

# CONVECTIVE WEATHER AVOIDANCE WITH UNCERTAIN WEATHER FORECASTS

*Robert Windhorst, NASA Ames Research Center, Moffett Field, California*

*Mohamad Refai and Sinan Karahan, University of California Santa Cruz, Moffett Field, California*

## Abstract

This paper describes simulations of an automated planning system that routes flights around airspace impacted by forecasted convective weather. If the system predicts that a flight will enter a weather-impacted airspace within a predefined time horizon, it generates a new route. Because the forecasts are uncertain, the system periodically generates, using updates of the weather forecasts and radar tracks, new reroutes. The simulations included convective weather in the northeastern quadrant of the United States over a 24-hr period. Multiple simulations investigated the system performance as the planning horizon and planning frequency varied. As the planning horizon and frequency increased, the system successfully routed more traffic around weather but with more route changes. For a planning horizon of 20 to 120 minutes and a planning frequency of four cycles per hour, the reroutes increased flight time by 3.3% and avoided 79% of the weather-impacted airspaces that were detected. Most flights required one to three reroutes to pass by a weather-impacted airspace, while the worst case flights required six reroutes.

## Introduction

For safety and passenger comfort, flights avoid airspace containing severe weather. One way of doing this is by rerouting flights around the airspace. For encounters predicted to occur in more than an hour, reroutes are currently selected from predefined sets, called playbooks and coded departure routes. Because the reroutes in these sets were pre-designed without knowledge of the specific weather events, they are usually not the most efficient. In practice, parts of playbook routes can be combined to refine a route, yet the parts are still pre-defined. An automated planning system could generate, extempore, reroutes that are tailored to the size and shape of the weather-impacted airspace and to the surrounding air traffic. In addition, the automated system could be integrated with other future

automated systems designed to solve traffic congestion problems in all parts of the airspace [1 and 2].

Several researchers have developed algorithms that generate reroutes around airspace with severe weather. In [2-5], grid-based search methods were used to find routes around severe weather. Reference [2] developed a sequential rerouting system and measured its performance as look-ahead-distance was varied. As the look-ahead-distance increased, rerouting delays decreased and the number of reroutes produced increased. Sridhar et. al. [6] proposed a geometric algorithm for generating a path stretch maneuver, consisting of auxiliary waypoints. It was used to route flights around congested sectors. The maneuver avoided the airspace by deviating the flight around the shorter side of the airspace. In [7], Love et. al. modified an automated conflict detection and resolution system to reroute flights around weather. The system not only prevents weather encounters, but also prevents conflicts with other aircraft. Although weather forecasts were used in these references, none analyzed the missed encounters due to the forecast uncertainty.

In this paper, an automated planning system that routes airborne flights around airspace impacted by forecasted weather is developed. Since the weather forecasts are uncertain, i.e. they are not perfectly accurate, the system periodically re-plans based on the most recent weather forecast, flight tracks, and flight plan. Flights predicted to encounter hazardous airspaces within a predefined planning time horizon are included in the plan. Encounters predicted to occur before the beginning of the planning horizon are accounted for by another system, such as the one developed in [7]. This is because that system builds reroutes that do not violate separation distances with other aircraft. Embedded within the planning system developed here is the algorithm from [6]. It is used to build reroutes, sequentially, for each predicted encounter. When a flight has more than one predicted encounter, reroutes are generated in order of nearest

time-to-encounter. The performance of the system is evaluated as the length of the planning horizon is varied between 20 and 100 minutes and the planning frequency is varied between 2 and 12 cycles per hour. Performance is measured by number of missed encounters, average delay per reroute, and number of reroutes per flight.

This paper is organized as follows. First the weather forecast used in the experiment is presented. Next, the automated planning system is described. Results of a series of simulations designed to study the effect of the planning horizon and frequency on performance are illustrated. The paper concludes with a summary, conclusion, and future work.

## Convective Weather Forecasts

Weather, traffic congestion, military operations, and spacecraft launches are examples of events occurring within an airspace that may require flights to avoid it. The nature of an event determines the size, shape, and movement of the undesirable airspace. Although this paper is focused on airspace impacted by convective weather, the planning system can build reroutes around any hazardous airspace.

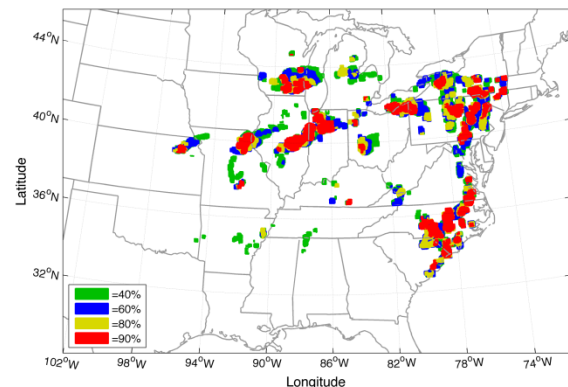
There are various meteorological forecasts that predict the location, size, and severity of convective weather. These forecasts consist of multiple datasets describing various meteorological parameters, such as vertically integrated liquid and echo tops. The parameters need to be combined and interpreted to identify undesirable airspace. References [8-10] discuss a model for translating these forecasts into a set of dynamic airspaces that flights are likely to avoid. The model is named the Convective Weather Avoidance Model (CWAM).

For the purposes of this paper, a CWAM forecast consists of a set of polygons that define the contours of airspaces containing convective weather. The set is divided into four subsets, each having a specified probability that flights will avoid the airspaces defined by the polygons. The probabilities are 40%, 60%, 80%, and 90%. In addition, the polygons are defined for specific altitudes, beginning at 25,000 ft. and ending at 45,000 ft. with 1,000 ft. increments.

A set of CWAM forecasts is periodically produced every five minutes. A set includes 24 forecasts plus one nowcast, which represents the

weather at the time that the set was produced. The forecasts range from five minutes to two hours with five-minute intervals.

Figure 1 depicts a CWAM forecast covering an area slightly larger than the northeastern quadrant of the United States. The one-hour forecast was generated at 3:00 PM Eastern Daylight Time (EDT) on 27 July 2006 and was valid at 4:00 PM. In the figure, the polygons are filled in to make them more visible. Green areas are 40% polygons, and blue areas are 60% polygons. Yellow areas are 80% polygons, and red areas are 90% polygons. Higher probability areas are drawn on top of lower probability areas. For areas representing the same convective weather cell, lower probability ones enclose higher probability ones.



**Figure 1. One-Hour CWAM Forecast for 4:00 PM EDT on 27 July 2006**

The shape and size of the polygons for all forecasts on 27 July 2006 were analyzed. Between 4 and 600 vertices defined each polygon, and in general, polygons were non-convex. Their cross sectional lengths ranged from 4 km. to 200 km., and their aspect ratios ranged from 1/1 to 1/10.

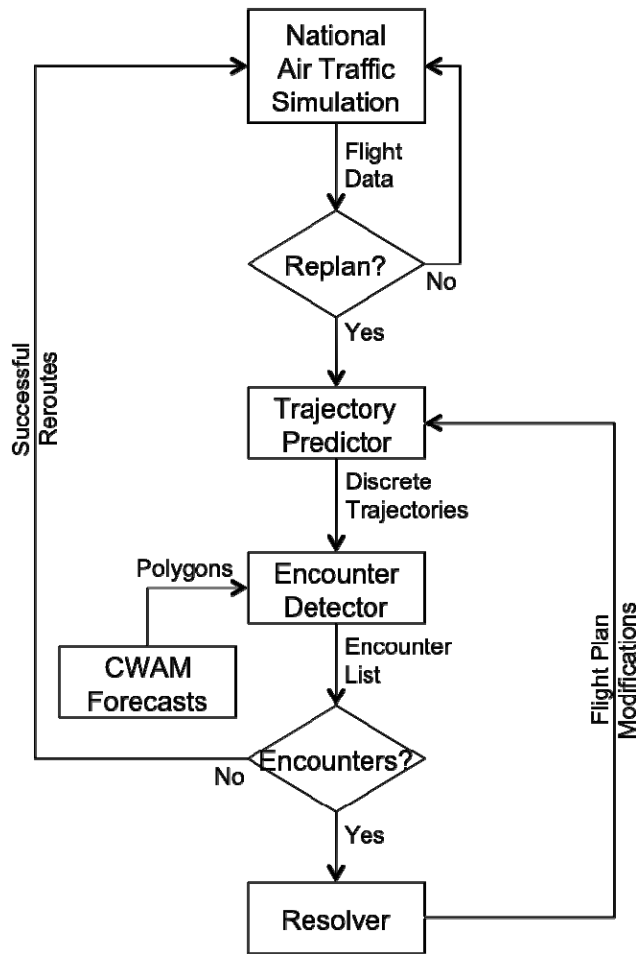
## Automated Planning System

The automated planning system periodically generates reroutes for all flights predicted to encounter within a specified planning time horizon CWAM forecasted polygons. It consists of multiple functions, which are described next.

### System Architecture

Figure 2 shows a block diagram of the automated planning system. The system consists of three main blocks: the trajectory predictor, the

encounter detector, and the resolver. Other blocks represent the national air traffic simulation and the CWAM forecasts. The two diamonds illustrate decision points.



**Figure 2. System Architecture**

Planning occurs periodically based on a specified frequency, which was nominally 4 cycles per hour. In the experiment described in the results section, the frequency was varied. At the beginning of a planning cycle, current flight data, consisting of tracks and plans, are collected from the air traffic simulation, see the decision block in the upper half of Figure 2.

Flight tracks and plans are used in the trajectory predictor to build two-hour trajectories for each flight. The trajectories are discrete with one-minute resolutions. They are created using a four degree-of-freedom flight dynamics model. Aircraft lift and drag coefficients, fuel burn rates, weights, and climb and

descent speeds were selected from the Base of Aircraft Data [11].

The encounter detector receives CWAM forecasts and the discrete trajectories. Each discrete point in a trajectory has a flight time. The polygons in the forecast for that time are checked to determine if the point is located inside one of them. An instance where a point lies in a polygon is called an encounter. Since the time resolution of the forecasts is five minutes and that of the trajectories is one minute, the forecasts are held valid until the next forecast.

To minimize computations, the detector does not check parts of the trajectory between points for intersections with polygons. Since the trajectory points are one-minute apart, which translates to 7 nm for a flight traveling 420 knots, encounters where the flight does not penetrate the polygon by more than 7 nm could be missed. Although not done here, these encounters could be detected by adding buffer zones to the polygons, interpolating, or increasing the trajectory resolution.

A list of encounters is passed from the detector to the resolver, which builds a reroute for each encounter. Since the resolver uses straight-line approximations for the trajectory, the reroutes may not successfully avoid all of the polygons. To check for failed reroutes, the resolver sends all of the reroutes, in the form of flight plan modifications, back to the trajectory predictor. New trajectories are predicted for flights with plan modifications, and the process is repeated. When no more encounters are detected, the process ends, and the flight plan modifications are sent to the flights in the air traffic simulation. The process will also end if it has iterated more than three times, even if there are unresolved encounters remaining. In the next planning cycle, both the weather forecasts and the flight trajectories will evolve. This evolution may allow a feasible path to be computed.

Encounters that occur before the start of the planning horizon are not resolved. This is because reroutes for those encounters must also prevent the flight from violating separation criteria with other aircraft in the area. This problem is out of the scope of this research.

The next two subsections detail the algorithm that creates the reroutes.

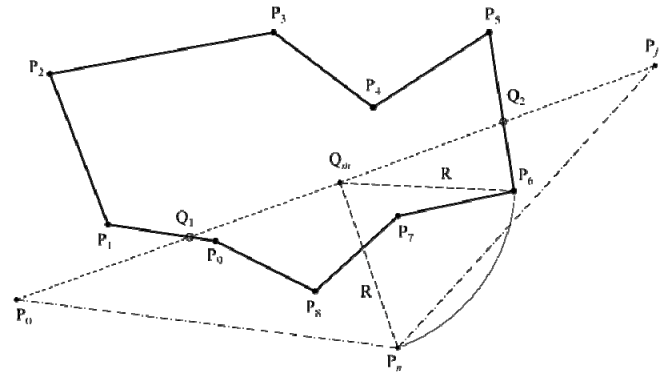
## Resolver

The resolver uses the algorithm from [6] to build reroutes around polygons. This algorithm was selected because it is more computationally efficient than those in [2-5]. Efficiency was important because the algorithm was used to plan reroutes for 53,000 flights operating in a 24-hour time period.

The algorithm identifies the side with the shortest path around the polygon. Then, auxiliary waypoints on that side are inserted into the route. The steps of the algorithm are given in [6]. They are repeated here for completeness.

Let  $P_1$  through  $P_9$ , in Figure 3, be the vertices of the polygon and  $P_0$  and  $P_f$  be the starting and ending points, respectively, of the reroute. Here,  $P_0$  and  $P_f$  were placed 10 nm before and after the polygon.

1. Find the intersections of  $\overline{P_0P_f}$  with the polygon and label them  $Q_1$  and  $Q_2$ .
2. Find the midpoint of  $\overline{Q_1Q_2}$  and label it  $Q_m$ .
3. Divide the vertices by their location relative to  $\overline{Q_1Q_2}$ . In Figure 3, the right set is  $\{P_6, P_7, P_8, P_9\}$ , and the left set is  $\{P_1, P_2, P_3, P_4, P_5\}$ .
4. For each set, find the vertex ( $P_l$  or  $P_r$ ) furthest from  $Q_m$ . In Figure 3,  $P_l$  is  $P_2$  and,  $P_r$  is  $P_6$ .
5. Select the vertex,  $P_l$  or  $P_r$ , closest to  $Q_m$  and label the distance from  $Q_m$  to it  $R$ . In Figure 3,  $P_6$  is closer to  $Q_m$ .
6. Construct a line of length  $R$  perpendicular to  $\overline{Q_1Q_2}$ . It starts at  $Q_m$  and points towards the side of the vertex chosen in step 5. In Figure 3, this line points right.
7. The end of the line drawn in step 6, point  $P_n$ , is the new auxiliary waypoint in the flight plan.



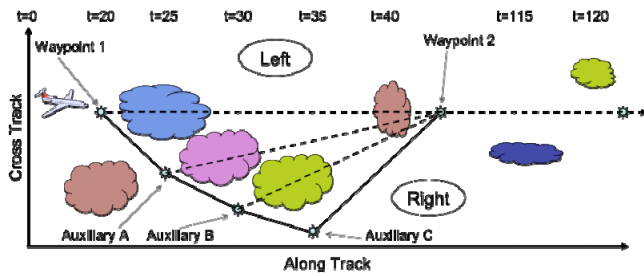
**Figure 3. Construction of a Reroute Around a Polygon**

If any part of the new route intersects the polygon, define either  $\overline{P_0P_n}$  or  $\overline{P_nP_f}$ , whichever intersects the polygon, as the new  $\overline{P_0P_f}$  and repeat, starting with step 1. On subsequent iterations in step 7, a new  $P_n$  is generated. In reference [6], the old  $P_n$  from the past iteration is discarded. Here, the old  $P_n$  is inserted into the flight plan and kept. The cycle continues until the new route does not intersect the polygon, or it has iterated fifty times.

## Flights with More than One Encounter

The algorithm described in the previous section only considers one encounter at a time. Some flights have more than one encounter within the planning horizon. The encounters for these flights are ordered by encounter time. The encounter with the earliest time is selected and solved, adding an auxiliary waypoint to the route.

Auxiliary waypoints change the downstream path of the route and affect downstream encounters. For example, they may be eliminated, as the one occurring at 40 minutes in Figure 4. Or, new encounters that were not there during the first detection may appear, see the encounter at 35 minutes in Figure 4. Each time an auxiliary waypoint is added to the route, the downstream route must be rechecked for encounters.



**Figure 4. Application of Resolver**

Figure 4 illustrates the process for building reroutes for downstream encounters. The encounter predicted to occur at 25 minutes is considered first. Auxiliary waypoint A, which avoids the polygon at 25 minutes, is inserted into the route. The trial flight segment that connects auxiliary waypoint A to waypoint 2 is shown as a dashed line segment. This new flight segment is checked for possible intersections with other polygons on its path. It is found to intersect a polygon at 30 minutes. Auxiliary waypoint B is computed to avoid this polygon, and a new trial flight segment is computed, shown as the dashed line from auxiliary waypoint B to waypoint 2. The procedure continues iteratively until all flight tracks within the planning horizon are clear of any polygons.

When there are downstream polygons that need to be avoided, the final reroute does not, in general, go around the shortest side of the group of polygons. In Figure 4, it is clear that the final route would have been shorter had it gone around the left of the first polygon instead of the right.

## Results

The automated planning system was implemented in a simulation of the air transportation system called the Airspace Concepts Evaluation System (ACES) [12 and 13]. Simulations were conducted to assess the performance as the planning frequency and planning horizon were changed. In addition, reroutes for a single flight are tracked to illustrate how the planning system operates.

### Simulation Setup

Air traffic simulations were configured using CWAM forecasts generated from weather on Thursday, July 27, 2006. Convective weather that day was severe. It impacted the Midwest, especially affecting the Chicago area. Although the western

United States was also impacted by severe convective weather, only weather in the northeastern quadrant of the United States was included in the simulations. This is because at the time the simulations were performed, CWAM forecasts were only available for that region. The CWAM probability of flight deviation was 60%, which was selected to be consistent with [2, 3, and 7].

Flight plans were not selected from the same day that the weather was recorded because the traffic that day was routed around the weather. Instead, flight plans were taken from a clear weather day so that the routes would intersect the weather and test the planning system. They were extracted, using the Enhanced Traffic Management System, from the national airspace system on Tuesday, August 26, 2006. The traffic schedule contained 53,000 commercial flights that operated during a 30-hour time period. This represents a heavy traffic day.

Winds were not modeled in the simulations. Airport arrival and departure rates and airspace capacities were unconstrained. Reroutes were only computed for flights in cruise.

Multiple simulations varying three parameters of the system were executed. Table 1 shows the experiment matrix. The planning horizon was reduced in two different ways. In case 1, the far horizon boundary was moved backward, and in case 2, the near horizon boundary was moved forward. Figure 5 illustrates the relationship of the near and far boundaries with the full prediction horizon, which includes the near, planning, and far horizons. In case 3, the planning frequency was varied. The frequency is the inverse of planning interval, or time between successive plans.

**Table 1. Experiment Matrix**

Case	Near Horizon (min)	Far Horizon (min)	Interval (min)
1	20	40, 60, 80, 100, 120	15
2	10, 20, 40, 60	120	15
3	20	120	5, 10, 15, 20, 30

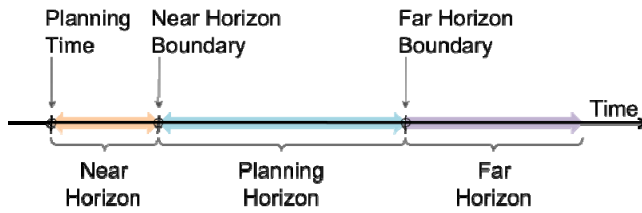


Figure 5. Prediction Horizon

### Performance Metrics

The performance of the system is measured using five metrics: number of successful reroutes, number of failed reroutes, number of popup encounters, number of non-popup encounters, and normalized delay per aircraft.

Each planning cycle the resolver generates a reroute for each predicted encounter that is detected. A successful reroute solves the encounter and avoids the polygon. A failed reroute does not avoid the polygon. The sum of successful and failed reroutes is the total number of resolver attempts to solve predicted encounters. The success to failure ratio is a measure of the resolver performance. Ideally, the resolver should solve all of the predicted encounters, but for the reasons discussed in the resolver section and at end of the planning horizon results section, it does not. The resolver performance does not depend on the quality of the forecast.

Predicted encounters occur when flights are predicted within the planning horizon to encounter forecast polygons. Another type of encounter, called an actual encounter, is identified after the simulation. Recorded flight tracks are checked against polygons from the CWAM nowcasts. Actual encounters are instances where flights actually encountered nowcast polygons. There are two types of actual encounters: popup and non-popup. Popup, also called missed, encounters are cases where the planning system did not have the opportunity to reroute the flight because the encounter was not detected within the planning horizon. This happens because the forecasts are poor. Non-popup encounters are cases where the system had an opportunity to create a reroute. Either the reroute was successful, but the forecast was poor, so the encounter reoccurred. Or, the reroute failed. The analysis here does not distinguish between the two. Figure 6 shows the taxonomy of an encounter.

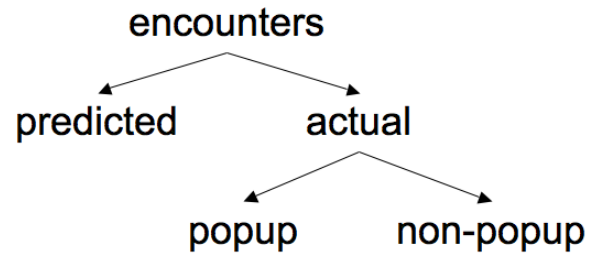


Figure 6. Taxonomy of an Encounter

Normalized delay is the average delay per flight that was rerouted. It is normalized by the time for the flight to travel along its original flight plan from the departure to arrival airport.

### Planning Horizon

Figures 7 and 8 show performance as the planning horizon is reduced. The planning interval was 15 minutes. The baseline case is where the planning system is turned off, i.e. the planning horizon is zero and the planning interval is infinite. For the other cases, the numbers on the horizontal axis are the near and far horizon boundaries. The green and pink bars denote the number of non-popup and popup encounters, and the blue and red bars denote the number of successful and failed reroutes. The percentage of successful reroutes was printed inside the blue bar. Finally, the light blue bars are the normalized average delays. The normalized delay was printed inside of the light blue bar for readability.

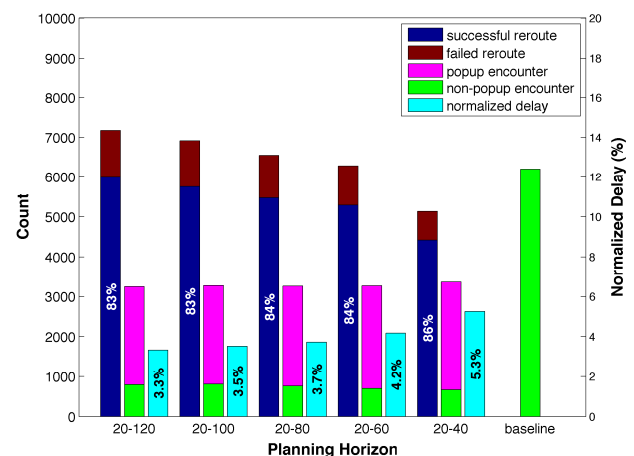
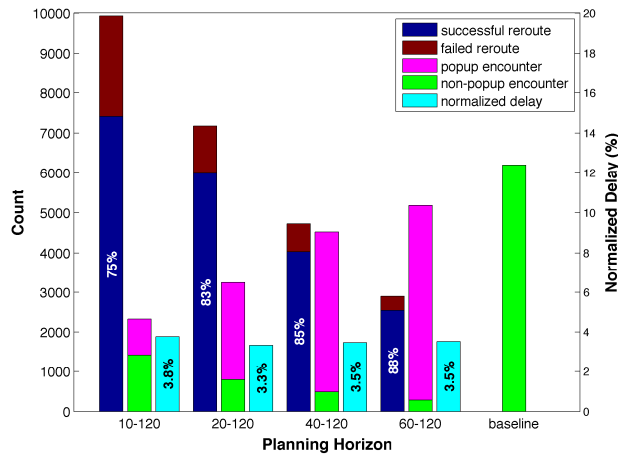


Figure 7. Performance with Varying Far Horizon Boundary (Case 1)



Figure 7 shows performance for case 1 in Table 1. As the far boundary is moved backward, the planning horizon shrinks. Fewer encounters are detected, because ones predicted to occur later are excluded. The number of resolver attempts, denoted by the length of the red and blue bars, goes down. The number of actual encounters, denoted by the length of the pink and green bars, does not change much, and the normalized delay increases.



**Figure 8. Performance with Varying Near Horizon Boundary (Case 2)**

Figure 8 shows performance for case 2 in Table 1. As the near boundary is moved forward, the planning horizon shrinks. Fewer encounters are detected, because ones predicted to occur sooner are excluded. The number of resolver attempts goes down. The number of actual encounters goes up, with most of the increase occurring in popup encounters. The normalized average delay remains approximately constant.

As seen in both figures, the cases with the planning system reduce from the baseline case the number of actual encounters by a large amount. In the 20-120 case, the reduction is 79%, excluding popup encounters. As mentioned before, another system is required to plan reroutes for popup encounters.

The resolver performs only reasonably well. There are still 25% - 12% failures. Many of these are due to polygons with unusual geometries, to many polygons located in a small area, or to encounters occurring at the boundaries of the planning horizon. Using the convex hulls of the polygons could alleviate the problem with unusual geometry. A grid-

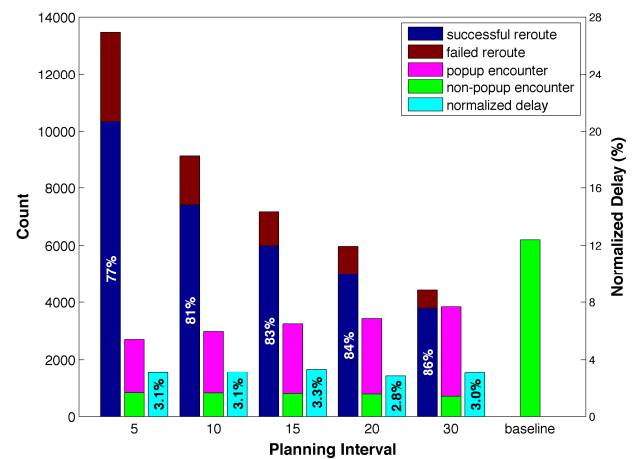
based search, similar to the ones used in [2-5], may perform better, especially when there are many polygons located close together, which causes the method used here to generate reroutes that intersect the neighboring polygons. These performance data form a baseline to compare against future work.

### Planning Interval

Figure 9 shows performance for case 3 in Table 1. The time horizon was 20-120 minutes. As the planning interval is increased, the number of resolver attempts goes down. However, the number of actual encounters slightly increases, with all of the increase in popup encounters. The normalized delays do not change. For the 10-minute interval, the number of non-popup encounters is reduced by 79% from the baseline.

### Number of Reroutes per Hazardous Airspace

The number of reroutes per flight needed to pass a hazardous airspace is an important performance parameter. It is a measure of the quality of the forecast and the reroute. Ideally, this number would be one. In that case, only one reroute would be sent to the flight to get around each hazardous airspace. In this section, this number will be investigated for a single flight and then for the whole simulation. The simulation used had a 20-120 minute planning horizon and 15 minute planning interval.



**Figure 9. Performance with Varying Planning Interval (Case 3)**

UPS2901 traveled from Ontario, California to Philadelphia, Pennsylvania. It departed at 12:00 PM EDT and cruised at 25,000 ft. On 27 July 2006, there

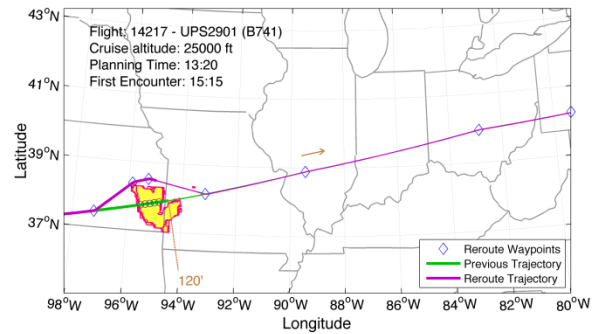
were severe thunderstorms building and decaying from the Midwest to the East Coast. Six hazardous airspaces, caused by convective weather, were encountered by UPS2901. Table 2 summarizes the number of reroutes required to pass each hazardous airspace. The red numbers denote that the reroute at 13:35 failed. At 17:25 the flight had one popup encounter over Pennsylvania. It is not shown in Table 2 or Figures 10-13.

**Table 2. Summary of UPS2901 Reroute Planning Cycles**

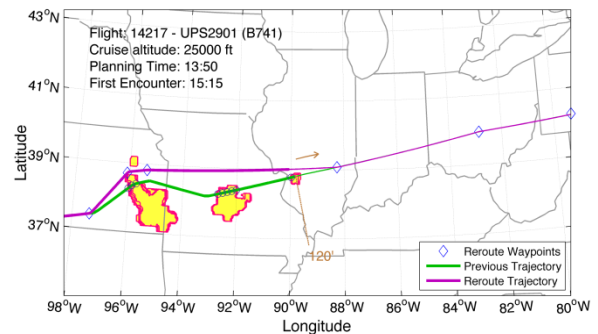
Hazardous Airspace #	Planning Cycle #	Planning Time (EDT)	Time-to-Encounter (minutes)
1	1	13:20	115
	2	13:35	100
	3	13:50	85
	4	14:05	70
	5	14:20	50
	6	14:50	20
2	1	13:35	115
	2	13:50	100
3	1	13:50	120
4	1	14:20	100
	2	14:35	120
5	1	14:50	120
	2	15:20	80
	3	16:05	40
6	1	16:35	25

Airspace 1 was located over the border of Kansas and Missouri. A predicted encounter between it and UPS2901 was detected at 13:20, see Figure 10. There was 115 minutes-to-encounter, see Table 2. The planning system created a successful reroute. However, the 13:20 forecast for airspace 1 was too

small. In subsequent forecasts, airspace 1 grew, and at 13:35, 15 minutes later, an encounter was detected again. The reroute produced by the planner failed. At 13:50, an encounter with airspace 1 was detected again. In addition, encounters with two new hazardous airspaces, 2 and 3 over Missouri and Illinois respectively, were detected. As illustrated in Figure 11, the planner builds a successful reroute around all three hazardous airspaces.



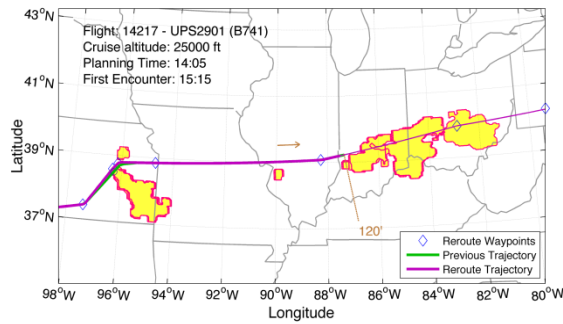
**Figure 10. 13:20 Reroute for Airspace 1**



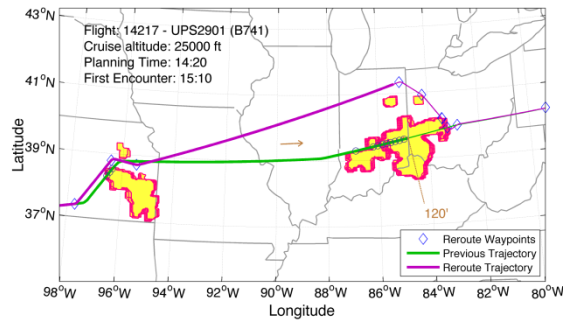
**Figure 11. 13:50 Reroute for Airspaces 1, 2, and 3**

The forecast for airspace 1 continues to change, and at 14:05, UPS2901 is again predicted to encounter it. Figure 12 shows the successful reroute. Notice that a new large hazardous airspace over the junction of Indiana, Ohio, and Kentucky appears just past the far planning horizon boundary, marked as 120° in Figure 12. This airspace is labeled 4 in Table 2. At 14:20, the system builds a successful reroute for airspaces 1 and 4. The reroute is presented in Figure 13. One more reroute, generated at 14:50, was needed to pass UPS2901 around airspace 1.



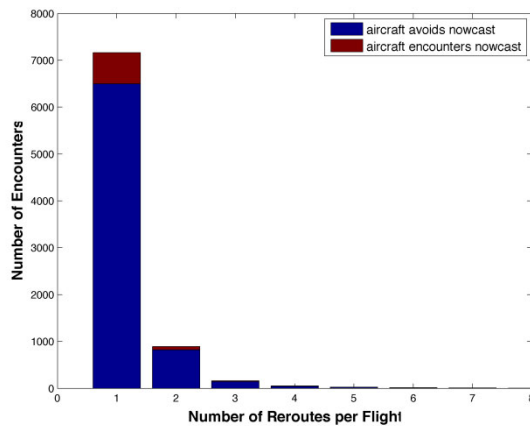


**Figure 12. 14:05 Reroute for Airspace 1**



**Figure 13. 14:20 Reroute for Airspace 1 and 4**

Table 2 and Figures 10-13 illustrated the number of reroutes per hazardous airspaces for one flight. For the whole simulation, Figure 14 shows a histogram of the number of reroutes per flight needed to pass an encounter with a hazardous airspace. The majority of encounters required only one reroute. In the case of UPS2901, the encounter with airspace 1 required 6 reroutes, which is high. The other airspaces needed one to three, which is normal. The few encounters with 7 or higher reroutes were found to have anomalous reroutes.



**Figure 14. Histogram of the Number of Reroutes per Flight Required to Pass an Encounter**

## Conclusions

An automated planning system for rerouting flights around airspace containing convective weather was developed. The system detects encounters between flights and undesirable airspace. It builds reroutes for all encounters predicted within a specified planning horizon. To compensate for uncertainty in the weather forecast, it periodically re-plans at a specified planning frequency. The algorithm generates the reroutes using a simple geometric approach, which identifies a path around the shortest side of the airspace. When a flight has more than one encounter within the time horizon, the encounters are solved in order of earliest to latest.

The planning system was implemented in a model of the national air traffic system. Simulations show that the planning system with a horizon of 20-120 minutes and an interval of 15 minutes is able to solve 79% of encounters that are detected. The reroutes delay flights on average about 3.3%. The algorithm that generates the reroutes is successful only 75% to 85% of the time. In most cases, one to three reroutes were required to route a flight around a hazardous airspace, but for the worst case, six reroutes were required.

It is still unclear which planning horizon and frequency performs best. This is because the resolver needs to have a higher success rate. Based on the results shown here, extending the far horizon boundary past 40 minutes does not cost more missed encounters or excessive delays and reduces the number of reroutes. Moving the near horizon as far back as possible (10 minutes in this paper) reduces the number of actual encounters and increases the number of reroutes. However, the rerouting algorithm did not perform well for the 10-120 minute horizon (25% failures). Reducing the planning interval decreases the number of actual encounters and increases the number of reroutes.

Future work will investigate several items. The performance of the algorithm that generates the reroutes needs to be improved. This could be accomplished by using a grid-based method and/or modifying the polygons so that they are convex. In addition, polygons located in close proximity could be combined into one. The reroutes could be optimized to minimize delays by generating alternate routes and selecting the best one. The reroutes need

to be analyzed to understand their affect on traffic congestion and flight separation.

## References

- [1] Sridhar, Banavar, Shon Grabbe, and Avijit Mukherjee, December 2008, "Modeling and Optimization in Traffic Flow Management," *Proceedings of the IEEE*, Vol. 96, No. 12, pp. 2060-2080.
- [2] Grabbe, Shon, Banavar Sridhar, and Avijit Mukherjee, May 2009, "Sequential Traffic Flow Optimization with Tactical Flight Control Heuristics," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, pp. 810-820.
- [3] Hok, Ng, Shon Grabbe, and Avijit Mukherjee, August 2009, "Design and Evaluation of a Dynamic Programming Flight Routing Algorithm Using Convective Weather Avoidance Model," AIAA-2009-5862, Chicago, Illinois, AIAA Guidance, Navigation, and Control Conference.
- [4] Krozel, Jimmy, Steve Penny, Joesph Prete, and Joeseph Mitchell, 2007, "Automated route generation for avoiding deterministic weather in transition airspace," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, pp. 144-153.
- [5] Krozel, Jimmy, Changkil Lee, and Joseph Mitchell, 2006, "Turn-Constrained Route Planning for Avoiding Hazardous Weather," *Air Traffic Control Quarterly*, Vol. 14, No. 2, pp. 159-182.
- [6] Sridhar, Banvar, Gano Chatterji, Shon Grabbe, and Kapil Sheth, August 2002, "Integration of Traffic Flow Management Decisions," AIAA-2002-5014, Monterey, CA, AIAA Guidance, Navigation, and Control Conference.
- [7] Love, John, William Chan, and Chu Han Lee, September 2009, "Analysis of Automated Aircraft Conflict Resolution and Weather Avoidance," AIAA-2009-6995, Hilton Head, South Carolina.
- [8] Chan, William, Mohamad Refai, and Richard DeLaura, 2007, "An Approach to Verify a Model for Translating Convective Weather Information to Air Traffic Management Impact," AIAA-2007-7761, Belfast, Northern Ireland, AIAA Aviation Technology, Integration, and Operations Conference.
- [9] DeLaura, Rich, and James Evans, January 2006, "An Exploratory Study of Modeling Enroute Pilot Convective Storm Flight Deviation Behavior," 12<sup>th</sup> Conference on Aviation, Range and Aerospace Meteorology, American Meteorological Society, Atlanta Georgia.
- [10] DeLaura, Rich, Mike Robinson, Margo Pawlak, and Jim Evans, 2008, "Modeling Convective Weather Avoidance in Enroute Airspace," 13<sup>th</sup> Conference on Aviation, Range, and Aerospace Meteorology, American Meteorological Society, New Orleans, LA.
- [11] Nuic, A., July 2003, "User Manual for the Base of Aircraft Data (BADA), Revision 3.5," EEC Note No.11/03, Eurocontrol.
- [12] Meyn, Larry, Robert Windhorst, Karlin Roth, Donald Van Drei, Greg Kubat, Vikram Manikonda, Sharlene Roney George Hunter, Alex Huang, and George Couluris, August 2006, "Build 4 of the Airspace Concept Evaluation System," AIAA-2006-6610, Keystone, Colorado, AIAA Modeling and Simulation Technologies Conference.
- [13] Zelinski, Shannon, and Larry Meyn, August 2006, "Validating the Airspace Concept Evaluation System for Different Weather Days," AIAA-2006-6115, Keystone, Colorado, AIAA Modeling and Simulation Technologies Conference.

## Acknowledgments

Acknowledgement is due to Rich Pawlowicz, author of `m_map`, [http://www.eos.ubc.ca/~rich/#M\\_Map](http://www.eos.ubc.ca/~rich/#M_Map). He created a map projection package for Matlab, which the authors used to render some of the figures.

*28th Digital Avionics Systems Conference  
October 25-29, 2009*