

# Planning Routes with Wiggle Room in En Route Weather-Impacted Airspaces

Valentin Polishchuk, Ph.D. <sup>\*</sup>, Anne Pääkkö <sup>†</sup>  
*Helsinki Institute for Information Technology, Helsinki, FI-00014*

Joseph S. B. Mitchell, Ph.D. <sup>‡</sup>, and  
*State University of New York, Stony Brook, NY, 11794*

Jimmy Krozel Ph.D. <sup>§</sup>  
*Metron Aviation, Inc., Dulles, VA, 20166*

We investigate a form of robust route planning by imbedding path-stretch zones into routes that avoid hazardous weather constraints in en route airspace. Along every path, every so often, we plan for “wiggle room” – a constraint-free triangular area whose purpose is to control the delay of a flight by executing a path-stretch maneuver at these locations. We designed and implemented an algorithm that computes the paths by searching an underlying disk packing in the “bottommost” fashion. Our solution allows the user to specify the constraints for the flights, the size of the path-stretch zones, and the maximum and minimum distance between the consecutive zones along a path. The paths computed by our algorithm fulfill a number of additional requirements, e.g., we take care of never placing the path-stretch zones on top of the sector boundaries. We present theoretical guarantees provided by the algorithm and outputs produced on sample airspaces.

## Nomenclature

<i>2D</i>	= 2 Dimensional	<i>NAS</i>	= National Airspace System
<i>3D</i>	= 3 Dimensional	<i>NextGen</i>	= Next Generation Air Transportation System
<i>4D</i>	= 4 Dimensional	<i>nmi</i>	= nautical mile
<i>AFP</i>	= Airspace Flow Program	<i>NWS</i>	= National Weather Service
<i>ATM</i>	= Air Traffic Management	<i>RNP</i>	= Required Navigation Performance
<i>ATSP</i>	= Air Traffic Service Provider	<i>RTA</i>	= Required Time of Arrival
<i>DAC</i>	= Dynamic Airspace Configuration	<i>SUA</i>	= Special Use Airspace
<i>DST</i>	= Decision Support Tool	<i>TBO</i>	= Trajectory-Based Operation
<i>EDA</i>	= En Route Descent Advisor	<i>TFM</i>	= Traffic Flow Management
<i>FCA</i>	= Flow Constrained Area	<i>TM</i>	= Trajectory Management
<i>FMS</i>	= Flight Management System	<i>VNAV</i>	= Vertical Navigation
<i>LNAV</i>	= Lateral Navigation	<i>WAF</i>	= Weather Avoidance Field

## I. Introduction

Contingency planning is a fundamental task in Air Traffic Management (ATM). With the increasing volume of air traffic expected in the Next Generation Air Transportation System (NextGen), and with the introduction of Trajectory-Based Operations (TBO), new methods of 4-Dimensional (4D) Trajectory Management (TM) will emerge. Managing 4D time-space trajectories of individual flights and their interactions with hazardous weather will require the assistance of Decision Support Tools (DSTs); these DSTs will include algorithms that reason with the geometry of the forecasted weather and traffic demand situation. In NextGen, as well as for today’s National Airspace System (NAS), there is a need for local adjustment of the time-of-arrival of a flight at certain points of the

<sup>\*</sup> Postdoctoral researcher, University of Helsinki and Helsinki University of Technology

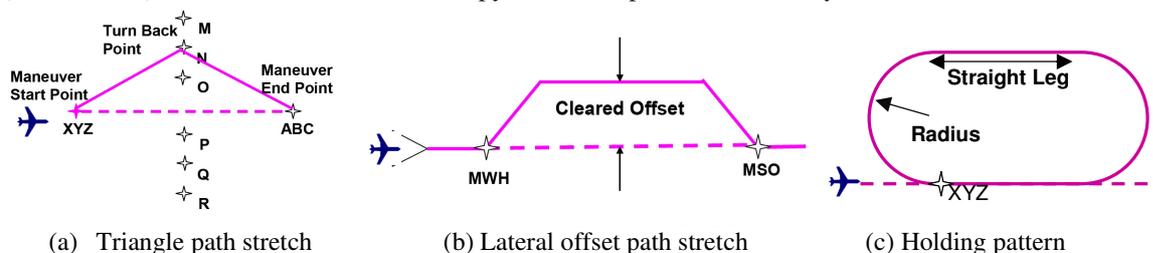
<sup>†</sup> Graduate Student, Department of Computer Science, University of Helsinki

<sup>‡</sup> Professor, Department of Applied Mathematics and Statistics, Stony Brook University

<sup>§</sup> Senior Engineer, Research and Analysis Department, 45300 Catalina Court, Suite 101, AIAA Associate Fellow

airspace, e.g., the sector boundary, boundaries of Flow-Constrained Areas (FCAs), or the boundary of the Terminal area. In many cases, such local time-of-arrival adjustment is achievable via speed control and/or path stretching (path stretching preferred over speed control given that the engines receive less wear this way). In extreme cases, adjustment is performed by the use of circular holding patterns – a spatially expensive solution used today that we wish to avoid in NextGen. The focus of this paper is on the automatic design of weather avoidance routes that include path-stretch maneuvers to address time-related uncertainties.

In the NextGen, we anticipate that standard maneuvers will be accurately performed by the Flight Management System (FMS) of an aircraft. These maneuvers (and their associated parameters) can conceivably be commanded by the Air Traffic Service Provider (ATSP) via a data link in order to control traffic flow properties. Examples of standard FMS maneuvers and some key parameters are shown in **Figure 1**. Path stretching alone may be used to delay traffic for small amounts of time (3-5 min); coupled with speed control, path stretching also allows absorption of longer delays. The stretching is implemented by a horizontal maneuver, in which the aircraft temporarily deviates from the nominal path typically by using either a triangle path stretch or a lateral offset path stretch, thereby increasing the overall path length. In principle, the maneuver can be executed at any point of the path and potentially at different points for each aircraft; however, this would create a very complex situation to monitor by controllers. Alternatively, some number of special regions of the airspace can be reserved for the path-stretch maneuvers in advance, when designing the air lanes for the traffic flow. Such advance planning will reduce the voice communication overhead (transferring it to a computer-to-computer communication, i.e., from ATSP computer to FMS computer), and will give rise to a more stable control of the flow. Another advantage of such planning is the possibility to maximize throughput, by providing the path-stretch zones along the routes only as often as it is needed (or forecasted). That is, the zones do not occupy more free space than necessary.



**Figure 1. In NextGen, most FMS systems will provide the ability to execute path stretch maneuvers (triangle or parallel offset) at any point in space (including today’s standard Navaid locations).**

The ability to command path-stretch maneuvers addresses a robustness property to path planning. Historically, weather forecasts have had errors associated with the growth and decay of the weather cells included in the forecast. So, for instance, if two weather cells have a gap between them and an aircraft is expected to fit through that gap, then the uncertainty associated with the size of the gap must be taken into account when timing the passage of an aircraft through the gap. If the plan is to expect a nominal gap size, but the uncertainty indicates that the gap size is smaller or larger than planned, then contingency plans must be in place in case aircraft must be delayed for a given period of time to allow the weather cells to decay and open the gap up. When route planning algorithms reason about the weather forecast without taking into account the uncertainty associated with the weather forecasts (or uncertainty about the geometry of the weather hazards), then the properties of traffic flows are adversely affected. Inevitably, weather forecasts will have some error, and the path-stretch maneuvers in our approach provide the wiggle-room (located in clear airspace, free of hazards) for adjusting the time at which traffic arrives at gaps opening up or closing between weather cells.

## II. Background

The En Route Descent Advisor (EDA) [1] is a DST that helps to deliver aircraft at a specified point according to the schedule, while avoiding violation of separation with other aircraft. One of the trajectory processes underlying EDA is the Path-Stretch mode. The mode is based on a simple “dogleg” maneuver. EDA is one of the building blocks of the novel TBO routing technology introduced in [2], [3], [4], [5]. This new operational concept aims at a significant improvement in operation efficiency, increasing airspace capacity, and taking full advantage of the currently available systems functionality. One of the features of the new technology is the possibility of resolving local conflicts without the airplanes leaving the Lateral Navigation (LNAV) and/or Vertical Navigation (VNAV) modes of the FMS. Path delay is an important part of the planning mechanism developed in [2], [3], [4], [5]. The delay is used as a means of resolving predicted loss of separation between aircraft. The delay mechanism is very flexible

and effective in assuring the separation. It works by adjusting the amount of length variation, necessary to achieve the change of the timing at the conflict point as desired.

In our work, we build on some of the findings in [3], when deciding on the shape of the path-stretch zones. We note that path stretch maneuvers may be used to control time of arrival at boundary crossing locations, at merge points, and at metering fix locations. At any point downstream where there is a Required Time of Arrival (RTA) at a point, upstream path stretch maneuvers can help insure that these RTAs are met. For instance, in the case of Airspace Flow Programs (AFPs) [6], [7], it is necessary to control the flow rate crossing into an FCA, which can be achieved by adjustment of individual path-stretch maneuvers along several routes leading into the FCA.

Theoretical methods and numerical results for capacity estimation of en route airspace were presented in [8], [9]. In our previous work [10], the performance of the Flow-Based Route Planner [11] was evaluated against the theoretical upper limit on capacity. In that work, it was shown that “parallel” non-intersecting flows of traffic provide a high throughput, low complexity Traffic Flow Management (TFM) solution for passing traffic across an airspace. In [12], a large set of real data was analyzed by artificial intelligence techniques to identify robust routes.

The problem addressed in this paper – that of judiciously placing the path-stretch zones along the routes while achieving the maximum capacity of an airspace – has not been studied before. This work complements past efforts in providing high capacity with the ability to adjust flow properties through path-stretch maneuvers. This can be particularly useful in situations where the uncertainty in the situation cannot be avoided until a tactical time frame.

### III. Modeling

In this section, we give a formal description of the model that we employed. We consider aircraft in the en route portion of the flight, at a constant flight level. This makes our path planning problem 2-dimensional (2D) in space (or 3-Dimensional (3D) in space-time). In fact, even in the terminal area, the 2D projection of a full 3D path may serve as a sufficient specification of the path for planning wiggle room. This is true, in particular, when the plane follows a predefined descent profile, or the altitude is specified along the waypoints of the path [2], [3], [5].

#### A. Airspace Boundaries

Our algorithm works for a general-shaped airspace (center, sector, FCA, etc.). Initially the airspace is modeled by a rectangle, to simplify our implementation. Later, we discuss how our implementation takes care of more complex sector geometry.

The West (left) side of the rectangle is designated as the source edge; it is through this edge that the traffic enters the airspace. The East (right) side of the rectangle is the sink, through which the traffic must exit. The assumption of mostly unidirectional flow is justified by the fact that the traffic with West-to-East heading is altitude-separated from East-to-West traffic according to the “Alternating Altitude Rule”.

#### B. Airspace Constraints

The constraints for the air traffic are given by no-fly zones (e.g., Special Use Airspace (SUA) boundaries) and by hazardous weather cells within the airspace. The no-fly zones are human-specified, and hence there is no ambiguity in defining SUA boundary conditions. On the contrary, objectively understanding what weather cells constitute obstacles for an aircraft, is not so straightforward.

Defining the boundary for convective weather hazards is an active research topic for ATM researchers. The National Weather Service (NWS) [13] provides a scale for classification of convective weather based on the reflectivity level. Early studies show [14], [15] that pilots tend to avoid weather cells with NWS level 3 and higher (reflectivity higher than 41 dBZ). Follow-on research has shown that there are numerous other factors that contribute to determining what regions of the airspace should be marked as obstacles: the structure and shape of the weather cells, accuracy of prediction, pilot experience, airline policies, altitude of reflectivity echo tops etc. [16], [17], [18]. These models are being researched for determining if certain routes are blocked by convective weather and for defining what general geometric constraint regions should be considered to be a weather-induced obstacle at a given flight level. These constraints are referred to as Weather Avoidance Fields (WAFs), for a given forecast look-ahead time and flight level.

In some airlines, dispatchers draw their own boundaries around regions of hazardous weather. In this way, airline policy may be taken into consideration after viewing convective weather forecasts or turbulence forecasts.

In view of this ambiguity in delineating obstacles boundaries from the weather data alone, we decided to offer the user the ability to input the obstacles manually or by thresholding the weather forecast data at a user-preferred level. Given a snapshot of real weather, the user may explicitly specify the areas identified as obstacles, based on the observed severity of the weather cells. In particular, the user can add the necessary safety margins around the weather cells, so that the paths do not come close to the hazardous features.

In what follows, we will not differentiate between the constraints induced by the no-fly zones and the constraints induced by hazardous weather cells, possibly, with an added safety margin. We will use the term *obstacles* for any region through which flying is not permitted.

### C. Route Structures

We model flight paths of aircraft as polygonal paths connecting the source edge to the sink edge on the airspace boundary. Formally, a polygonal path is a sequence of straight-line segments, which are pairwise disjoint except that consecutive segments share a common endpoint. The endpoints of the segments are called the vertices of the path. The vertices correspond to the flight waypoints. In any collection of feasible paths, the paths must be (laterally) separated according to the Required Navigation Performance (RNP) parameter. Typical RNP values are 1-10 nmi, depending on the type of aircraft and domain of flight (en route or terminal area). In NextGen, we expect that RNP near 1 nmi may be used to optimize traffic throughput.

### D. Wiggle Room

The model described up to now has been one of the standard abstractions used in route planning for ATM. A variety of computational-geometry tools have been applied in the past to plan the paths according to the model [10]. In this paper, we add another twist to the motion planning problem: advance positioning of the path-stretch “wiggle room” along routes. In the model, we use the analogy of a string “necklace” whose “beads” correspond to the locations for possible (contingency) time stretch maneuvers. The paths are the strings (“threads”) of the necklaces, and the beads are the locations where additional space for potential path stretch maneuvers will be planned.

### E. Wiggle Room Shape

Since the beads correspond to areas for implementing path-stretch maneuvers, the shape of the beads is very important for the ATM application. For instance, while circular beads do provide space for the wiggle room along the paths, the circular shape is suboptimal to fit actual path-stretch templates (for example, as shown in **Figure 1**). Reference [3] discusses different ways to implement path stretch, paying attention to a variety of factors that influence the efficiency of the implementation: voice communication overhead, necessary data storage and retrieval, etc. They also consider path stretch via lateral path offsets, which, in our terms, corresponds to necklaces with trapezoidal beads. The overall conclusion of [3] is that the most suitable, at least for current flight operations, is the implementation of path-stretch maneuvers via a “dogleg”, which in our terminology corresponds to triangular shaped beads. Triangles most closely resemble today’s maneuvers employed by controllers [3]. Taking this into account, we implemented our path finder so that it produces necklaces with sufficient space to imbed triangular beads.

### F. Formal Definition

Mathematically, a necklace is defined by two parameters: the stretch  $L$  and the bead size  $r$ . For a triangular bead, we define its size as the radius of the circle circumscribing the triangle. We define a necklace to be a (source-sink) polygonal path with the following properties:

- *Stretch*: the distance between consecutive vertices of the path is at most  $L$ .
- *Free space*: at each vertex of the path, there exists an obstacle-free triangular bead of size  $r$ .
- *Independence*: the beads, located at different vertices of the path, are disjoint.

A collection of necklaces is *feasible* if the necklaces in the collection are pairwise-disjoint. That is, no necklace is allowed to intersect another necklace’s thread nor any of its beads. Refer to **Figure 2**.

In ATM terms, the beads correspond to wiggle room along a route. The stretch property ensures that the wiggle room happens often enough, along a path, to provide flexibility to deliver a necessary time delay or RTA. The free space property ensures that the path-stretch maneuvers happen in areas unconstrained by hazardous weather cells. The independence property means that aircraft executing the path-stretch maneuvers in different wiggle room locations do not conflict with each other. The independence property allows for the flow rate of traffic on one flow to be adjusted independent of the flow rate on another flow.

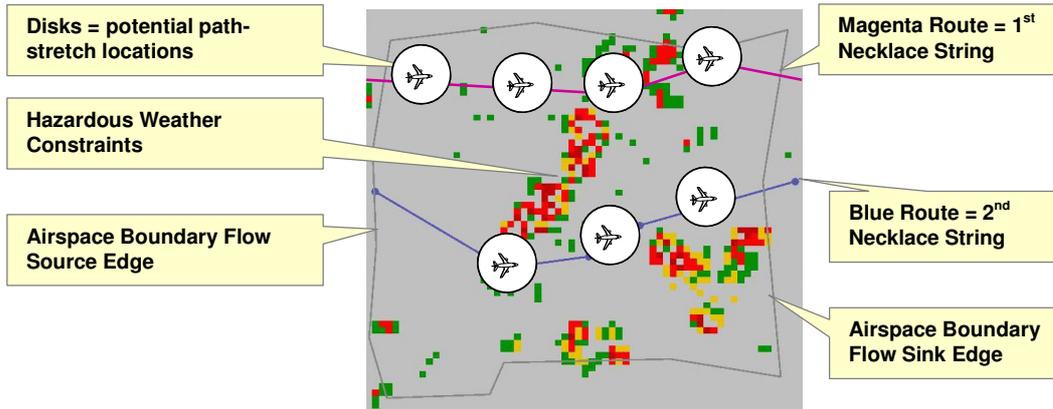


Figure 2. Two necklaces (with circular beads) within an airspace boundary.

#### IV. Solution Approach

Our solution approach is to implement a well-defined algorithm with theoretical guarantees.

##### A. Algorithm

The algorithm we implement finds a maximum number of necklaces in a given polygonal domain by iteratively routing “bottommost” necklaces across the domain. The search is guided by a hexagonal packing of disks of radius  $r$ . Within the packing, we find the necklaces by doing a depth-first search. The steps of the algorithm are shown in **Figure 3**. In the remainder of this section we give the details of the algorithm.

We start with a hexagonal packing of radius- $r$  disks within the airspace (**Figure 3a**). Disks intersected by obstacles are removed (**Figure 3b**); they are not safe locations for wiggle room. Then we start iteratively searching for necklaces. We first find necklaces with circular beads, and then transform the beads to triangles, the shape of the path stretch maneuvers that we are planning (see **Figure 1a**).

For a route, we choose the bottommost disk at the source to be the first bead of the necklace (the light orange disk in **Figure 3b**). This disk is marked as the *current* disk. We then look at all disks whose centers are reachable from the current disk by a straight-line segment of length  $L$  (disks colored yellow in **Figure 3b**). Among these disks we choose the rightmost (bottommost) as the next bead (the disk with thick blue boundary in **Figure 3b**). The disk becomes the current bead, and the process is repeated: the rightmost reachable disk is chosen as the next bead (the green disk in **Figure 3b**), it becomes the current bead, and the next bead is chosen (the blue disk in **Figure 3b**).

The routing of a necklace is concluded when a disk at the sink is reached (**Figure 3c**). Next, the disks intersected by the necklace thread or touched by a bead are removed from consideration (**Figure 3d**). This is done to enforce separation between the routed necklace and the next necklace to be routed.

The next necklace is routed by the same procedure. The bottommost disk at the source is chosen as the first bead of the necklace (the light orange disk in **Figure 3d**), the rightmost disk reachable from it becomes the next bead (the yellow disk with thick blue boundary in **Figure 3d**), and so on. Again, after the necklace is found, the disks that are in contact with the necklace, are removed.

The routing is performed in a depth-first-search manner. If there is nowhere to go from the current disk, we “pop” the current disk and retract one step back. When all disks at the source have been popped, the maximum number of necklaces has been found, and the search stops (**Figure 3e**). We then do a cleanup – remove the disks not used by any necklace, and transform the circular beads into triangular ones (**Figure 3f**). This is the final output of our algorithm.

##### B. Algorithm Guarantee

We can prove that our algorithm provides the following theoretical guarantee: If there exist  $K$  necklaces with stretch  $0.75L$  and beads of size  $3r$ , we will find (at least)  $K$  necklaces with stretch  $L$  and size- $r$  beads. That is, our algorithm is a “dual-approximation” one: it finds an optimal number of paths, albeit for slightly “worse” necklaces – ones with slightly smaller beads and larger stretch.

In ATM terms, this means that we are able to achieve the *maximum available* capacity of the airspace, although the path stretch maneuver zones found by our algorithm are not as large as they potentially could have been. As for the number of path-stretch locations, we will find approximately  $3/4$  of the *maximum possible* number of locations

along every path. For instance, if it is (just) possible to have 4 stretch zones along a path, we might find only 3 zones (though, often we will still find the optimal, 4). This should provide sufficient opportunities for path stretching.

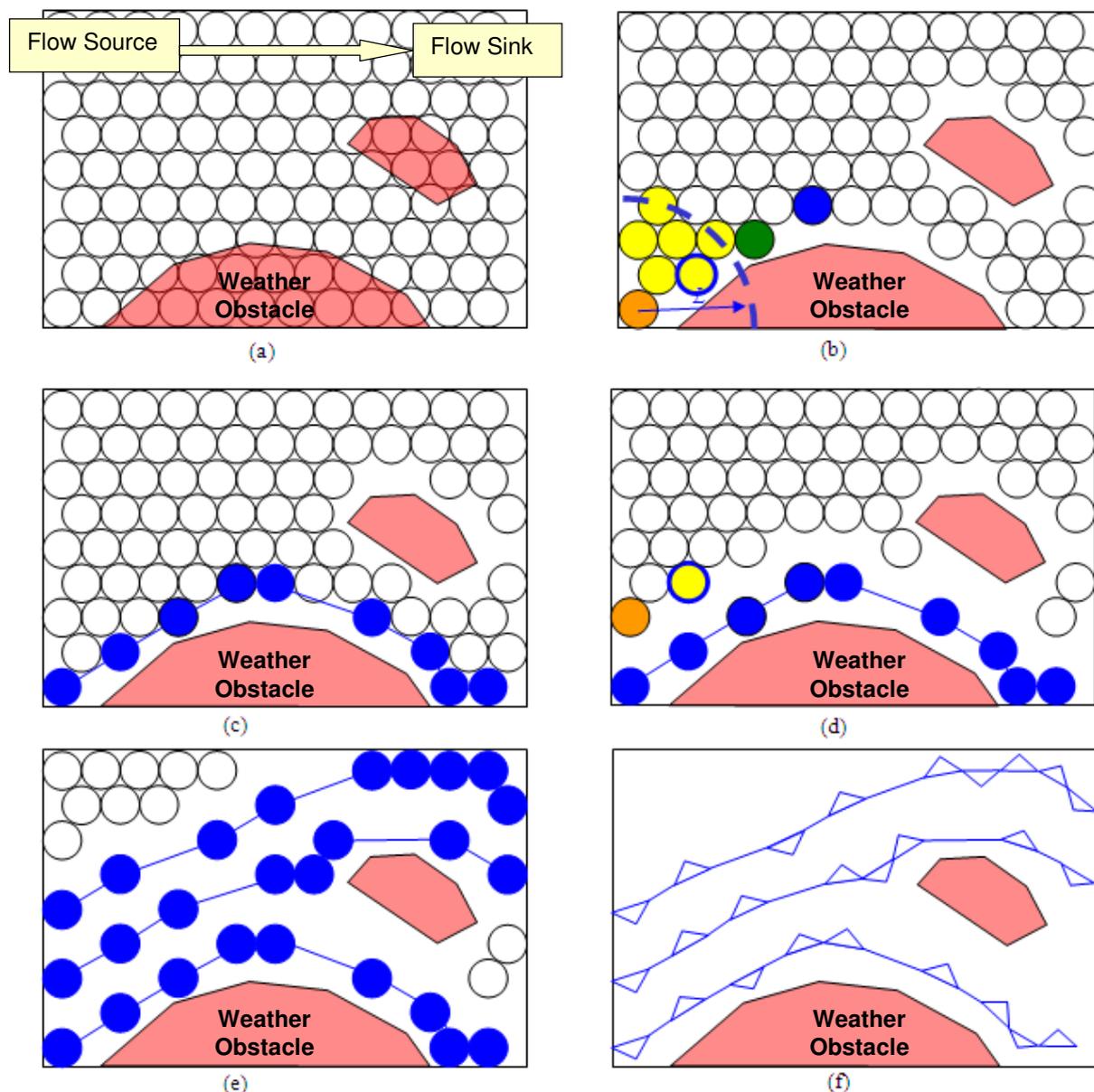


Figure 3. Steps of the algorithm.

(a) Disk packing in the given polygonal domain. (b) Disks intersecting obstacles are removed, and the routing begins from the bottommost disk at the source (light orange). The disk is marked as current, and all disks reachable from it with a length- $L$  segment, are found (yellow disks). Among those, the rightmost disk is chosen (yellow, with thick blue boundary) as the next bead. The next bead becomes the current bead, and the search continues – green and blue disks are subsequent beads of the first necklace. (c) The first necklace reaches a disk at the sink. (d) The disks, touching the first necklace are removed, and the second necklace starts being routed from the bottommost available disk at the source (the light orange disk). The rightmost disk, reachable from the current disk (the yellow disk with thick blue boundary) is the next bead of the second necklace. (e) The routing continues in the depth-first-search fashion until no more necklaces can be routed. (f) The disks, not used as the necklaces' beads, are removed, and the beads are transformed into the desired standardized path-stretch triangles.

## V. Additional Requirements for Paths

In this section we describe few additional constraints that our algorithm is capable of handling.

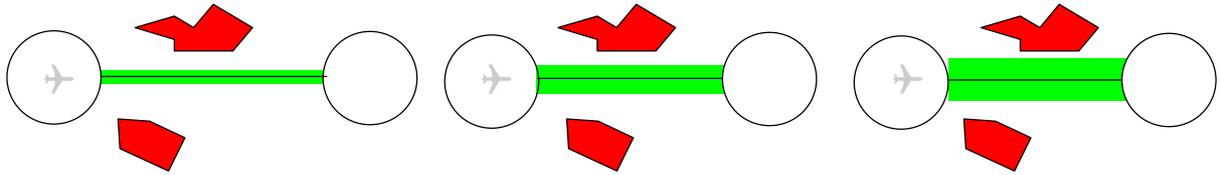
### A. Lower bound on Path Stretch

A first possible criticism of the output in **Figure 3** is the fact that beads may happen “too often” along a path. To address this, we introduce a lower bound,  $\ell \geq 2r$ , on the stretch of the thread of a necklace between two consecutive beads. Specifically, we require that the distance between two beads along the same necklace lies in the interval  $[\ell, L]$ . In this way, no two beads touch each other, and there is always some spacing between consecutive beads. The ATM application requires strategically located (clear of weather and uniformly dispersed) path-stretch maneuvers to provide wiggle room across the sector or airspace domain that needs robustness to uncertainties.

This is easy to implement within the framework of bottommost necklaces. When searching from the current bead for the next bead, we consider, as possible candidates for the next bead, only those disks whose center lies at distance at least  $\ell$  and at most  $L$  from the (center of) the current bead. This ensures that the beads are spaced out, with no two consecutive beads of one necklace touching each other.

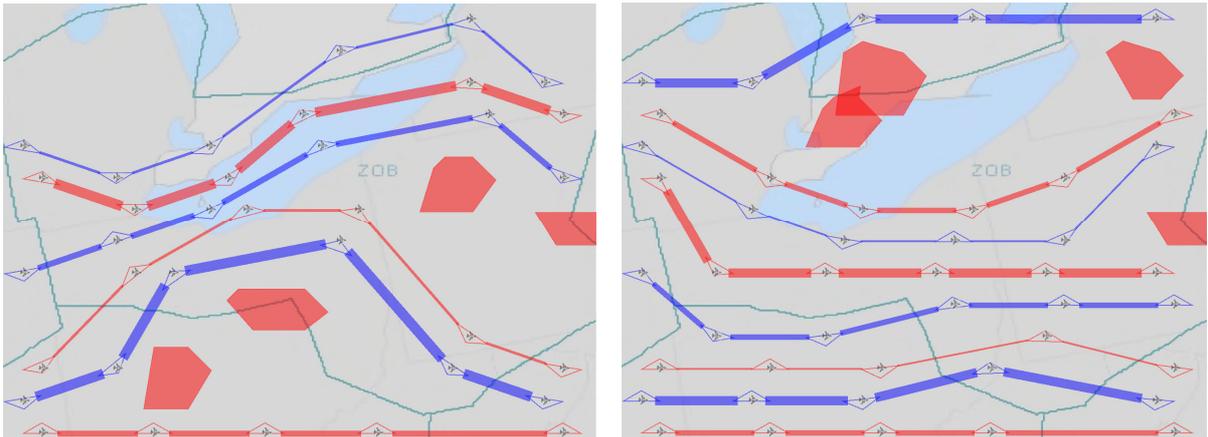
### B. RNP Requirements between Beads

Our algorithm connects paths with segments that do not intersect weather constraint obstacles. This path may range from “pencil thin” to really “thick” (**Figure 4**); the thickness represents the RNP of the aircraft that fly along the route. In this way, the thick path models the region of containment for all traffic within the flow, and represents the robustness of the flight with respect to the capability of the aircraft to follow the route. By definition, the RNP requirement of RNP- $x$  means that an aircraft stays within  $x$  nmi of the centerline between waypoints 95% of the time. For 99.999% containment, we define the “thick path” width to be RNP-2. Thus, a thick path of thickness of RNP-2 should not ever touch any weather hazard constraint. The output of our algorithm with paths having different RNP requirements is presented in **Figure 5**.



(a) High precision aircraft (low RNP) (b) Medium precision (medium RNP) (c) Low precision aircraft (high RNP)

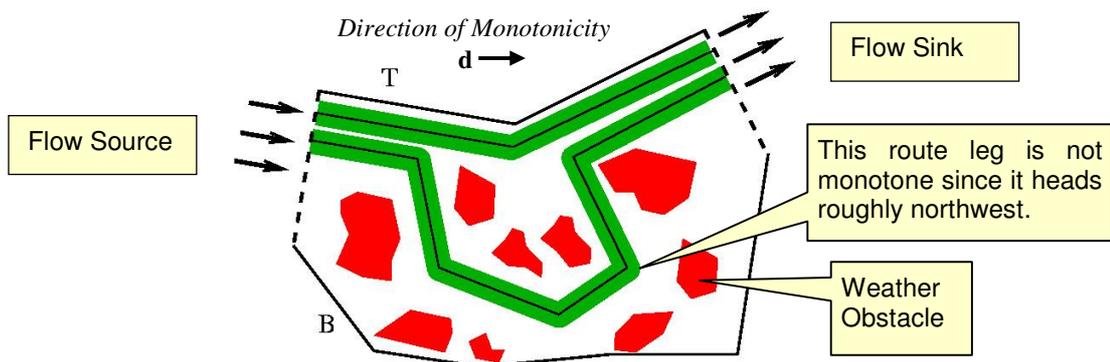
**Figure 4. Thick paths to represent RNP for a lane.**



**Figure 5. Routes with different RNP requirements (path thickness).**

### C. Turn Constraints

We may also impose turn constraints so that the paths do not turn too sharply. In most ATM applications, flows progress monotonically across an airspace domain, as illustrated in **Figure 6**, and flows that turn perpendicular or “turn back” on themselves are not desirable. To incorporate such turn constraints, while searching for the next bead, we locally ensure that the next leg of the path does not make too sharp an angle with the previous one.



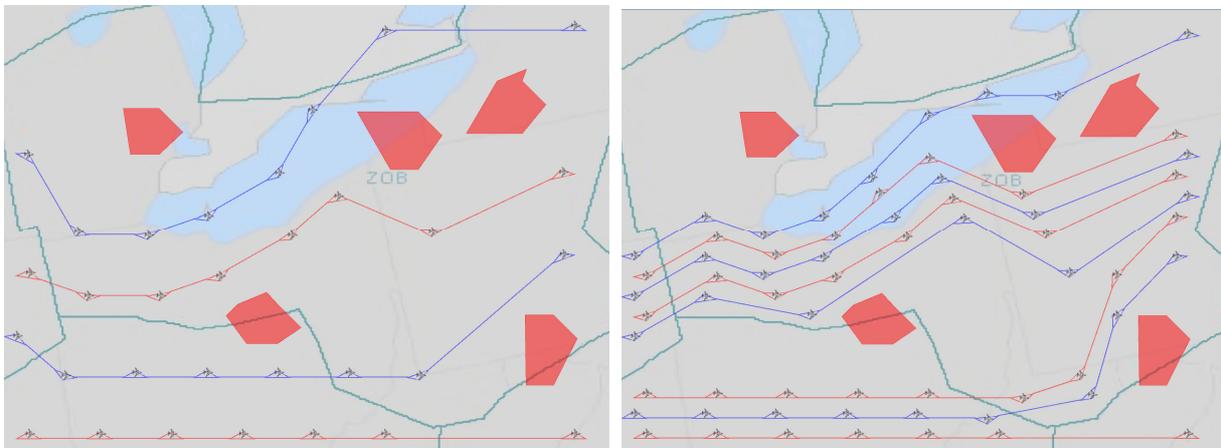
**Figure 6.** The northern air lane is monotone (west-to-east) with a single turn (waypoint), while the southern air lane is not monotone.

#### D. Specification of Path Entry Points

Our algorithm automatically decides the start and exit points for the paths. The entry points are dictated by available disks at the source, and the exit points are where the necklaces end. Note that within our framework, it is easy to give the user the ability to define the entry points. Indeed, the controller may specify a few points along the source, through which he would like to see the aircraft enter into the airspace, in order to control where he expects to have aircraft handed off to him (from upstream controllers), based on typical flows of traffic. Our algorithm would then search for the necklaces exactly in the same way as before, but starting from these predefined points. In the current version of the program this is not implemented.

#### E. Separation between Paths

One of the user-controlled parameters in our implementation is the separation between the neighboring paths. Naturally, with a larger separation fewer routes are found, and vice versa (**Figure 7**).



**Figure 7.** Separation between the routes may be specified by the user.

### VI. Additional Requirements: Airspace Boundary Geometry

Up to now we considered a simplified scenario in which the airspace is represented by a rectangle. In reality, of course, the sector may have a complicated geometry. We now discuss the challenges that arise from this and how they are addressed with our algorithms.

#### A. No Path Stretching over Airspace Boundaries

One would expect that a controller would delay aircraft inside his sector and complete the maneuver prior to handing off the aircraft to another controller in an adjacent sector. This translates into another constraint: not to have path stretching occur as an aircraft is passing over sector or airspace boundaries. In order to address this constraint, the disks intersecting the sector boundary are not considered as potential beads. Then, no bead is ever placed on top of the sector boundaries.

### B. Requiring Routes to Stay within Airspace Boundaries

Another constraint is to try to have the path legs between path stretch maneuvers fully stay inside the sector or airspace boundary. That is, suppose a request to route 4 paths through the sector is issued. We can first try to accommodate the paths within the sector. For that, we treat the upper and lower boundaries of the sector as obstacles, and also remove all disks that lie outside the sector. We then search for the maximum number of paths in the sector. If we are not able to find 4 paths, we report “Not possible”. We can then relax the top and/or bottom constraints, allowing the paths to “spill” into adjacent sector(s) (e.g., **Figure 8**), and search for 4 paths within the extended airspace. In the current version of the program this is not implemented.

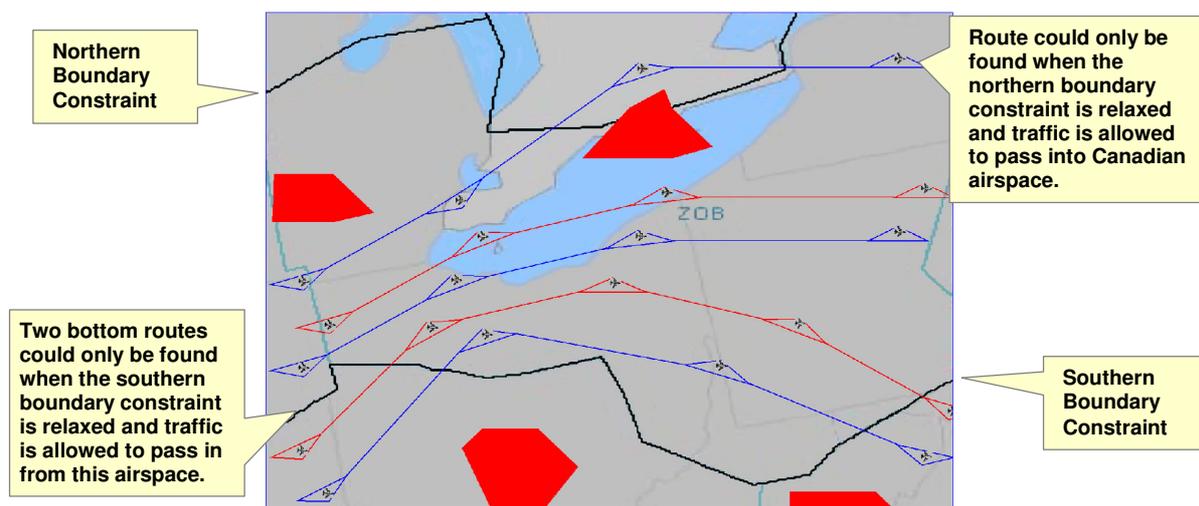


Figure 8. Routing solution found by relaxing the Northern and Southern Sector Boundary constraints.

### C. Crossing Airspace Boundaries when Necessary

Suppose that a route has to cross the sector or airspace boundary. After entering the airspace, the route must stay within it for some period of time. The reason is that flying in/out of a sector involves a handoff between the controllers; i.e., crossing of the boundary involves a communication overhead. If the route exits the airspace immediately after the entrance, the communication spans a substantial portion of the flight within the airspace. Another requirement is that the paths, when entering the airspace, should make a large angle with the boundary edge of the airspace. Otherwise the path stays too close to the boundary for some time, and may accidentally deviate into the other airspace due to imprecise navigation.

In the current version of the program, this is not implemented; e.g., in **Figure 9**, the bottommost path crosses the boundary at a small angle, and then leaves the airspace almost immediately. This route may be discarded by the controller for not satisfying the crossing requirements.

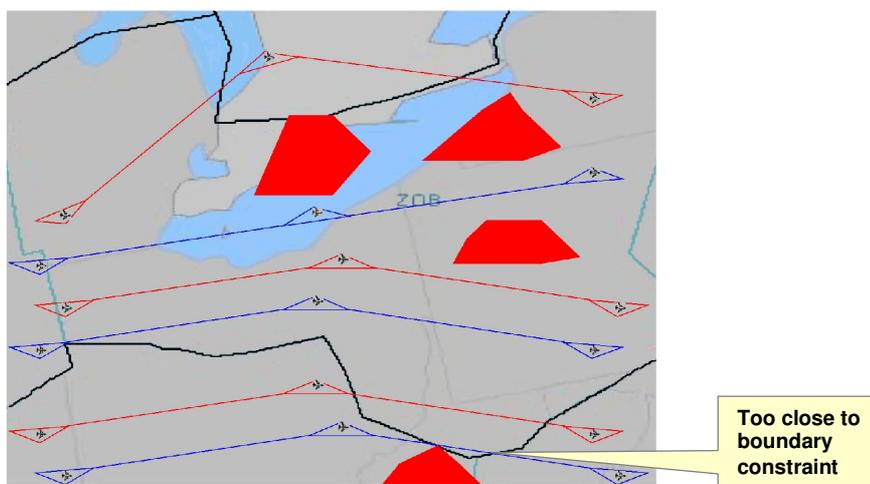
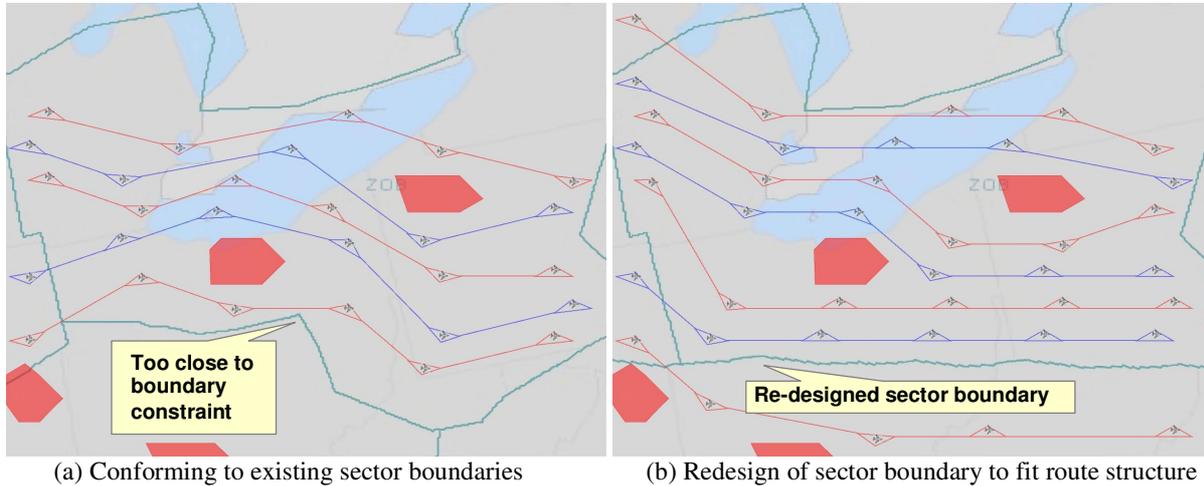


Figure 9. A sample output with the solution route clipping over the sector boundary constraint.

#### D. Combined Solution to Dynamic Route Planning and Dynamic Airspace Configuration (DAC)

Our algorithm designs routes that conform to the existing airspace boundaries; i.e., the boundaries are given as a part of the input to the problem. A different line of research, called Dynamic Airspace Configuration (DAC) management (see, e.g., [19], [20], [21], and references thereof), focuses on designing the airspace sector and center boundaries so that they conform well to existing routes and traffic flow patterns; there, the routes are the input. Undoubtedly, our route planner will benefit from intelligent design of the boundaries (as suggested by **Figure 10**).



**Figure 10. Comparison of routing with existing sector boundaries vs redesigning sectors to match routes.**

#### VII. Discussion

A number of directions for future research remain.

- In this paper, we considered planning paths in the en route airspace. At the same time, the highest congestion is found in the terminal area. Robust route planning for the terminal area, including planning of arrival trees (dynamically designing the Standard Arrival Routes) is a current research effort that we are pursuing.
- Although the third – vertical – dimension is not necessarily used for absorbing the delay [2], [3], [4], [5], going to full 3D is important for planning arrival trees in the terminal area. This is especially true for the case of *airports* – a concentration of several airports in a relatively small area.
- Another aspect, not addressed in this paper, is the dynamic nature of the problem. The weather systems are not static; their motion is often predicted by forecasts. Long term forecasts are also *probabilistic*, which should also be taken into account by the route planner.

#### VIII. Conclusion

We have studied the problem of designing routes with wiggle room uniformly distributed across an airspace with weather constraints. We designed an algorithm for planning routes with embedded path-stretch zones for providing wiggle room. The algorithm gives a theoretical guarantee of providing the number of routes, reaching the maximum capacity of the airspace. The number of path-stretch zones along a route is controlled by the user who can specify the minimum and maximum distance between the consecutive stretch locations. The user is also able to control the size of the zones.

#### IX. Acknowledgments

This research is partially supported by Academy of Finland grant 118653 (ALGODAN). This work is being performed under NASA's NextGen Airspace Super Dense Operations (ASDO) research focus area under contract NNA07BA84C, from NASA Ames Research Center. We appreciate the financial support of the sponsor of the research, NASA Ames Research Center and the NextGen Project Manager, Dr. Paramal Kopardekar. J. Mitchell is partially supported by the National Science Foundation (CCF-0528209, CCF-0729019).

## X. References

- [1] Coppenbarger, R., Lanier, R., Sweet, D., and Dorsky, S., "Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering", *AIAA Guidance, Navigation and Control Conf.*, Providence, RI, Aug., 2004.
- [2] Haraldsdottir, A., Scharl, J., Berge, M.E., Schoemig, E.G. and Coats, M.L., "Arrival Management with Required Navigation Performance and 3D Paths", *7th USA/Europe Air Traffic Management R&D Seminar*, Barcelona, Spain, July, 2007.
- [3] Schoemig, E.G., Armbruster, J., Boyle, D.A., Haraldsdottir, A., and Scharl, J., "3D Path Concept and Flight Management System (FMS) Trades", *25th Digital Avionics Systems Conf.*, Portland, OR, Oct., 2006.
- [4] Scharl, J., Berge, M.E., Haraldsdottir, A., and Schoemig, E.G., "A Near-Term Arrival Management Operational Concept and Preliminary Considerations", *25th Digital Avionics Systems Conf.*, Portland, OR, Oct., 2006.
- [5] Haraldsdottir, A., Berge, M.E., Kang, L.S., Schoemig, E.G. Alcabin, M.S., Repetto, B.W., and Carter, M.L., "Required Navigation Performance and 3D Paths in High-Traffic ATM Operations", *25th Digital Avionics Systems Conf.*, Portland, OR, Oct., 2006.
- [6] Brennan, M., "Airspace Flow Programs – A Fast Path to Deployment," *Journal of Air Traffic Control*, Vol. 49, No. 1, pp. 51-55., Spring, 2007.
- [7] Krozel, J., Jakobovits, R., and Penny, S., "An Algorithmic Approach for Airspace Flow Programs," *Air Traffic Control Quarterly*, Vol. 14, No. 3, 2006.
- [8] Mitchell, J.S.B., Polishchuk, V., and Krozel, J., "Airspace Throughput Analysis Considering Stochastic Weather", *AIAA Guidance, Navigation, and Control Conf.*, Keystone, CO, Aug., 2006.
- [9] Krozel, J., Mitchell, J.S.B., Polishchuk, V., and Prete, J., "Capacity Estimation for Airspaces with Convective Weather Constraints", *AIAA Guidance, Navigation, and Control Conf.*, Hilton Head, SC, Aug., 2007.
- [10] Krozel, J., Mitchell, J.S.B., Polishchuk, V., and Prete, J., "Maximum Flow Rates for Capacity Estimation in Level Flight with Convective Weather Constraints", *Air Traffic Control Quarterly*, 15(3):209-238, 2007.
- [11] Prete, J., *Aircraft Routing in the Presence of Hazardous Weather*, Ph.D. thesis, Stony Brook University, Stony Brook, NY, 2007.
- [12] Michalek, D., and Balakrishnan, H., "Identification of Robust Routes using Convective Weather Forecasts", *8th USA/Europe Air Traffic Management R&D Seminar*, Napa Valley, CA, June/July, 2009.
- [13] National Weather Service <http://www.nws.noaa.gov/> .
- [14] Rhoda, D.A., and Pawlak, M.L., "The Thunderstorm Penetration / Deviation Decision in the Terminal Area", *8th Conf. on Aviation, Range, and Aerospace Meteorology*, American Meteorological Society, Dallas, TX, pp. 308–312, 1999.
- [15] Rhoda, D.A., and Pawlak, M.L., *An Assessment of Thunderstorm Penetrations and Deviations by Commercial Aircraft in the Terminal Area*, MIT Lincoln Laboratory Tech. Report NAS/A-2, June 3, 1999.
- [16] DeLaura, R., and Evans, J., "An Exploratory Study of Modeling En Route Pilot Convective Storm Flight Deviation Behavior", *12th American Meteorological Society Conf. on Aviation, Range, and Aerospace Meteorology*, Atlanta, GA, Jan./Feb. 2006, Paper P12.6.
- [17] Martin, B., Evans, J., and DeLaura, R., "Exploration of a Model Relating Route Availability in En Route Airspace to Actual Weather Coverage Parameters", *12th Conf. on Aviation, Range, and Aerospace Meteorology*, Atlanta, GA, Jan./Feb. 2006, Paper P12.7.
- [18] Chan, W., Rafai, M., and DeLaura, R., "An Approach to Verify a Model for Translating Convective Weather Information to Air Traffic Management Impact," *AIAA Aviation Technology, Integration, and Operations Forum*, Belfast, Ireland, Sept., 2007.
- [19] Mitchell, J.S.B., Sabhnani, G., Krozel, J., Hoffman, R., and Yousefi, A., "Dynamic Airspace Configuration Management Based on Computational Geometry Techniques". *AIAA Guidance, Navigation, and Control Conf.*, Honolulu, HI, Aug., 2008.
- [20] Basu, A., Mitchell, J.S.B., and Sabhnani, G., "Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing", *Workshop on Algorithm Engineering and Experiments (ALENEX'08)*, San Francisco, CA, 2008.
- [21] G. Sabhnani, *Geometric Algorithms for Dynamic Airspace Sectorization*, Ph.D. Thesis, Stony Brook University, Stony Brook, NY, 2009.