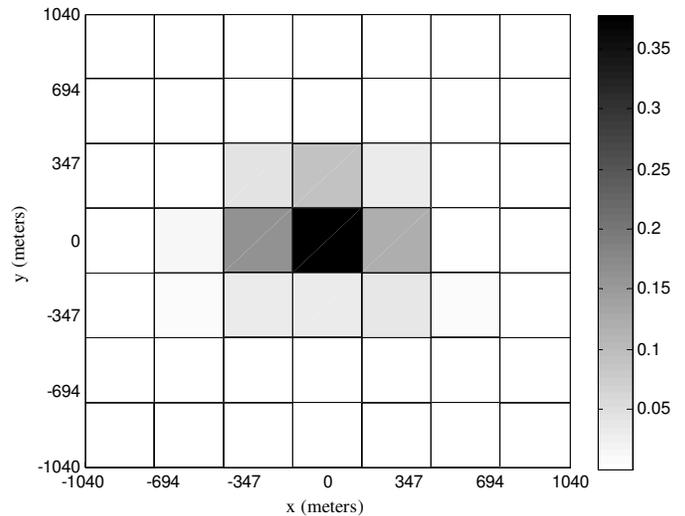


Figure 2. DISCRETIZED HEAT MAP OF CRASH DENSITY DISTRIBUTION. THE SCALE CORRESPONDS TO THE PROBABILITY THAT THE VEHICLE WILL LAND IN THAT CELL.

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.

Table 1. INITIAL CONDITIONS FOR MONTE CARLO SIMULATIONS.

Velocity (m/s)	Mean	Deviation
$\dot{x}$	50	50
$\dot{y}$	0	10
$\dot{z}$	0	10
<b>Position (m)</b>		
$x$	0	0
$y$	0	0
$z$	1,524	0
<b>Orientation, Euler angles (degrees)</b>		
$\Phi$	0	11.25
$\Theta$	0	11.25
$\Psi$	0	11.25
<b>Angular Velocity (degrees/s)</b>		
P	0	11.25
Q	0	11.25
R	0	11.25
<b>Control surface deflection (degrees)</b>		
Elevator Deflection ( $\delta_E$ )	0	11.25
Rudder Deflection ( $\delta_R$ )	0	11.25
Aileron Deflection ( $\delta_A$ )	0	11.25
<b>Control surface deflection rates (degrees/s)</b>		
Elevator deflection rate ( $\dot{\delta}_E$ )	0	0
Rudder deflection rate ( $\dot{\delta}_R$ )	0	0
Aileron deflection rate ( $\dot{\delta}_A$ )	0	0

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  $a = 1, \dots, m$ , do the following:

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

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

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

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

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

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

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

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

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

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

Step 4. Loop over the census tracts. For each census tract  $k$ , determine which points in the “cloud” are in that tract’s polygon  $\Gamma_k$  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  $k$ :

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

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

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

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.

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

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  $w_t + w_r = 1$ :

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

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 paper.

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

### 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  $(x^S, y^S)$  and  $(x^F, y^F)$ , 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 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:

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

The network optimization approach found the minimal cost path  $X^N$  using Dijkstra's algorithm [22]. 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.

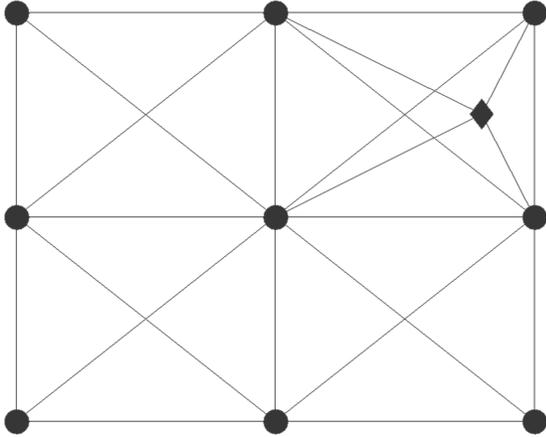


Figure 3. IN THIS SECTION OF THE GRID, THE SOLID CIRCLES ARE NODES IN THE GRID, THE DIAMOND IS THE START (OR FINISH) POINT, AND THE ADDITIONAL EDGES SHOW HOW THAT POINT IS CONNECTED TO THE NODES IN THE GRID.

### 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  $f_y$ :

$$\begin{aligned} x_i^N - f_x \Delta_x &\leq x_i \leq x_i^N + f_x \Delta_x \\ y_i^N - f_y \Delta_y &\leq y_i \leq y_i^N + f_y \Delta_y \end{aligned} \quad (7)$$

Pseudocode:

$$X_0 = X^N$$

$$X_{sol} = \text{Minimize}(f(X), X_0, \text{Eq. 7})$$

### 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  $i = 1 \dots n$

$$X_{sol}[i] = \text{Minimize}(f(X[i]), X_0, \text{Eq.7})$$

$$X_0[i] = X_{sol}[i]$$

End

### 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.

## 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  $w_t = 0, 0.1, 0.2, \dots, 1.0$ , 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:  $30 \times 12, 40 \times 16, 50 \times 20$ . (For example, the  $30 \times 12$  grid began with 360 nodes arranged in 30 columns and 12 rows.) Examples of the types of grids used can be seen in Figure 4 and Figure 5. 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 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` [23] function was used to solve the continuous optimization problems in the local improvement, greedy improvement, and non-network-based approaches.

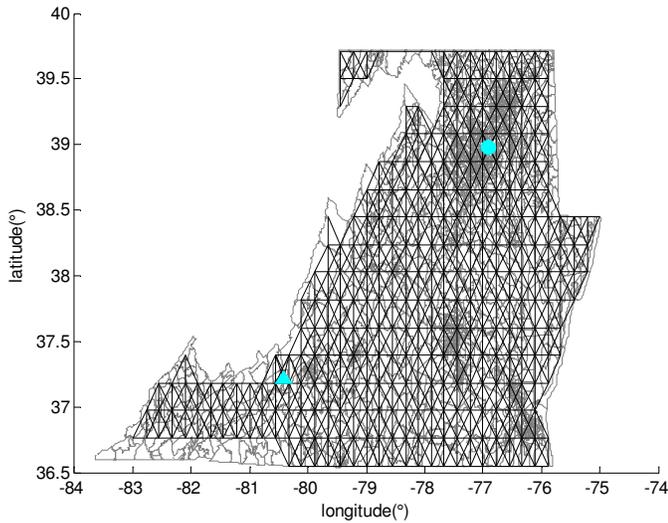


Figure 4. THE 40x16 NETWORK (THE “GRID”) FOR THE COLLEGE PARK CASE. THE GRAY LINES SHOW THE CENSUS TRACTS IN VIRGINIA, MARYLAND, AND THE DISTRICT OF COLUMBIA. THE BLACK LINES SHOW THE EDGES IN THE GRID. THE CYAN CIRCLE SHOWS THE START POINT (COLLEGE PARK, MD), THE CYAN TRIANGLE IS THE END POINT (BLACKSBURG, VA).

Table 2. HORIZONTAL AND VERTICAL EDGE LENGTHS FOR DIFFERENT CASES CONSIDERED.

Case	Grid size	Horizontal edge length (°longitude) ( $\Delta_x$ )	Vertical edge length (°latitude) ( $\Delta_y$ )
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 [24] 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 6 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).

## 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

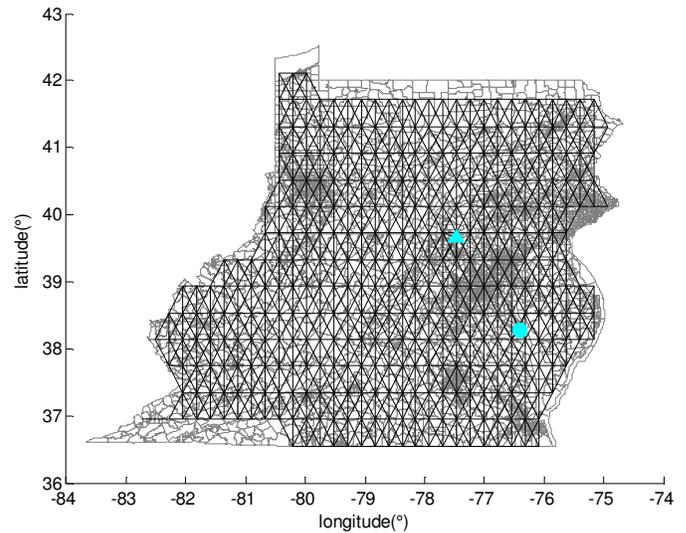


Figure 5. THE 40x16 NETWORK (THE “GRID”) FOR THE PAX RIVER CASE. THE GRAY LINES SHOW THE CENSUS TRACTS IN PENNSYLVANIA, DELAWARE, VIRGINIA, WEST VIRGINIA, MARYLAND, AND THE DISTRICT OF COLUMBIA. THE BLACK LINES SHOW THE EDGES IN THE GRID. THE CYAN CIRCLE SHOWS THE START POINT (PAX RIVER, MD), THE CYAN TRIANGLE IS THE END POINT (CAMP DAVID, MD).

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 11 and 12 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.

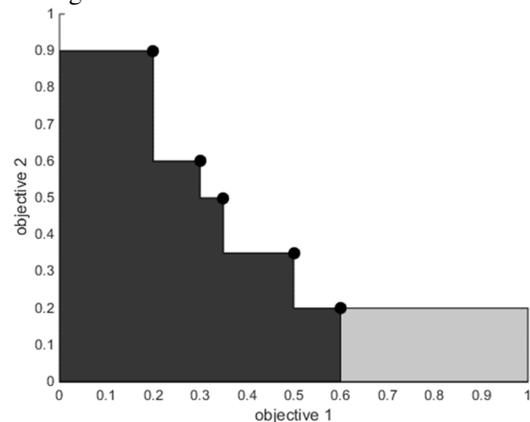
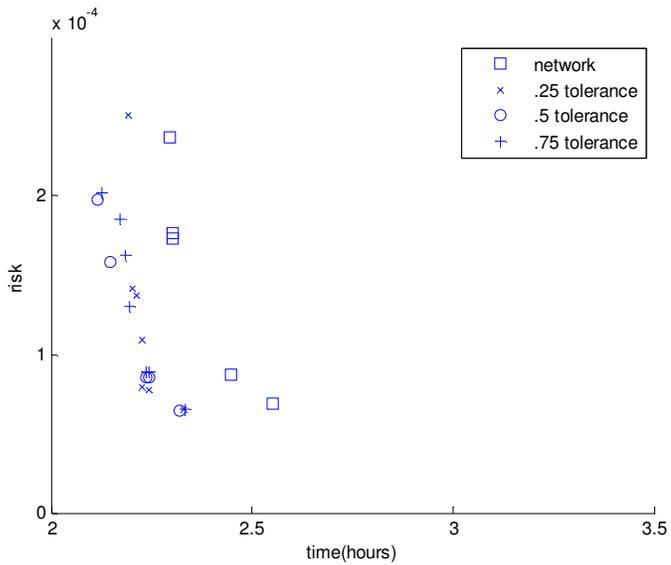
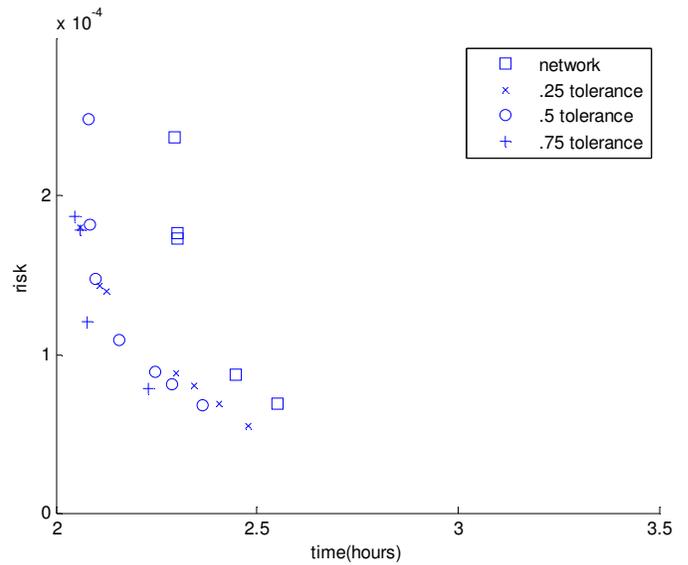


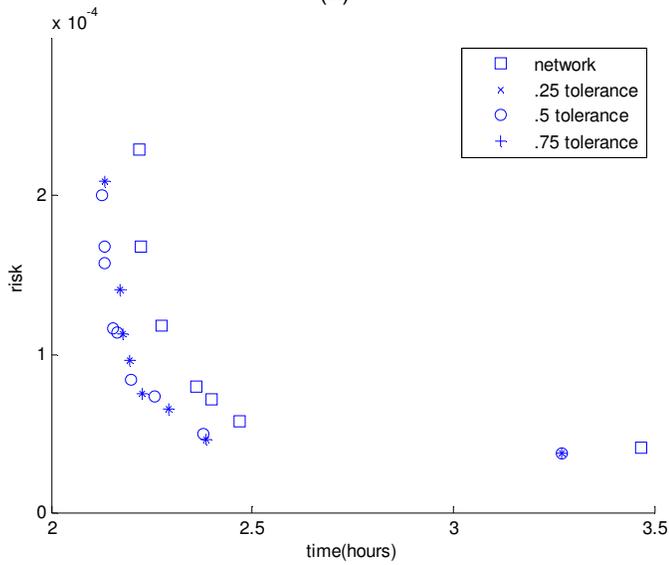
Figure 6. EXAMPLE OF CLOSNESS METRIC, THE BLACK POINTS ARE THE PARETO FRONTIER, THE BLACK AREA SHOWS THE AREA CONSIDERED IN THE METRIC FROM [24], AND THE GRAY AREA SHOWS THE ADDITIONAL AREA THAT IS CONSIDERED BY THE METRIC DETAILED HERE.



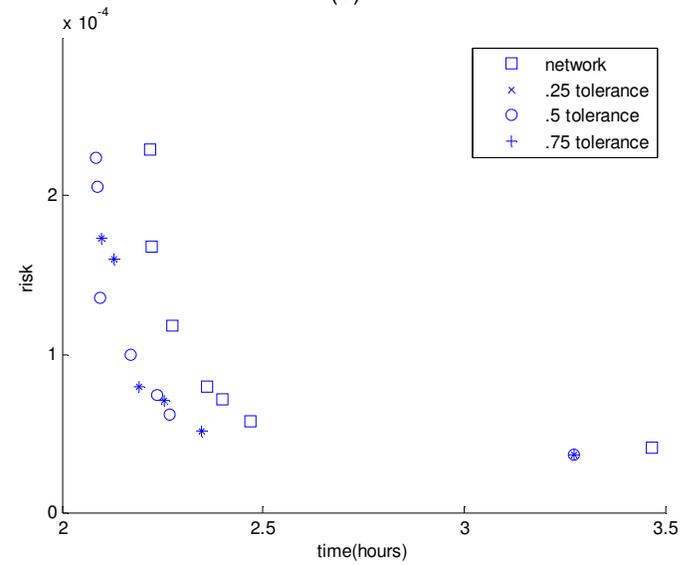
(a)



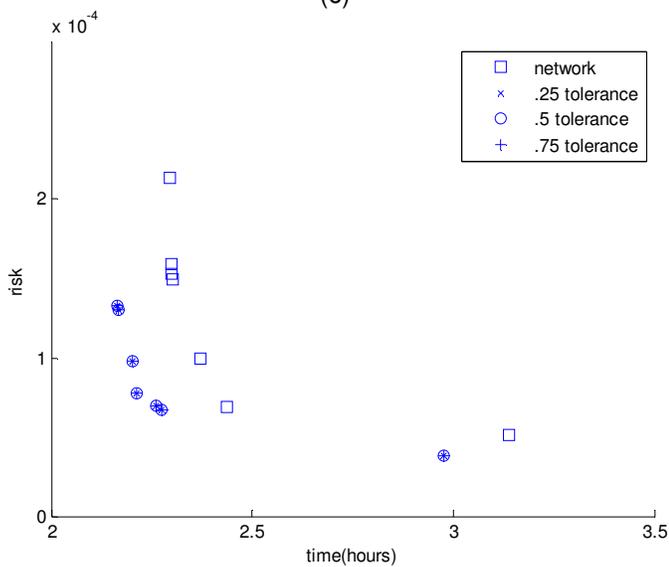
(b)



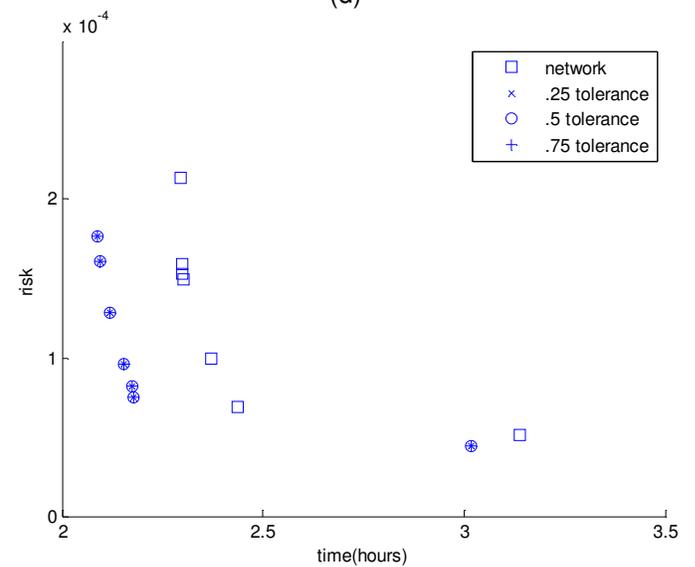
(c)



(d)

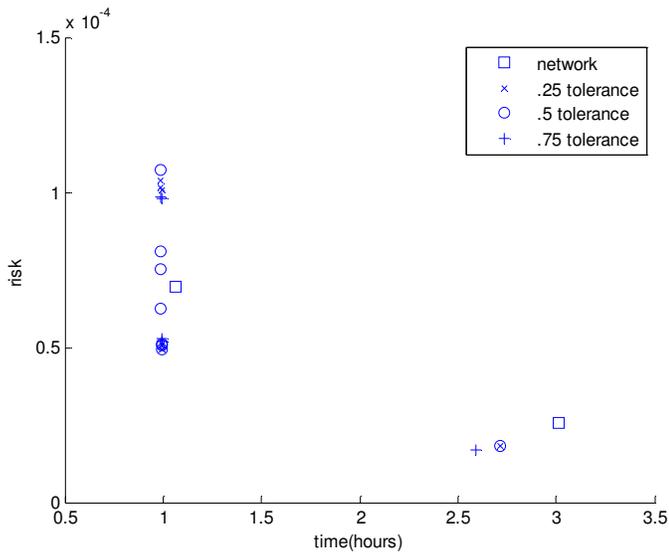


(e)

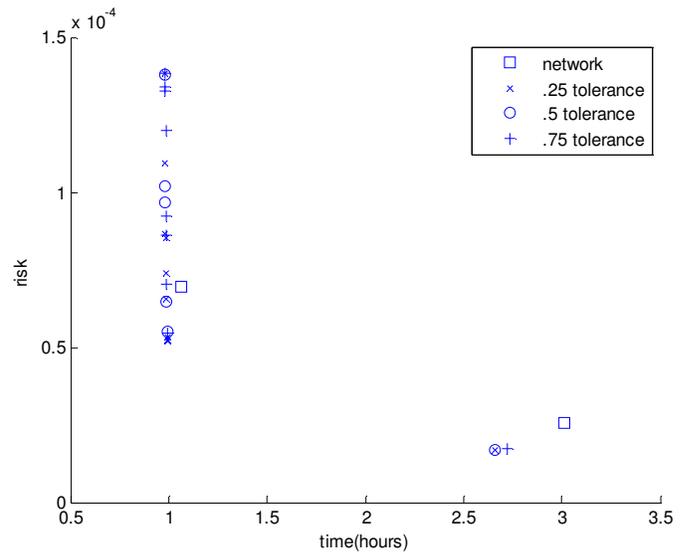


(f)

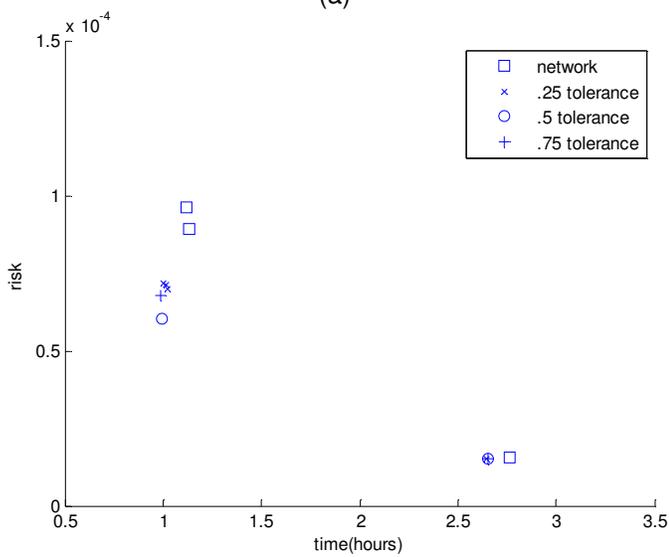
Figure 7. PARETO FRONTIER RESULTS FOR THE COLLEGE PARK CASE: (a) GREEDY APPROACH, 30x12 GRID, (b) LOCAL APPROACH, 30x12 GRID, (c) GREEDY APPROACH, 40x16 GRID, (d) LOCAL APPROACH, 40x16 GRID, (e) GREEDY APPROACH, 50x20 GRID, (f) LOCAL APPROACH, 50x20 GRID.



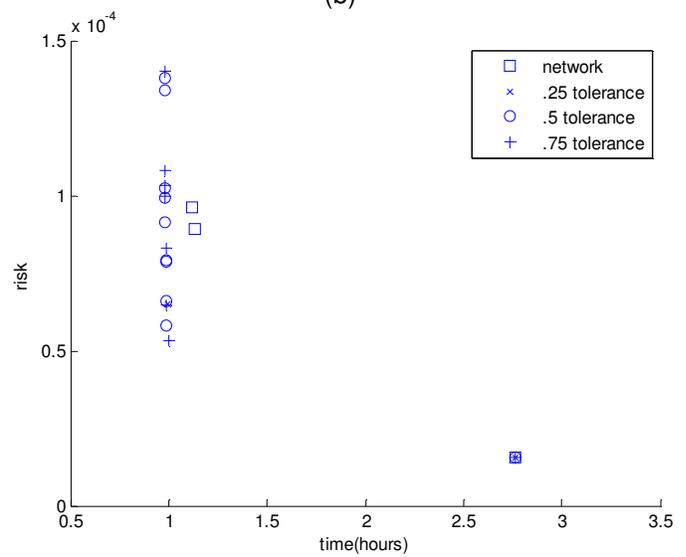
(a)



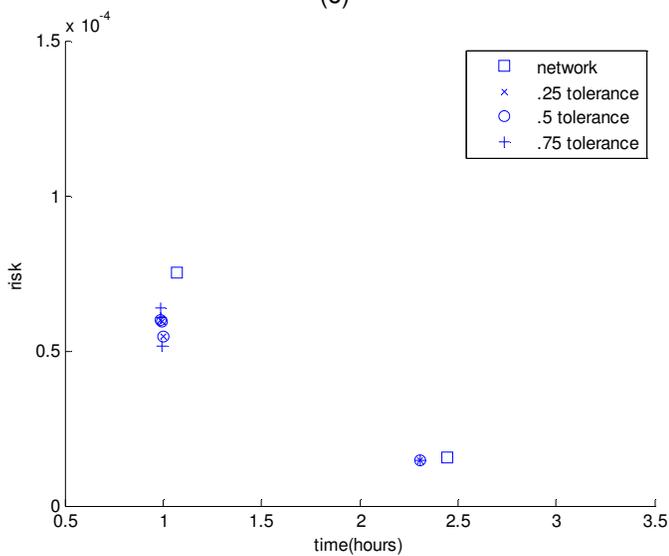
(b)



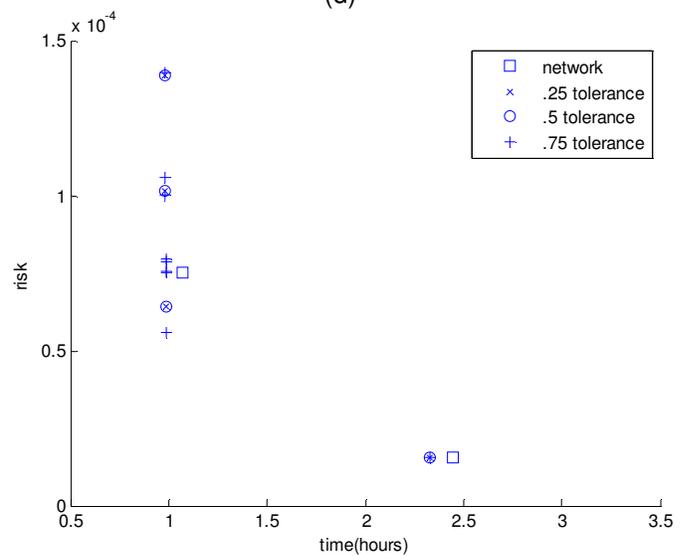
(c)



(d)



(e)



(f)

Figure 8. PARETO FRONTIER RESULTS FOR PAX RIVER CASE: (a) GREEDY APPROACH, 30x12 GRID, (b) LOCAL APPROACH, 30x12 GRID, (c) GREEDY APPROACH, 40x16 GRID, (d) LOCAL APPROACH, 40x16 GRID, (e) GREEDY APPROACH, 50x20 GRID, (f) LOCAL APPROACH, 50x20 GRID.

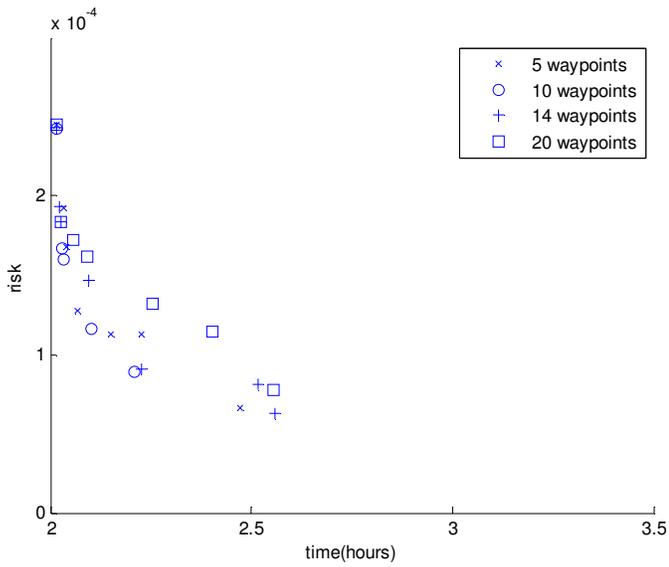


Figure 9. PARETO FRONTIER RESULTS FOR THE COLLEGE PARK CASE USING THE NON-NETWORK METHOD FOR DIFFERENT NUMBERS OF WAYPOINTS.

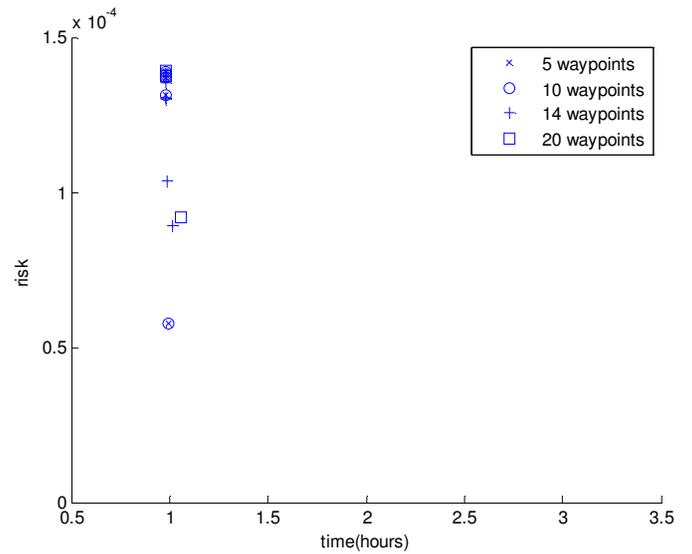


Figure 10. PARETO FRONTIER RESULTS FOR THE PAX RIVER CASE USING THE NON-NETWORK METHOD FOR DIFFERENT NUMBERS OF WAYPOINTS.

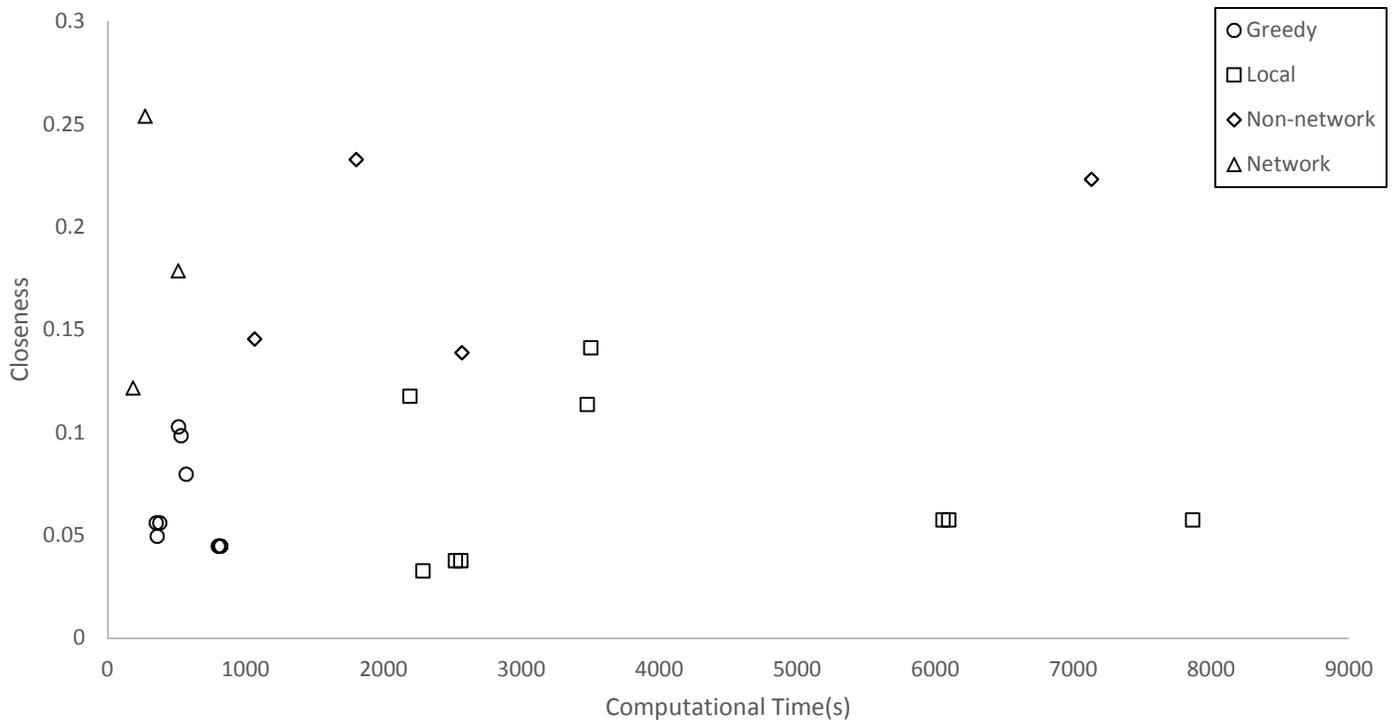


Figure 11. CLOSENESS AGAINST COMPUTATION TIME FOR THE COLLEGE PARK CASE.

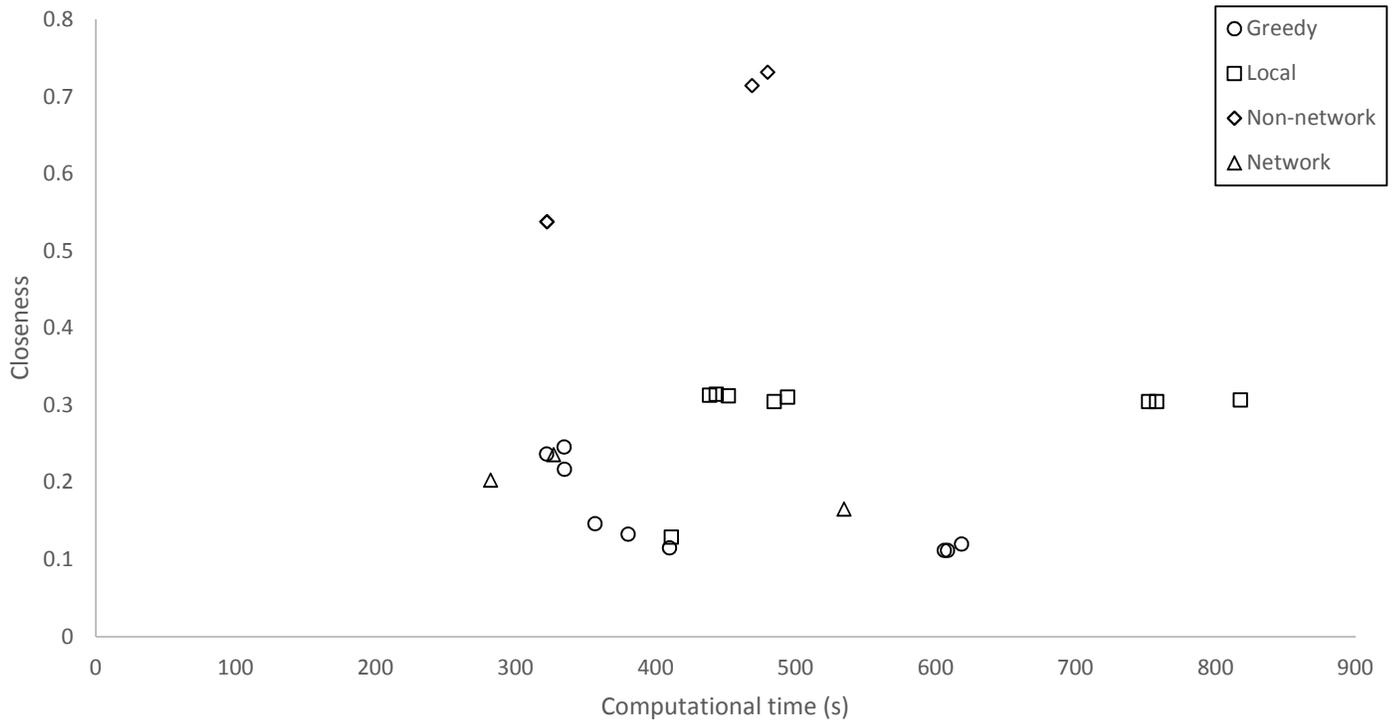


Figure 12. CLOSENESS AGAINST COMPUTATION TIME FOR THE PAX RIVER CASE.

The results displayed in Figure 7 and Figure 8 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 7. 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 9.

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 8). 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 9 and Figure 10.

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 Figure 7 and Figure 8, 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 9 and Figure 10 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 13. The greedy and local approaches appear to be the best of the approaches that were considered in this paper (that is, they produced the best Pareto frontiers of solutions).

As can be seen in Figure 11 and Figure 12, 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

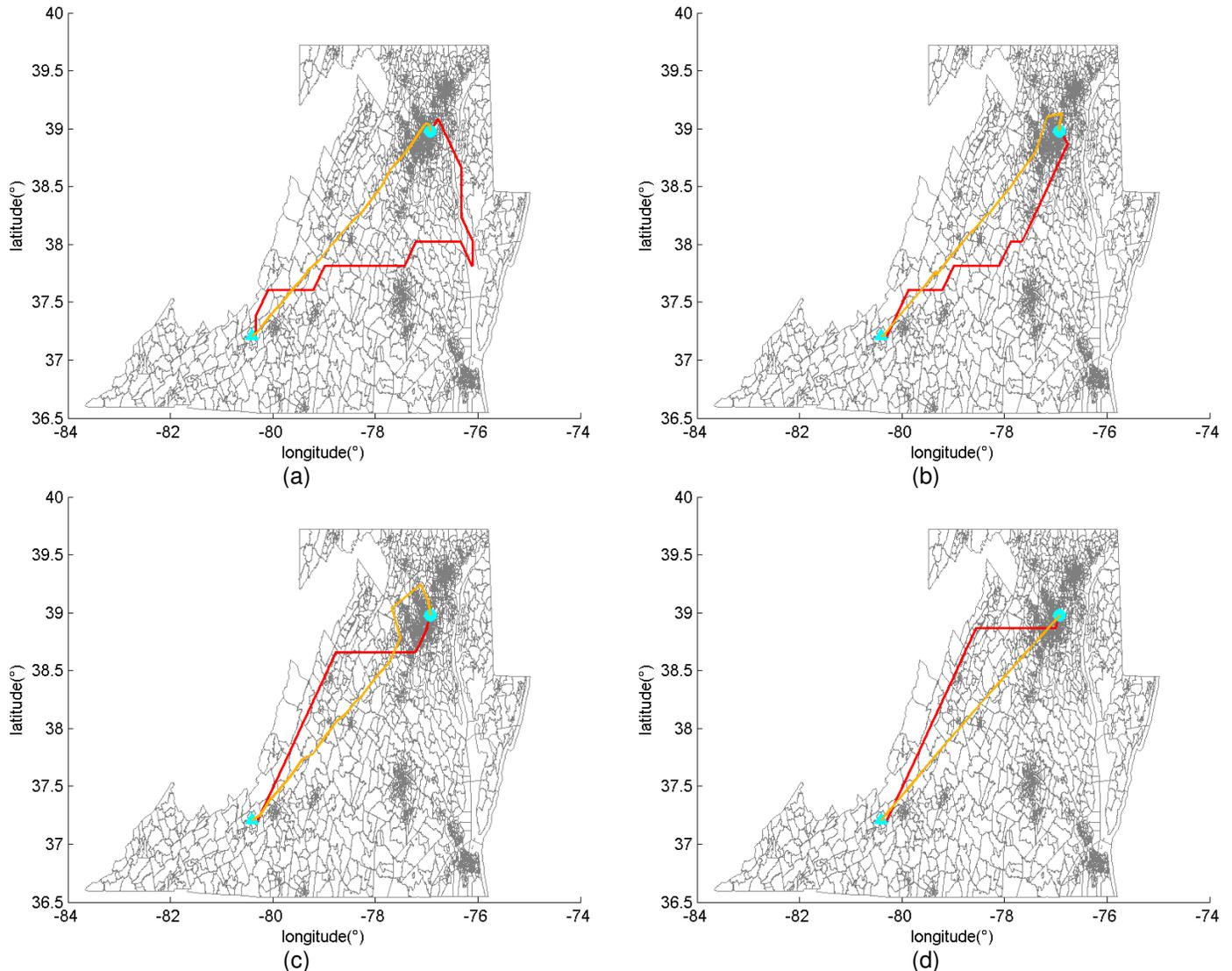


Figure 13. EXAMPLES OF THE SOLUTIONS GENERATED BY THE NETWORK APPROACH WITH THE 40X16 GRID (RED) AND THE NON-NETWORK APPROACH WITH 20 WAYPOINTS (ORANGE) FOR THE COLLEGE PARK CASE: (a)  $w_t = 0, w_r = 1$  (b)  $w_t = 0.3, w_r = 0.7$  (c)  $w_t = 0.6, w_r = 0.4$  (d)  $w_t = 1, w_r = 0$ .

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.

## 6. CONCLUDING REMARKS

This paper 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.

## 7. ACKNOWLEDGMENTS

The authors acknowledge Dr. David Burke and Dr. John Tritschler and the financial support of the Naval Air Warfare Center under cooperative agreement N00421132M006. Such support does not constitute an endorsement by the funding agency of the opinions expressed in the paper.

## REFERENCES

- [1] Burke, D., 2010. "System level airworthiness tool: A comprehensive approach to small unmanned aircraft system airworthiness". PhD Thesis, North Carolina State University, Raleigh, NC.
- [2] Goerzen, C., Kong, Z., and Mettler, B., 2010. "A survey of motion planning algorithms from the perspective of autonomous UAV guidance". *Journal of Intelligent and Robotic Systems*, **57**(1-4), pp. 65–100.
- [3] Mittal, S., and Deb, K., 2007. "Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms". In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, IEEE, Singapore, pp. 3195–3202.
- [4] Sanders, G., and Ray, T., 2007. "Optimal offline path planning of a fixed wing unmanned aerial vehicle (UAV) using an evolutionary algorithm". In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, IEEE, Singapore, pp. 4410–4416.
- [5] De Filippis, L., Guglieri, G., and Quagliotti, F., 2011. "A minimum risk approach for path planning of UAVs". *Journal of Intelligent & Robotic Systems*, **61**(1-4), pp. 203–219.
- [6] Bortoff, S. A., 2000. "Path planning for UAVs". In Proceedings of the 2000 American Control Conference, Vol. 1, IEEE, Chicago, IL, pp. 364–368.
- [7] Medeiros, F. L. L., and Da Silva, J. D. S., 2011. "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), pp. 181–202.
- [8] Weibel, R. E., 2005. "Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system". Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.
- [9] Lum, C. W., and Waggoner, B., 2011. "A risk based paradigm and model for unmanned aerial vehicles in the national airspace". *Infotech@ Aerospace 2011*, pp. 2011–1424.
- [10] Cobano, J. A., Conde, R., Alejo, D., and Ollero, A., 2011. "Path planning based on genetic algorithms and the monte-carlo method to avoid aerial vehicle collisions under uncertainties". In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, IEEE, Shanghai, China, pp. 4429–4434.
- [11] Reinhardt, L. B., and Pisinger, D., 2011. "Multi-objective and multi-constrained non-additive shortest path problems". *Computers & Operations Research*, **38**(3), pp. 605–616.
- [12] Lamont, G., Slear, J., and Melendez, K., 2007. "UAV swarm mission planning and routing using multi-objective evolutionary algorithms". In IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, IEEE, Honolulu, HI, pp. 10–20.
- [13] de la Cruz, J. M., Besada-Portas, E., Torre-Cubillo, L., Andres-Toro, B., and Lopez-Orozco, J. A., 2008. "Evolutionary path planner for UAVs in realistic environments". In Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, ACM, Atlanta, GA, pp. 1477–1484.
- [14] Pikaar, A., De Jong, C., and Weijts, J., 2000. *An Enhanced Method for the Calculation of Third Party Risk Around Large Airports: With Application to Schiphol*. Nationaal Lucht-en Ruimtevaartlaboratorium.
- [15] Wu, P. P., and Clothier, R. A., 2012. "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), Helsinki.
- [16] Ford, A., and McEntee, K., 2010. "Assessment of the risk to ground population due to an unmanned aircraft in-flight failure". In Proceedings of the 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA, Fort Worth, TX.
- [17] Lum, C., Gauksheimy, K., Deseure, C., Vagnersx, J., and McGeer, T., 2011. "Assessing and estimating risk of operating unmanned aerial systems in populated areas". In Proceedings of the 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA, Virginia Beach, VA.
- [18] Stevens, B., 2003. *Aircraft control and simulation*. J. Wiley, Hoboken, NJ, Chapter 3.
- [19] Stengel, R., 2004. *Flight dynamics*. Princeton University Press, Princeton, NJ, Chapter 3.
- [20] MATLAB, 2012. *version R2012b*. The MathWorks Inc., Natick, MA.
- [21] Roskam, J., 1995. *Airplane flight dynamics and automatic flight controls*. DARcorporation, Lawrence, KS.
- [22] Dijkstra, E. W., 1959. "A note on two problems in connexion with graphs". *Numerische mathematik*, **1**(1), pp. 269–271.
- [23] MATLAB, 2012. *Optimization Toolbox User's Guide, Version R2012b*. Natick, MA.
- [24] Wu, J., and Azarm, S., 2001. "Metrics for quality assessment of a multiobjective design optimization solution set". *Journal of Mechanical Design*, **123**(1), pp. 18–25.