Aircraft Rerouting under Risk Tolerance during Space Launches

Oliver J. Bojorquez *, Nathan Dolan †, Jun Chen‡
San Diego State University, San Diego, CA, 92182-1308

The recent improvements in commercial space transportation have raised questions on methodology used for integrating space launched vehicles into the National Airspace System (NAS) regarding the risks to nearby aircraft. Current safety regulation procedures are used to safely integrate launch vehicles in the NAS by closing large blocks of airspace to any vehicle, this restrictions are known as hazard areas. Regulations cause commercial aircraft to reroute around hazard areas increasing flight distance, causing delays, and raising overall cost per flight. Air space restrictions are designed to decrease risk to the public, where the area and time of restriction are based on space vehicle profile. This paper describes a methodology to dynamically construct a risk level map on nearby aircraft due to space launch operations. The impacted area is divided into multiple sections with each section dynamically evaluated under a risk level, which is a comprehensive index considering debris uncertainty. To speed up the evaluation process for the projectile model, a Graphics Processing Unit (GPU) based parallel computing framework is developed. Using a customized A* algorithm, an aircraft rerouting plan is generated under risk tolerance where a minimal time and distance trajectory can be found. Utilizing SpaceX Falcon 9 as a reference, rerouting models in this project are demonstrated to provide safe and efficient plans by comparing with the current method used.

I. Introduction

To limit the risk to aircraft in the airspace in the case of an explosion during space launch operations, the Federal Aviation Administration (FAA) has a set of licensing procedures which creates large blocks of space considered a no-fly zone. These blocks of space follows the projected vehicle trajectory prohibiting any vehicles from entering this area. Restrictions applied following licensing procedures are temporary flight restrictions covering a vast range, from the launch pad to an altitude of 30 nautical miles to preserve the safety of commercial aircraft. Aviation restrictions cause all aircraft to reroute around hazard areas causing flight delays and increasing flown distance, time, and the total operational cost per flight. During the launch of a SpaceX Falcon 9, restrictions were activated 7 hours prior to the launch and remained in effect 30 minutes after the launch affecting dozens of aircraft. These restrictions affect operational costs including maintenance, fuel, and billable hours. The restrictions also affect external needs, including additional gates and crew at the landing site in order to help maintain efficient air traffic flow [1].

With recent improvements in commercial space transportation, space launches have been occurring more frequently in recent years. The increasing demand for space travel is sure to cause more restricted areas in the future, and the efficiency of the procedures and methodologies that are currently used will be questioned [2]. The current methodology creates a substantial impact on the surrounding air traffic flow, with an estimated cost averaging thousands per flight. According to Airlines of America in 2018, the average cost of aircraft delay was estimated to be $74.20 each minute [3] plus any additional external cost is approximated to be billions of dollars annually. To meet necessary safety demands and reduce aircraft delays during space launch operations, a more flexible and efficient method is essential to maintain a reasonable traffic flow.

According to the FAA Aerospace Forecasts Fiscal Years 2019-2039 [2], the anticipated increase in aircraft is 1.4 percent annually. Having the national airspace containing more vehicles inevitably causes a linear overall cost increase regarding aircraft delays. Due to the increase in the number of aircraft in the airspace, and the substantial demand for space access, it can be surmised that current protocol may not be sufficient to meet the expected increase in aircraft in the airspace.

Historically, the Air Route Traffic Control Centers (ARTCC) determine the required manpower and equipment needed at each facility based on the current forecasts. To maintain an efficient traffic flow, FAA uses a tool known as Traffic Flow Management (TFM) [4]. Air traffic controllers will use the en-route data flow as well as other programs such as Miles In Trail (MIT), Ground Delay Programs (GDP), Ground Stop (GS) and other tools necessary to maintain

*Graduate Student, Department of Aerospace Engineering, Student Member AIAA, obojorquez@sdsu.edu
†Graduate Student, Department of Aerospace Engineering, Student Member AIAA, NDolan@sdsu.edu
‡Assistant Professor, Department of Aerospace Engineering, Member AIAA, jun.chen@sdsu.edu(Corresponding Author)

the most suitable and efficient traffic flow [5]. This decision making process is completed for numerous air traffic difficulties ranging from regular arriving and departing flights to navigating around constrained regions due to historical bad weather conditions. Similar to rerouting around hazard areas, severe weather has been a continuous cause of areas being closed off to incoming air traffic. Rerouting around these areas is accomplished utilizing a collection of known historical routes known as National Playbook, which are well validated and used by the ARTCC.

The National Playbook has indeed facilitated the decision-making process for air traffic personnel, but the TFM remains an area plagued with complex problems prohibiting it from being solved by any singular method. Therefore, constantly relying solemnly on tools similar to The National Playbook may not be the most reliable course of action. Due to the continuous challenges TFM has endured, numerous studies have been done to further improve the current tools, software, and methods that are used to obtain more efficient traffic flow. As launch operations become more frequent, hazard areas created by licenses procedures would demand a more flexible and low risk level TFM tool to improve operational efficiency surrounding hazard areas.

Although the space transport industry is in the early development stage, technology has made vast improvements since the beginning of space travel. Many of the issues that previous launch vehicles had encountered while attempting to access space, have been solved with the new tools and methods that have been employed. The revolutionary idea of a reusable rocket has been demonstrated in the SpaceX Falcon 9, which has already been launched several times and had a significantly lower manufacturing cost than previous operations. Unfortunately, since the space transportation industry is still currently under development, few studies have focused on integration of new era into the NAS. Previous research at Stanford [6–9] presented ideas and potential methods related to this area of study. For instance, Tompa and colleagues [6] used a Markov Decision Process (MDP) to model the uncertainty of possible debris during space launches. To efficiently manage the computational tractability, an adaptive spatial discretization is suggested [7]. Moreover, the work by Colvin T. J. and Alonso, J. J. [8, 9] provided some useful tools for evaluating the impacts of a launch failure. Our recent work [10] further studied the risk level analysis for hazard area. However there is little research regarding a dynamic risk assessment for aircraft rerouting during launches. This paper aims to compose a dynamic rerouting plan based on the dynamic risk assessment.

Dynamic aircraft rerouting requires us to accurately obtain and analyze the risk level within the hazard area. This paper provides a method to thoroughly inspect the hazard area during launch. Utilizing risk level maps, several risk level rerouting plans are modeled and evaluated. In this paper, the space launch operation evaluated is a hypothetical scenario utilizing a vehicle launched from Cape Canaveral Spaceport. The launch vehicle follows a two stage to orbit trajectory, in which the altitude of the mapped hazard area is set at the common commercial cruise altitude of 35,000 feet. Staging a realistic launch operation allows genuine debris uncertainty to be projected following the intended trajectory, which would provide us with a risk level map where risk level is a comprehensive index that multiple considers debris models and historical failure probabilities.

Methodologies used in this paper follow a set of steps to properly retrieve desired results. First, we develop an accurate method to dynamically compile a risk level map of the hazard area. Second, we propose the aircraft rerouting plan with the lowest risk, and lastly we validate the proposed method by comparing results with the current method by utilizing realistic launch scenarios. The main structure of this paper is arranged in the following order. The formulation of the risk level assessment is thoroughly described in section II. Section III presents the method used to simulate a proposed rerouting plan that is created based on the risk map. A GPU based framework is introduced in Section IV for fast evaluation. Section V displays a comparison between our proposed rerouting versus the method currently used. Section VI concludes results obtained in this paper.

II. Dynamic Hazard Area Analysis

The method proposed in this paper is intended to improve aircraft maneuverability and efficiency over hazard areas while maintaining low risk to aircraft. Depicting the impacted area that has the lowest risk requires the area in question to be divided into grids. This is accomplished by building a grid mesh in which each section is dynamically evaluated under a risk tolerance by using a Monte Carlo simulation. Risk level per section is decided using a comprehensive index containing two separate parts: Debris Model and Launch Failure Probabilities.

A. Debris Model

Developing a dynamic rerouting plan requires that the impacted area simultaneously updates over time. Obtaining this area heavily relies on creating an accurate method to project the debris model. Having an efficient process to analyze the debris model is necessary, in order to create a grid and calculate the probability of each individual grid section being hit. Properly calculating area hit by debris allows for more accurate evaluation of hazard area.

In order to effectively formulate the debris model, and provide results of the area impacted by debris, a simulation

was forged to mimic a space launch operation from Cape Canaveral following a two stage trajectory over the Atlantic Ocean. This location was chosen mainly due to being the United States primary launch site in recent years. Its proximity to the ocean with launch trajectory oriented away from populated areas makes Cape Canaveral the most ideal launch site when conducting launch operations. Due to its known reusable capabilities, SpaceX Falcon 9 space launches are more probable to occur in the future. Therefore, to simulate a launch operation that constantly impacts the air space, in this project we are using the vehicle specs from the SpaceX Falcon 9.

Several studies have presented method for calculating the uncertainty of debris trajectory in the case of potential rocket anomalies, accounting for weather conditions and analyzing the impacts endured by nearby aircraft [9]. Although numerous studies have provided adequate results when computing debris uncertainty, this paper further improves these methods with detailed models. In the case of an explosion, the model represents an assessment of some random given mass, number of pieces, initial velocity and explosion impulses. The piece in question must still abide by the laws of physics therefore the system must follow the conservation of mass principle.

$$m_{total} = \sum_{i \in N} m_i \tag{1}$$

Where total mass (m_{total}) is the weight the vehicle had at the time of incident and m_i is the mass for each debris piece. Meanwhile, the system should follow the momentum conservation as well.

$$m_{total} V_0 = \sum_{i \in N} m_i V_i \tag{2}$$

Where V_i is the initial velocity of debris i, calculated as the velocity vector obtained using the velocity of the rocket at the moment of incident plus the impulse velocity after explosion, V_0 is the initial velocity of the rocket at the moment of incident. During the debris propagation, three main forces are being addressed with these equations of motion, including lift, drag and weight per piece. Trajectory traveling over the Atlantic ocean indicates a spherical rotating Earth with a set of 3 degrees of freedom (3 DOF).

$$\dot{r} = V \sin\gamma \tag{3}$$

$$\dot{\theta} = \frac{V \cos\gamma \sin\psi}{r \cos\phi} \tag{4}$$

$$\dot{\phi} = \frac{V \cos\gamma \cos\psi}{r} \tag{5}$$

$$\dot{V} = \frac{L - D}{m} - g \sin\gamma + r\Omega^2 \cos\phi(\sin\gamma \cos\phi - \sin\phi \sin\psi \cos\gamma) \tag{6}$$

Where r is the radial distance from the center of the Earth to the vehicle, V is the Earth-relative velocity, θ and ϕ are the geodetic longitude and latitude, respectively, γ is the flight-path angle, ψ is the velocity heading (track) angle, g is the gravitational acceleration, and Ω is the Earth's self-rotation rate, m is the vehicle mass, L and D are the lift and drag, respectively.

$$L = \frac{1}{2} \rho V_R^2 S C_L(V) \tag{7}$$

$$D = \frac{1}{2} \rho V_R^2 S C_D(V) \tag{8}$$

Where ρ is the air density, S is the reference area, C_L, C_D are the lift and drag coefficients respectively and V_R is the velocity relative to the wind: $V_R = V - V_{wind}$. We assume the aerodynamics coefficients depend only on velocity (Mach number).

Due to potential debris having a vast range of size, shape and velocity, a large change in altitude throughout debris trajectory has a high chance of occurring. Having a large range in altitude indicates that some parameters will change, including air density, pressure, temperature and other parameters. The following equations are a set of barometric equations used to calculate these parameters at different altitudes.

$$P = P_b^{\frac{-g_0 M(h-h_b)}{RT_b}} \tag{9}$$

$$\rho = \rho_b^{\frac{-g_0 M(h-h_b)}{RT_b}} \tag{10}$$

$$T = T_b - L(h - h_b) \tag{11}$$

$$g = g_b \frac{(r_e)^2}{(r_e + (h - h_b))^2} \tag{12}$$

Where P is the pressure at the point of question, P_b is the static pressure, M is the molar mass of earth's air (0.0289644), $h - h_b$ is height from previous to current location, R is the universal gas constant (8.314459 J/(mol*K)), ρ is the air density, ρ_b is the density from previous point, T is the temperature at point of interest, T_b is the temperature from previous point, L is standard temperature lapse rate (K/m), g is the gravity at the current point, g_b is acceleration due to gravity from previous point, and r_e is the radius of the earth.

B. Launch Failure Probabilities

Launch failure probabilities are a major component considered while evaluating risk level for each section. Following National Playbook as an example, these probabilities are based on the records of different rocket accidents, in order to assess which stage of the trajectory an accident is most likely to occur. Categorizing historical data of failed missions aids us in identifying time and position space launch explosions contain a higher chance of happening. Obtaining the most probable point in design trajectory allows a much more accurate risk level map by targeting this point and calculating debris trajectories based on flight conditions. Historical data was analyzed and all known first and second stage to orbit accidents around the world were plotted with the x-axis being the year of the accident, and the y-axis being the time that the accident occur. The data collected from the last two decades of known first stage space launch vehicle failures are shown in Figure 1.

Figure 1 is a histogram of all known space launch vehicle failures in the last two decades around the world. Each point on this scatter plot is located based to the year accident and the time from launch that the incident occur. After some observation of the timeline, we can clearly notice the slight increase in the time of an incident during the trajectory, suggesting a slight linear increment in time incidents tend to occur during trajectory.

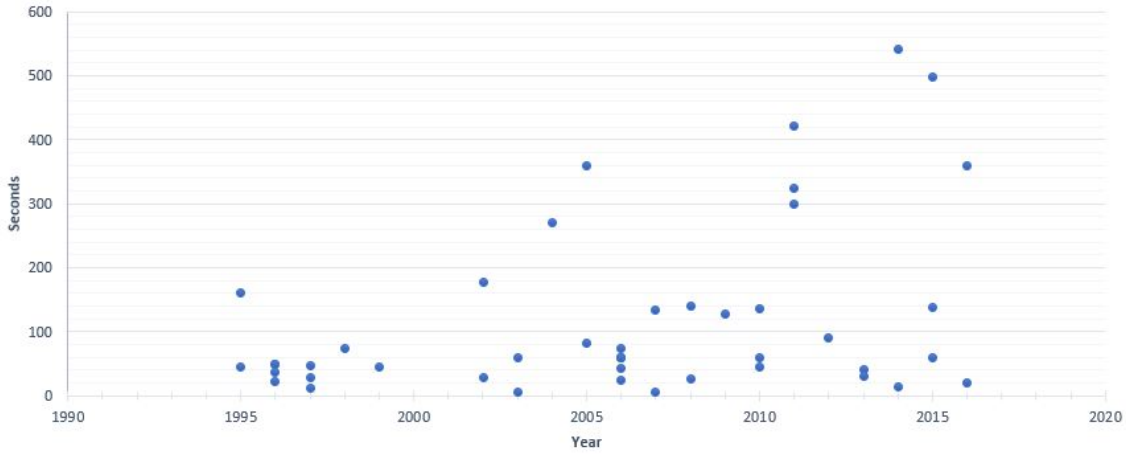


Figure 1. First Stage Launch Failure Histogram

Once research on historical events was carefully completed, the data found was separated into five different categories according to the year of accident. After the data was separated, Table 1 was developed. This table displays the number of incidents within the years in question as well as the average time of incident after launch.

Table 1. Historical Points of Incidents

Year	Number of Incidents	Average Time of Incident
1995 - 1999	18	53 seconds
2000 - 2004	9	108.4 seconds
2005 - 2009	13	92.1 seconds
2010 - 2014	11	182.3 seconds
2015 - 2019	5	215.2 seconds

As previously mentioned, significant improvements have been made in the space exploration field over the last few decades. Much of this improvement stems from technological advancements. Test launch procedure thoroughly done prior to any launch has also been heavily enhanced allowing for operations to be more efficient than in the past. Therefore, over these past 20 years, the number of incidents has decreased slightly as shown in Table 1. Aside from the number of incidents decreasing, one major parameter that has changed in the last two decades is the average time the incident occurred. By comparing the data from 1995-1999 to 2015-2019, launch failure occurs much later recently, around its first stage or early on its second stage, due to aerospace technology advancement.

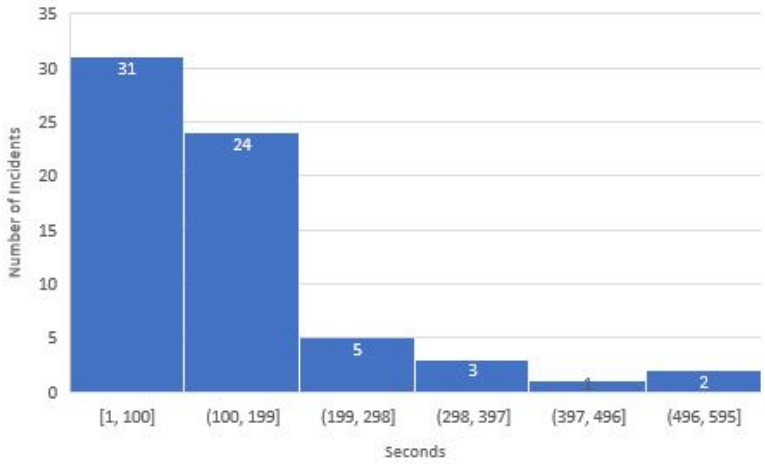


Figure 2. Probability Distribution

By separating data gathered into different times of incidents figure 2 is created. This figure is a probability distribution showing the failure probability at different times in vehicle trajectory with x axis being the time after launch and y axis being the number of incidents that occur during that specific time frame. Looking at the amount of incidents between 1 to 200 seconds, we can clearly see that most of the incidents happen earlier in the trajectory. From this analysis the time frame mainly being evaluated in this paper will be between 0 to 200, where the large majority of incidents occur.

C. Risk Assessment

Procedures are set to maintain the safety of vehicles in the air space. There are a variety of components to be considered while creating a risk assessment, in this paper we will focus on two key terms described before. After obtaining values of the two main factors we calculate the risk to the aircraft utilizing these two key terms mentioned to define a comprehensive risk index. The following equation represents the method used to calculating the total risk R_{Total}.

$$R_{Total} = 0.7R_{DM} + 0.3R_{LFP} \tag{13}$$

Where Debris model risk (R_{DM}) and Launch Failure Probability risk (R_{LFP}) contain different weight in which they affect the over all risk. These weights are determined based on parameters that need to be met, in this paper these values were decided by order of importance. Risk distribution used on this paper are set R_{DM} at 70 percent of total risk level while R_{LFP} is only 30 percent of overall risk. With set weight, overall risk is heavily governed by the debris model. Each section in the risk level map would have a risk level calculated, and depending on the level of risk tolerance, the area will be deemed opened or closed.

After creating the grid with each section dynamically assessed under a risk level, a risk level map is updated over time. Utilizing the dynamic risk level map, a section may open or close for a limited amount of time according to a tolerance level. It will help create alternative paths for nearby aircraft, which can significantly reduce flight time and distance. Although these areas open or close depending on the risk level, the most optimal path with lowest time and distance under risk tolerance has to be obtained. The method for assessing which path is most efficient is supported in section III.

III. Dynamic Aircraft Rerouting using Customized A*

Taking advantage of the risk level map created, three cases are proposed, where each case displays a different routing plan evaluating the efficiency and performance. Considered parameters when evaluating the efficiency of each routing case are the distance and time the vehicle takes to pass through the hazard area, and most importantly overall risk to the aircraft. The three methods assessed in this paper are historic method, original A* method and customized A* method.

A. Historic Method

The historic method used in this paper is assembled to resemble current methodology used by the FAA. Licensing procedures by the FAA are set to assure the safety of all air vehicles in the proximity in case of a rocket exploding during its trajectory. This method closes off large blocks of space following projectile trajectory, forcing air traffic to reroute around the blocks of space due to that being considered the safest course of action. Rerouting completely around the restricted area is a process that takes an extended amount of time, and increases the distance travelled. This hinders all air traffic in the area, and costs airline thousands of dollars per flight. The historic method is not the most efficient, especially when considering the additional operational costs.

As shown in Figure 3, static hazard area obtained using historic method is displayed in blue, this zone is considered a no fly zone. The area surrounding the rockets flight path is totally blocked off for a long time, which represents a worst case scenario where the flight path is blocked off once the rocket reaches the aircraft flight level. The aircraft navigates around the debris field and reaches its destination in a much longer time than in the dynamically updating cases. To further validate the method proposed in this research, we retrieve distance and time historical method requires to complete rerouting path and compare to results obtained from Original and customized A* Method.

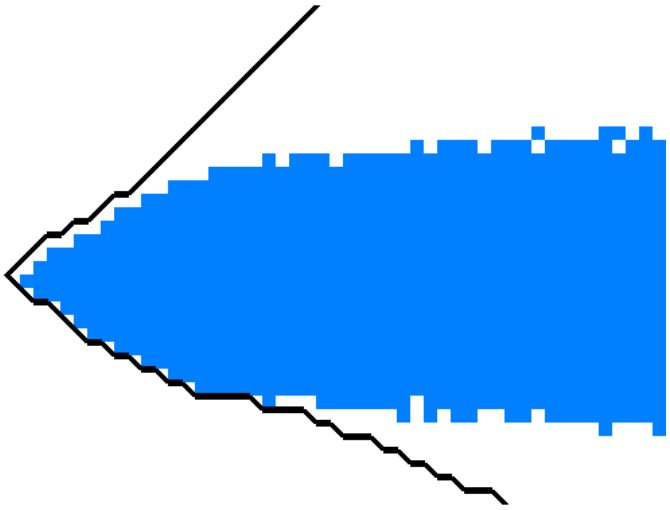


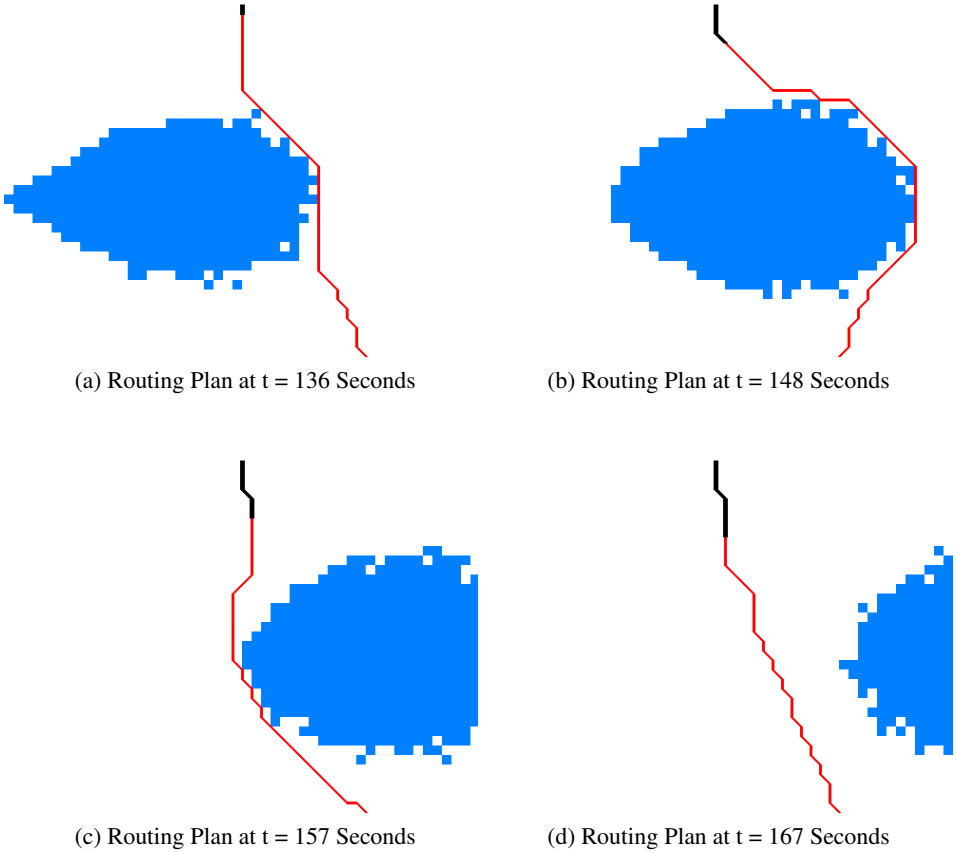
Figure 3. Worst Case Blocked Area

B. Original A* Method

Original A* method is a path search algorithm often used to go from one node to a final given goal with smallest cost. Based on risk assessment, the risk level map appears to generate a maze-like map. The maze is created after evaluating each section under set risk level tolerance. To navigate through the maze at the lowest expense, the algorithm used on this section is A* search algorithm, which intended to calculate the most optimal path through this maze-like map. The algorithm is driven by two different parameters, the first being cost of the path from start of node to the next node (g(n)) and the second being a heuristic function (h(n)) estimating the lowest cost path from one node to the final node. This algorithm is designed to render lowest cost, i.e. the least distance, traveled from the initial potion to the end location, making this the most appropriate method. F(n) being the total cost of section. Cost of the path is determined using the following equation evaluating the cost to move between any section within the grid section.

$$F(n) = g(n) + h(n) \tag{14}$$

The maze is dynamically created with each section evaluated every second following expected projectile trajectory. The A* search algorithm path is dynamically updated, providing expected routing path using current risk level map and aircraft location. This approach does provide a shorter path needed to pass through hazard areas with sections partially opened in comparison to the entire area being closed off.



(a) Routing Plan at t = 136 Seconds

(b) Routing Plan at t = 148 Seconds

(c) Routing Plan at t = 157 Seconds

(d) Routing Plan at t = 167 Seconds

Figure 4. Original A* Method

Figure 4 contains four screen shots taken from a simulation using original A* method. Images shown display debris impacted area in blue; the red line represents the most optimal path for vehicle to reroute around hazard area; and the black line is the recorded path taken by vehicle. Images are arranged to represent different times in trajectory, from 4a to 4d screen shots display the most optimal path calculated using original A* method dynamically updated according to current hazard area.

C. Customized A* Method

Although, the A* search algorithm provides the smallest cost to move from the starting position to the end goal. Due to the complexity of the problem at hand, this problem can be treated as two moving objects, in which hazard area containing a set trajectory while the aircraft dynamically updates its location according the location of hazard area. The problem becomes an object avoidance problem. With hazard area dynamically updating, there are instances in which the aircraft is moving in current shortest estimated path but due to the high speed of the rocket, the aircraft may already be in the location that will soon be considered a no-fly zone. The algorithm needs a look ahead method to evaluate if a section surrounding the current node is safe to travel to. Unfortunately in some cases the hazard area may shift too fast, and the aircraft may not have enough time to avoid a direct conflict. Therefore, A* search algorithm must be customized to support the problem.

A straightforward way is using previous cost function discussed in part B. Manipulating the path to avoid situations where the path travels directly in front of an expected hazard area path, the heuristic function is modified to provide a much lower risk path than the previous algorithm. Recall that on equation 14 provided in part B, heuristic equation is a value need to depict lowest cost path. Heuristic is obtained using the following equation.

$$h(n) = w\sqrt{(n_{1x} - n_{2x})^2 + (n_{1y} - n_{2y})^2} \tag{15}$$

Where w is the cost weight of the section, n_{1x} and n_{1y} are x and y values of the current node, similarly n_{2x} and n_{2y} are x and y values of the final node that aircraft is attempting to travel to. Algorithm's decision is generally completed after calculating lowest cost node. Normally, each section in the risk level map is assigned a weight depicting the cost of the aircraft moving from one node to the next. By adjusting the weight of the sections that will be closed off in the future, we are able to increase the cost of those sections, making it less likely for the algorithm to choose the higher cost nodes. In order to manipulate the decision making of this algorithm to derive away from hazard area, the expected weight equation below was developed.

$$w = \frac{\lambda}{ix - max(ox)} \tag{16}$$

Where λ is considered the max risk value assigned to the section, this value can be altered to meet parameters needed. ix is the term described as a node's location on the moving direction of hazard area, and $max(ox)$ is the right most boundary of current hazard area. Therefore, the weight is assigned based on the look-ahead distance. By continuously evaluating each section under a max risk value (λ) and the distance from the right most hazard blocked area, this method dynamically changes the cost of each section following the trajectory of the impacted area.

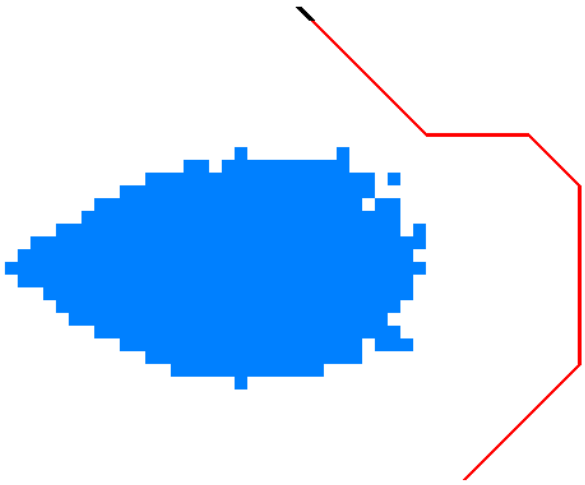


Figure 5. Customized A* Method

Figure 5 displays a simulation completed using the customized A* method. In contrary to the original method, the customized method has a predicted optimal path maintaining a distance from the hazard area. This is due to the weight cost making the sections directly in front of the hazard area slightly more expensive. A large increase in weight

cost can significantly increase the predicted path around hazard area. Therefore, a strong consideration of λ value is required. Due to this term containing the largest impact on the overall cost weight, careful analysis on λ values is recommended. A more detailed evaluation of this parameter is shown in Figure 12.

IV. GPU Based Framework for Efficient Evaluation

A. The GPU Framework Based on CUDA

The estimated debris area was compiled using GPU framework. This resulted in a faster compilation time when compared to the compilation time of traditional CPU framework. The GPU framework utilized Nvidia's CUDA platform to compute the estimated debris area in parallel. The program implemented CUDA as shown in the flowchart, figure 6. The GPU and CPU work in a similar manner for parallel computing but the GPU framework is able to compute magnitudes faster than the CPU framework.

Both the GPU and CPU took a similar approach to find the debris footprint, but the amount of computation time was vastly different. The computation time required to produce the debris footprint showed linear functions that were dependent on the number of pieces of debris. This relationship is shown in Figure 7. The GPU framework is able to compute 1,000 pieces of debris in about 0.00175 seconds, while the CPU framework is able to compute 1,000 pieces of debris in about 3.2 seconds.

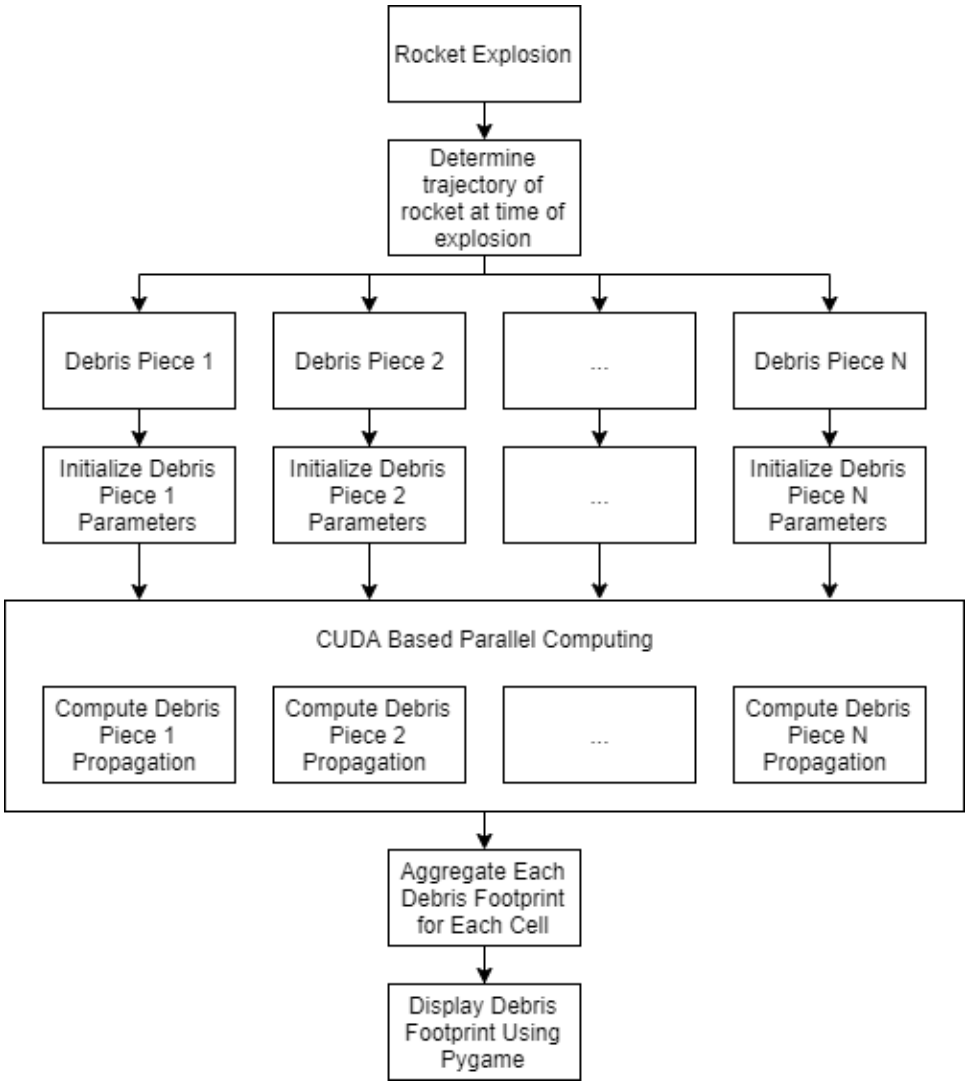


Figure 6. Flowchart for GPU Based Parallel Computing

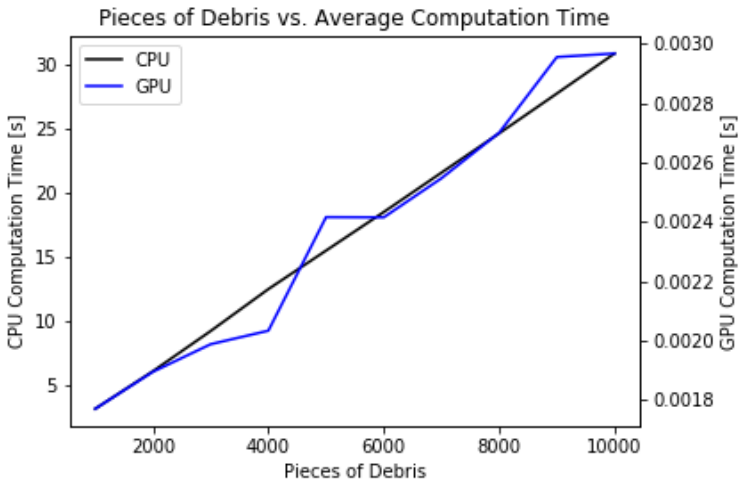


Figure 7. Pieces of Debris vs. Average Computation Time

The system used to run the simulation has an Intel Core i7-9700K CPU with 8 cores and 8 logical processors, and a NVIDIA Quadro P620 GPU. The GPU simulation calculated the trajectories for all the debris significantly quicker than the CPU simulation. The relationship between the speed of the GPU simulation and the CPU simulation is shown in Figure 8. The speed of the GPU over the CPU, hereafter referred to as GPU speed up, is a linear function that shows that as the number of debris pieces increase, the GPU speed up increases linearly. Due to the speed of the CPU framework, the GPU framework is the ideal parallel computing method to use while computing the footprint of a large scale debris calculation.

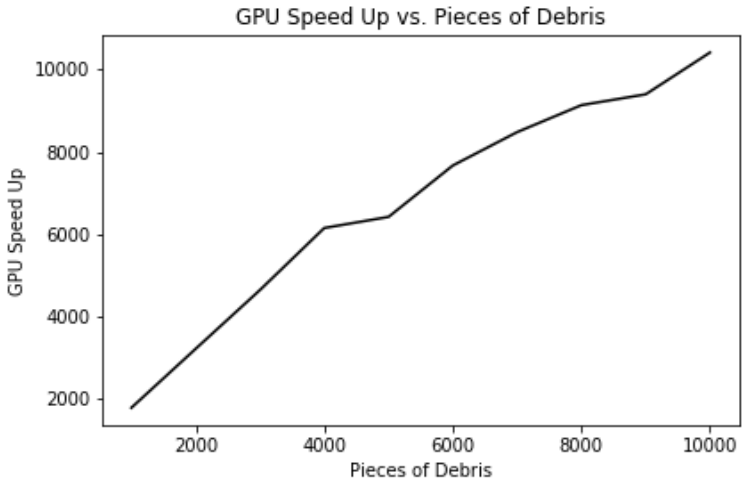


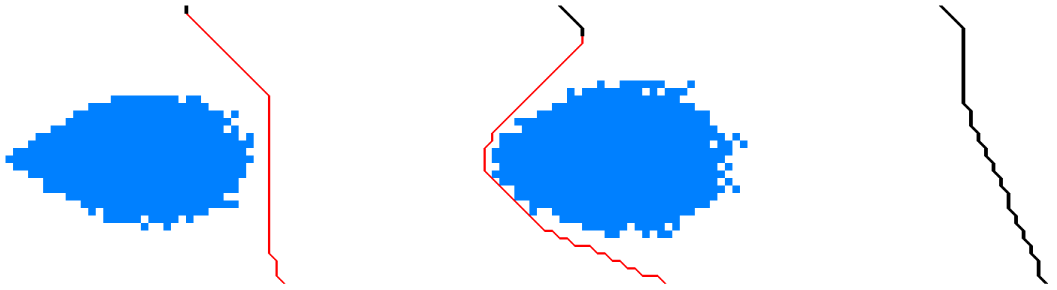
Figure 8. GPU Speed Up vs. Pieces of Debris

B. Integrated Simulator Based Pygame

A simulation was created in Pygame [11] to simulate the blocked area from the debris as the rocket launched into orbit. The goal for the simulation was to show the A* algorithm instantaneously update the aircraft trajectory as the rocket crossed its flight path. The aircraft had a trajectory that directly crossed the trajectory of the rocket. The simulation was broken up into a 50 km by 50 km blocked area with each block representing 1 km^2. The coordinates for the blocked area are relative to the launch location. In this case, the rocket has a trajectory in the positive horizontal direction without any vertical movement. The boundaries of the simulation are at 25 km and 75 km in the horizontal direction, and at -25 km and 25 km in the vertical direction. There was a tolerance added to the simulation of at least certain pieces (based on set tolerance level) for each block due to the unlikelihood of small pieces of debris hitting the

aircraft in the 1 km^2 area. In comparison to the aircraft, the rocket travels very quickly through the frame while the aircraft only moves a few blocks.

Each frame of the Pygame simulation represents one second of flight time. Figure 9 and figure 3 show snapshots of the simulation in Pygame. The blue blocks represent the blocked area, the black line represents the aircraft trajectory, and the red line represents the A* algorithms projected trajectory for the aircraft. The t = 137 second snapshot represents the time when the rocket reaches the flight level of the aircraft. The t = 149 second snapshot represents the time when the debris footprint is directly between the aircraft and the aircraft destination. The t = 281 second snapshot shows the flight path of the aircraft as it reaches the destination.



(a) Simulation at t = 137 seconds (b) Simulation at t = 149 seconds (c) Simulation at t = 281 seconds

Figure 9. Pygame Simulation

V. Simulation Results

A. Debris Data set

Data from the FAA was utilized to create a more accurate model of the debris map. The FAA data is from the Space Shuttle Columbia disaster on February 1st, 2003. The data set is a catalog of the debris which is divided into 11 groups based on the average weight of each piece. Each group has the number of pieces, total weight, reference area, ballistic coefficient, velocity increment, coefficient of drag, and coefficient of lift. This data set allowed the estimated debris area to reflect a real-world example of a rocket explosion in mid flight. The trajectory of each piece was calculated using the data from the FAA data set.

B. Results

Using the Columbia dataset, a simulation was formulated to follow a two stage to orbit trajectory with mass equal to roughly 500,000 kilograms. To facilitate the simulation using launch failure probabilities data, , the time in the trajectory with the highest chance of failure were analyzed in this research.

Debris model simulations are run for every second after launch, where the current status of the rocket was retrieved including velocity, mass, and location as well as other parameters such as density, pressure, temperature, etc. These parameters serve as the initial conditions for our debris model simulation, which calculates the potential location each debris would hit a normal aircraft cruise altitude of 35,000 feet. By creating a grid mesh, we can calculate the amount of pieces that each section, which allows us to decipher the risk level due to debris model. Using this method for each second on the expected trajectory, a dynamic risk level map is create. The map shifts with the projectile's trajectory, closing off any area with a risk level higher than a set tolerance. All three different cases in this paper follow the same risk level tolerance. Assuming commercial aircraft standard cruise speed is 250 m/s at an altitude of 35,000 feet, all three cases are tested for several simulations retrieving the distance and time for aircraft to pass through the hazard area. As an obstacle avoidance problem, there are some cases where the hazard area moves directly onto the location of the aircraft, which results in a infeasible solution for the A* search algorithm.

Initially, we set the starting location for the aircraft away from the projected hazard area with final destination clearly across the expected blocked space. Simulations for each case were run with the same starting and ending location. Table 2 shows the average performance for each case. The results displayed are an average of 20 path simulations for each case. Results obtained exhibit a large difference in distance and time of rerouting plan between the historic and the two A* search algorithms. Although the historic method has the lowest risk to aircraft, distance

and time required to pass through the hazard area for the historic method are nearly twice the amount of the A* algorithm. After completing 20 simulations for each case, there were zero collisions for all the simulations. Both the A* algorithm and the customized version displayed equal results, substantially reducing the required travel distance and time. Because the aircraft's starting location was away from the expected impacted area, both algorithms delivered the same results.

Table 2. Simulation Results: With Starting Position Far From Hazard Area

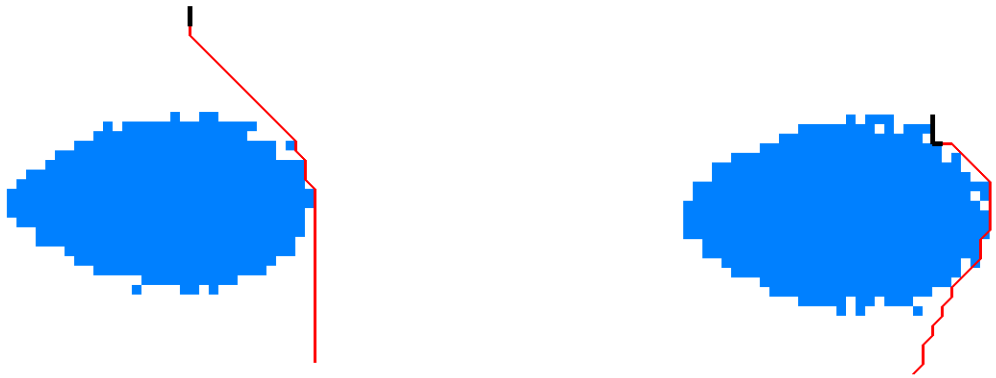
Case	Average Distance (km)	Average Travel Time (s)	Number of Collisions
Historic	1050	262.5	0/20
A* Algorithm	608	152	0/20
Customized A*	608	152	0/20

However, by adjusting the initial position closer to the danger zone we expected a difference in results. Shown in Table 3 are the average distance and time for each of the cases. The original algorithm does have shorter distance and time required, however there were instances where collision happened. With a 30 percent of collisions, original A* search algorithm is deemed not the safest rerouting method. After some observation we can clearly see that customizing the A* algorithm is the best method.

Table 3. Simulation Results: With Starting Position Near Hazard Area

Case	Average Distance (km)	Average Travel Time (s)	Number of Collisions
Historic	1200	300	0/20
A* Algorithm	560	140	6/20
Customized A*	576	144	0/20

Figure 10 displays the original A* algorithm having two different conditions. Condition one has a starting position far from the impacted area, allowing each method to update in a timely order, which provides us with adequate results. Meanwhile in figure 10b, the starting position is designed to start on a location, which is likely to be closed off in the near future, in order to analyze the behavior of original A* algorithm. After several simulations processed, a pattern constantly shown that the aircraft does not have enough time to maneuver away because the blocked area shifts from left to right at a faster rate than the aircraft is capable of avoiding.

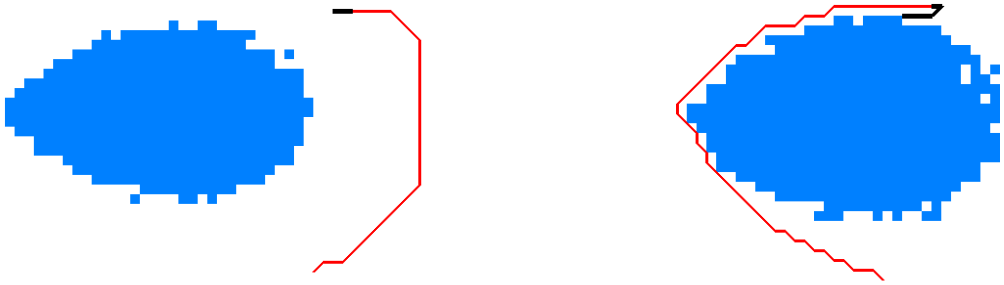


(a) Far from Impacted Area

(b) Near Impacted Area

Figure 10. Original A* Method

To overcome this issue with the original A* algorithm, a modified version is required. Figure 11 demonstrates the behavior of the modified method. The weighted cost in the heuristic equation for each section is dynamically updated according to the future blocked area. Specifically, the weighted cost in front of the hazard area is increased to encourage the aircraft to reroute to the back of the hazard area. This results in a safer trajectory for the aircraft. Depending on the weighted cost for traveling in front of the hazard area, the aircraft is allowed to pass in front of the hazard area in some cases, as shown in figure 11a. Unlike the original method, as hazard area approaches the aircraft's airspace, the vehicle begins to avoid the hazard area by drastically adjusting its course as shown in figure 11b.

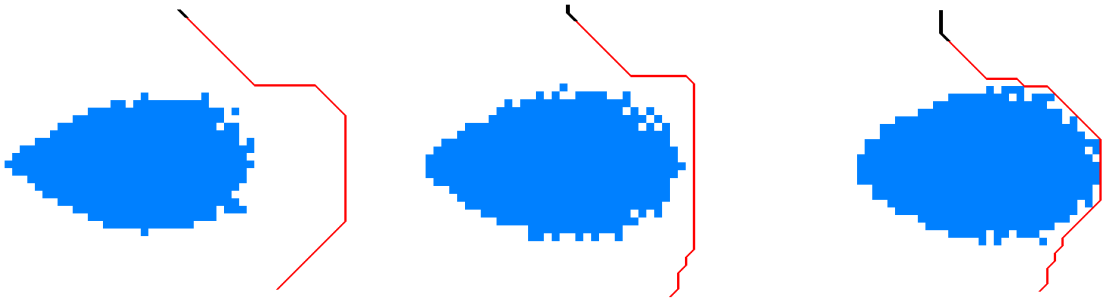


(a) Near Impacted Area

(b) Near Impacted Area

Figure 11. Customized A* Method

Customizing the A* algorithm requires the adjustment of the weight λ. By giving dangerous sections a significant larger weight cost, providing the safer route, the customized A* search algorithm provides a rerouting plan slightly longer than the original A* algorithm. Therefore, a careful evaluation of how to adjust the weight cost is examined. Figure 12 demonstrates the three potential changes these weight adjustment could produce. From the vast number of possible adjustments, we tested three different cases of λ, showing high, medium, and low weight cost adjustment. Considering weight cost equation (16) provided on section III part C, simulations were completed to search for the most proper λ term.



(a) $\lambda = 35$

(b) $\lambda = 20$

(c) $\lambda = 5$

Figure 12. Customized A* Method: High, Mid and Low Weight Adjustments

Table 4 contains the average results for the three different λ values evaluated. Comparing the distance, time, and number of collisions awards us with an idea of what level of risk we should be using in the routing simulations. As expected, values retrieved from the customized method with a large cost weight will inevitably have a longer expected path while an aggressive method, low value of λ, would render values approaching the original A* search algorithm. Therefore, a proper choice of λ in the customized method provides us with the most efficient solution under safety requirements to navigate through the hazard area.

Table 4. Simulation Results: For Different λ

λ	Average Distance (km)	Average Travel Time (s)	Number of Collisions
35	592	148	0/20
20	572	144	1/20
5	564	141	12/20

VI. Conclusion

In this paper, simulations on the safe integration of space vehicles into the national airspace were performed. Hazard areas are computed using a randomized model, displaying an area a rocket explosion may have on the NAS. A detailed

overview of traffic flow management is discussed, providing the overall process of aircraft rerouting as well as the challenges these methods will encounter in the near future. The method proposed in this paper contains a risk level map created as a grid mesh with each section dynamically evaluated for a risk level. By using a risk level tolerance, these section may be opened or closed depending on their risk. This means that some sections may be partially opened for a limited amount of time allowing aircraft to pass through. Three methods are discussed in this paper to generate the optimal route to navigate aircraft through a dynamic hazard area. The methods used are the historic method, original A* method, and customized A* method. Even though the historical method provides us with the lowest risk to aircraft, increasing operational costs make this method not the most efficient method. While the original A* search algorithm does provide us with adequate results, the risk it contains does not out weigh the distance and time improvements. Therefore, using the customized A* algorithm, which provided us adequate and low risk results, is more desirable. Utilizing this method results in fewer flights having to be rerouted with an average of 40 percent less distance needed to be covered and minimizes the overall cost exponentially. The Customized A* method indeed gave us much more desirable results but would be heavily dependent on the weight cost adjustment. Additional work must be done to further improve and obtain the most cost-efficient method to navigate through the hazard area.

Acknowledgments

The authors are grateful to Daniel Murray and the FAA Office of Commercial Space Transportation for the helpful discussion on problem formulation and the data set support.

References

[1] Murray, D., "The FAA's Current Approach to Integrating Commercial Space Operations into the National Airspace System," *Federal Aviation Administration*, 2013.

[2] FAA, "FAA Aerospace Fiscal Years 2019-2039," *Federal Aviation Administration*, 2019.

[3] AirlinesforAmerica, "U.S. Passenger Carrier Delay Costs," *Link: http://airlines.org/dataset/per-minute-cost-of-delays-to-u-s-airlines/*, 2019.

[4] Chen, J., Chen, L., and Sun, D., "Air traffic flow management under uncertainty using chance-constrained optimization," *Transportation Research Part B: Methodological*, Vol. 102, 2017, pp. 124–141.

[5] Chen, J. and Sun, D., "Stochastic ground-delay-program planning in a metroplex," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 1, 2017, pp. 231–239.

[6] Tompa, R. E., Kochenderfer, M. J., Cole, R., and Kuchar, J. K., "Optimal aircraft rerouting during commercial space launches," *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, IEEE, 2015, pp. 9B1–1.

[7] Tompa, R. E. and Kochenderfer, M. J., "Optimal aircraft rerouting during space launches using adaptive spatial discretization," *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, 2018, pp. 1–7.

[8] Colvin, T. J. and Alonso, J. J., "Compact envelopes and SU-FARM for integrated air-and-space traffic management," *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 1822.

[9] Capristan, F. M. and Alonso, J. J., "Range Safety Assessment Tool (RSAT): An analysis environment for safety assessment of launch and reentry vehicles," *52nd Aerospace Sciences Meeting*, 2014, p. 0304.

[10] Bojorquez, O. and Chen, J., "Risk Level Analysis for Hazard Area During Commercial Space Launch," *AIAA/IEEE 38th Digital Avionics Systems Conference (DASC)*, 2019.

[11] Shinners, P. et al., "Pygame," *link: http://pygame. org/[Online]*, 2011.