Turn-Constrained Route Planning for Avoiding Hazardous Weather

Jimmy Krozel, Changkil Lee, and Joseph S.B. Mitchell

We investigate the problem of algorithmically synthesizing turn-constrained routes that minimize exposure to hazardous weather. Such a problem is suitable to air traffic management automation and aircraft flight management systems. Our algorithm synthesizes routes that aircraft may follow from a specified start to finish location. Example applications include synthesizing routes (1) from airport metering fix locations to runway final approach fixes, and (2) from sector boundary crossing locations to airport metering fix locations. The algorithm takes into account aircraft dynamics limits on velocity and acceleration, pilot and controller workload considerations (e.g., the number of turn maneuvers), and other constraints (e.g., utilizing arrival vs departure corridors or avoiding special use airspace regions). The solution approach is based on searching in an appropriate discretization of a geometric model of the airspace using a dynamic programming algorithm for optimal paths having a bounded number of turns. Examples are given for Dallas Ft. Worth Airport.

INTRODUCTION

We propose a new algorithm for route planning. The objective is to compute a route that minimizes its *weighted length* with respect to a given weight function, that specifies the cost per unit distance, subject to turn constraints. While our algorithm is quite general, applicable to many constrained

---

Jimmy Krozel is with Metron Aviation, Inc., Herndon, VA. Changkil Lee and Joseph S.B. Mitchell are with Stony Brook University, Stony Brook, NY.

Received April 1, 2006; accepted June 5, 2006.

Air Traffic Control Quarterly, Vol. 14(2) 159-182 (2006)
 CCC 1064-3818/95/030163-20

optimal routing problems, we are specifically motivated by applications in Air Traffic Management (ATM). We demonstrate the application of our algorithm to specific ATM scenarios involving synthesizing routes that avoid hazardous weather in the vicinity of the Dallas Ft. Worth Airport (DFW).

Air Traffic Management Problem Addressed

ATM in the airport terminal area works well under normal operating conditions; however, under inclement weather, throughput is adversely affected. The Aviation Capacity Enhancement Plan (FAA ACE Plan [Federal Aviation Administration, 2003]) lists weather as the leading cause (65% to 75%) of delays greater than 15 minutes, with traffic volume as the second leading cause (12% to 22% of delays). Thunderstorms account for approximately 50% of the weather-related delays, low visibility 35%, and heavy fog the remainder ([Evans and Ducot, 1994]). Weather-related delays are likely to get worse as the volume of air traffic increases in the future. Adding to this situation, emerging ATM automation, e.g., the Center TRACON Automation System (CTAS), does not currently incorporate hazardous weather data in the synthesis of trajectories nor in computing Estimated Times of Arrival (ETAs). Thus, CTAS is currently not used in severe weather conditions. Unfortunately, very high workloads for pilots and air traffic controllers result from these conditions. Mitigating the effects of weather-related delays as well as addressing concerns for pilot and controller workload are motivating factors for the work presented in this paper.

In general, our algorithm synthesizes a set of paths that aircraft may follow from a specified start to finish location. There are two specific applications that will serve to demonstrate the algorithm: (1) synthesizing routes from airport metering fix start locations ending at the destination locations defined by runway final approach fixes, and (2) synthesizing routes from sector boundary crossing locations ending at airport metering fix destination locations. The algorithm synthesizes weather avoidance routes while taking into account aircraft dynamics limits on velocity and acceleration, pilot and controller workload considerations (e.g., the number of turn maneuvers), and other ATM constraints (e.g., utilizing arrival vs departure corridors or avoiding Special Use Airspace (SUA) regions). In this paper, some of the examples illustrate the first application, while other examples demonstrate the second application.

As with most optimal path algorithms based on dynamic programming, the algorithm computes not just a single optimal route, but an Optimal Path Map (OPM), allowing optimal routes from any fixed source (or to any fixed destination) to be traced easily after a single execution of the search algorithm.

Air Traffic Management Applications

One application of this algorithm is to automatically synthesizing weather avoidance routes by computer automation for aircraft approaching an airport. Such a problem is suitable to future ground-based ATM automation systems, e.g., CTAS. In this application, routes would be determined ahead of time for traffic arriving into an airport, and controllers would be advised of the safest routes with minimal number of turns, accepting or rejecting them based on conflicts from constraints not modeled in the problem statement. The OPM allows for controllers to view and assign many routes leading to a metering fix from sector boundary crossing location entry points. For aircraft that are properly equipped, routes can be data linked to aircraft for input into their Flight Management Systems (FMSs), or routes can be described verbally by the controller (for unequipped aircraft). In either situation, minimizing the total number of turns reduces the communications bandwidth, and thus reduces workload.

Another application of our algorithm is to route planning functions within FMSs of the future. The algorithm parameters introduced in this paper (a minimum length between way points and a maximum turn angle) will allow for FMSs to plan for weather avoidance routes while abiding by turning radius constraints of an aircraft, and therefore, solutions provided by our algorithm should be compatible with FMS requirements. In the future, weather avoidance routes that are synthesized within the FMS and accepted by the pilot may be broadcast to other aircraft and to ATM through a data link system, for instance, with Automatic Dependent Surveillance – Broadcast (ADS-B) and Trajectory Change Points (TCPs). In such an operation, both pilot workload and data link bandwidth limitations benefit from solutions with fewer numbers of turns. Furthermore, if ADS-B is limited to broadcasts of up to four TCPs at any given time (as specified in [RTCA, 2002]), the predictability of where aircraft will go in the future is greatly limited if weather avoidance solutions require more than four turn points – a property that cannot be controlled by many weather avoidance algorithms (see Related Work).

For researchers studying the long term modernization of the NAS, our algorithm helps to identify tradeoffs amoung routing efficiency, weather avoidance strategies that are designed to guarantee safety from hazardous weather, workload, and requirements for ground-based vs airborne applications.

Additional Applications

Our optimal routing algorithm is quite general; it can be applied to compute turn-constrained optimal routes for a wide variety of constraints, in a variety

of application areas. Specifically, the routing algorithm is applicable in geographic route planning systems and in robotics for autonomous ground vehicles and military Unmanned Air Vehicles (UAVs)

Organization of Paper

This paper is organized as follows. Section 2 describes the model of the problem. Section 3 focuses on weather data processing. Section 4 presents the main routing algorithm, and Section 5 gives experimental results of applying the algorithm to weather avoidance routes. Finally, Section 6 presents our conclusions.

PROBLEM FORMULATION

Flight Paths

Routes describe flight paths that aircraft may be directed to follow. We consider routes to be polygonal (piecewise-linear) paths. We let *s* denote the starting location (the origin, or source), and we let *t* denote the destination location. Associated with *s* is a cone C_s of initial headings, and associated with *t* is a cone C_t of final headings; these cones specify boundary conditions on how the path exits point *s* and how it enters point *t*, according to headings that lie within the respective cones.

A flight path π from *s* to *t* is specified by a sequence, $(s = v_0, v_1, \ldots, v_k = t)$, of *way points* (turn points) v_i. The segment $v_i v_{i+1}$ between two consecutive way points is a *link* (also referred to as a *leg* for FMSs) of the path.

We impose constraints on the geometry of the planned paths, specified by the following parameters: *K*, $\theta_{\max}$, and $L_{\min}$. Parameters $\theta_{\max}$ and $L_{\min}$ are discussed below, as they are related to aircraft dynamics. The parameter *K* is the maximum number of links (legs) in a planned path; i.e., the path is allowed to have at most *K-1* way points. In our ATM application, *K* is used

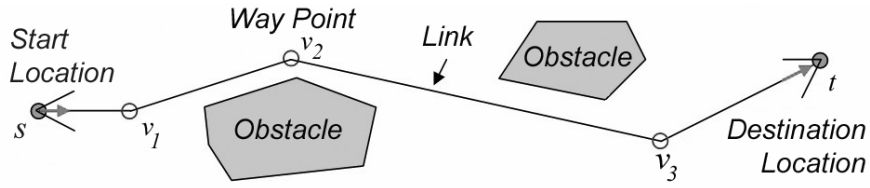


Figure 1. A polygonal flight path π avoiding constraints (obstacles) with 4 links, intermediate way points v_1, v_2, and v_3, a cone C_s of initial headings at the start location *s*, and a cone C_t of final headings at the destination location *t*.

to bound the pilot and ATM workload; in robotics applications, K may be used to bound the number of times a mobile robot executes a turn maneuver, which costs time and energy.

Turning Constraints Imposed by Aircraft Dynamics

The dynamics of an aircraft limit its turning radius. While a polygonal path technically has zero radius of curvature at its turn points, we are using polygonal paths only as an approximation of the exact routes. Thus, in our model we employ two parameters, θ_{max} and L_{min}, which determine discrete curvature constraints on a polygonal path.

The parameter θ_{max} is the maximum allowed heading (angle) change at a turn. The parameter L_{min} is the minimum length of a link (leg), i.e., the minimum distance between two consecutive turns. The two parameters (θ_{max}, L_{min}) provide a means of bounding the curvature of a trajectory that is approximated by a polygonal path; the minimum radius of curvature for a circular arc that passes through three consecutive way points of the polygonal path is $L_{min}/2\sin\theta_{max}$.

Acute turning angles in the flight path are prohibited in the terminal area and are not synthesized by the weather avoidance algorithm. For turning radius constraints, we consider a coordinated turn model to determine the turn limitations for different types of aircraft. The maximum bank angle is used to investigate the limitations on heading maneuvers, and the standard 3° per second turn rate is used to determine the nominal turn conditions. The roll response for commanded bank angles has fairly quick dynamics for most aircraft, so transient dynamics and time delays in performing turns are not considered.

The aircraft's minimum airspeed and maximum bank angle determine its turning radius limit. According to the Federal Aviation Regulations (FARs), an aircraft must comply with specific airspeed limits when entering certain classes of airspace. Most major airports are in Class B airspace which has a ceiling of 8,000 ft MSL (Mean Sea Level). Other airports may be in Class C airspace, which has 4,000 ft AGL ceilings. Altitude ranges vary for these classes of airspace, and within each range, the airspeed limits can vary.

Practical bank angle ranges are needed to calculate turning radius constraints. For reasons such as safety and passenger comfort, commercial pilots do not typically exceed a 30° bank angle. Even though the smaller types of aircraft bank angles might exceed 30°, it is not practical to surpass 45° bank angles for the purpose of designing flight paths. In addition, air traffic controllers do not typically issue heading changes that would require more than a 30° bank. Thus, by setting the bank angle limits to 30° for

commercial aircraft and 45° for general aviation, the turning radius constraints for the algorithm are established.

Next, we consider turning radii. Since our demonstration experiments involve vectoring aircraft through the terminal areas of major airports, such as Dallas-Ft. Worth (DFW), Chicago (ORD), or Atlanta (ATL), Class B airspace is assumed. A turbine powered aircraft performing a turn at 30° bank will achieve a minimum turning radius of approximately 0.7 nmi at 170 kn, and a maximum radius of 1.0 nmi at a speed of 200 kn. These numbers represent typical limits for commercial aircraft. In addition, a smaller general aviation aircraft banking at 45° will have a minimum turning radius of 0.3 nmi when traveling at 150 kn, and a maximum radius of 0.6 nmi at a speed of 200 kn.

Objective Function

Our objective is to determine routes that obey the path turn constraints while minimizing an objective function, which is determined by map data. In our motivating application to ATM route synthesis to avoid hazardous weather, the map data is specified by weather data that has been observed or predicted.

Let *M* denote a *map* that specifies the environment within which we are to determine optimal routes. Typically, the map is defined over a rectangular region and is considered to be two-dimensional; however, our algorithm generalizes readily to three-dimensional environments, at a moderate computational cost. In general, the map may consist of several layers of information, with each map layer specifying in some form a set of constraints or cost information. Our algorithm is able to handle map layers that are specified as weighted polygonal regions or as weighted regular grids (which can be thought of as a subdivision into very small squares). A region having infinite weight indicates an obstacle: the cost of passing through it is infinity. The *weight function*, *w(p)*, indicates the cost per unit distance of travel at point $p \in M$. If there are multiple layers of data making up the map, then *w(p)* is a composite function of the weights at point *p* in each layer of data.

Our objective is to compute a polygonal path π from *s* to *t* that obeys the cones of initial/final headings, has at most *K* links, obeys the turn constraints specified by (θ_{max}, L_{min}), and minimizes the objective function, the *path cost* $c(\pi)$, given by the path integral

$$c(\pi) = \int_{\pi} w(\sigma) d\sigma \tag{1}$$

where $d\sigma$ is the differential arc length at a point $\sigma \in \pi$. Since $\pi = (s = v_0, v_1, ..., v_k = t)$ is a polygonal path, with turn points v_i (which are to be determined), the path integral becomes a discrete summation,

$$c(\pi) = \sum_{i=1}^{k} c(v_{i-1}v_i) \tag{2}$$

where $c(v_{i-1}v_i)$ is the (straight) line integral of the cost function $w(\sigma)$ along the segment $v_{i-1}v_i$. While our focus here is on minimizing path cost subject to a bound on the number of links, our algorithm readily permits us to include in $c(\pi)$ a turn cost term, allowing us to have a cost associated with each turn, with the cost being a function of the turn angle. (This modification simply adds a turn cost term to the Bellman equations described below, without changing the algorithmic complexity of the method.) This option may be of utility in some ATM applications and is of interest in robotics applications in which turn angles directly impact the cost of a route.

When the weight function *w* is given by a piecewise-constant set of weights on the faces of a polygonal subdivision, the cost $c(v_{i-1}v_i)$ of a link is computed by determining the points where segment $v_{i-1}v_i$ crosses face boundaries of the subdivision, and summing the lengths of the resulting subsegments, each multiplied by the weights of the corresponding faces.

When the weight function *w* is given by a regular grid, as is often the case for weather data, the cost $c(v_{i-1}v_i)$ of a link is computed by approximating the path integral by performing a *scan conversion* of the segment $v_{i-1}v_i$. This determines the set of pixels (squares in the grid) crossed by $v_{i-1}v_i$; we then compute the average weight value of the crossed pixels, and multiply this average by the Euclidean length of the segment. Optionally, the cost function specification includes a threshold value, and the cost of the segment is infinity if the segment crosses any pixel having weight equal to or larger than the threshold. This allows us to prevent flight links (legs) that cross any portion of hazardous weather or SUA. Alternatively, we can compute the cost of a flight path by computing the exact lengths of all links (legs) lying in each pixel and then summing the correspondingly weighted lengths. (This approach is similar to the *unweighted area sampling* algorithm for anti-aliasing line scan-conversion, and the somewhat better technique of *weighted area sampling*; see, e.g., [Foley et al., 1997].) This amounts to computing an exact path integral of a piecewise-constant weight function on the regular grid subdivision induced by weighted pixels.

WEATHER PROCESSING

In our ATM application, the map *M* and the corresponding weight function *w* are provided as two-dimensional or three-dimensional weather data.

Typically, the weather data is given as a regular grid, W_G, of intensity values, e.g., indicating the amount of precipitation at each point of the grid. Data in the form of the ASR-9 grid are an example of two-dimensional data; the Nexrad 3D grid is an example of a three-dimensional data. The spatial resolutions of these two grids are similar. These data are typically registered according to the National Weather Service (NWS) standard reflectivity levels (7 weather levels) or measurements of radar reflectivity (0 – 60 *dBZ*). Vertically Integrated Liquid (VIL) data, which is a two-dimensional grid form of data, are the integral of the reflectivity reading over a vertical column of airspace. A three dimensional grid allows a route planning algorithm to vector aircraft both above and around storms. (Note, however, vectoring aircraft beneath storms is problematic in the terminal airspace because storms can quickly drop in altitude.) With a two-dimensional grid, the route planning algorithm has no knowledge of a storm's altitude extent, so it cannot exploit clear air that might be available above or below the storm. However, VIL data does integrate three dimensional weather data in a two-dimensional grid, so the third dimension is technically being accounted for in a worst case sense.

Weather Measurement and Classification

For our ATM application, the weather avoidance algorithm is nominally being designed to integrate with the Integrated Terminal Weather System (ITWS). More specifically, the VIL weather data of ITWS is input. The weather data grids W_G used in our experiments are typically two-dimensional grids with grid spacing of about 0.5 to 1 mile. Raw radar reflectivity data (in *dBZ*) or NWS data (Levels 0-6) can be passed through a filter which groups the data into bins with set weights, w_i, $i = 1,2,\ldots,6$ as described in Table 1. Note that in Table 1 there are bins and weights for all NWS Levels 0-6, but in the application, not all weights may be different. In particular, the weights for hazardous weather NWS Levels 4, 5, and 6 may all be set to the same very large weight so that all these hazardous regions are all equally avoided. The value for each weight w_i is determined through

Table 1. NWS Standard Reflectivity Levels, Weather Classification, and Weights

NWS Level	Color	Rainfall Rate x (mm/h)	Reflectivity y (dBZ)	Type	Weight
0	None	$x < 0.49$	$y < 18$	None	$w_0 = 1$
1	Light green	$0.49 \leq x < 2.7$	$18 \leq y < 30$	Light Mist	w_1
2	Dark green	$2.7 \leq x < 13.3$	$30 \leq y < 41$	Moderate	w_2
3	Yellow	$13.3 \leq x < 27.3$	$41 \leq y < 46$	Hazardous	w_3
4	Orange	$27.3 \leq x < 48.6$	$46 \leq y < 50$	Very Hazardous	w_4
5	Deep orange	$48.6 \leq x < 133.2$	$50 \leq y < 57$	Intense	w_5
6	Red	$133.2 \leq x$	$57 \leq y$	Extreme	w_6

engineering analysis so that the resultant variable routes look acceptable to pilots and air traffic controllers. Alternatively, these weights can be determined by minimizing an error (in ETA or spatially for solution paths) between our algorithm results and actual recorded flight data from adverse weather scenarios.

We perform some simple filtering on input weather data in order to remove some noise and small clusters of green weather cells (NWS Level 1 and 2). We convolve the data with a Gaussian smoothing kernel and threshold the result at a low value; this results in some small clusters of green weather cells being removed. Because the yellow, orange, and red weather cells (NWS Levels 3, 4, 5 and 6) are also affected by this filter, we retain the yellow, orange, and red cells of the original weather data, since we want our routing algorithm to avoid all such cells.

Hazardous Weather Uncertainty

From an MIT Lincoln Laboratory study [Rhoda and Pawlak, 1999], the general rule-of-thumb that pilots use for directing safe passage through weather is to keep aircraft from having to fly through precipitation that is NWS Level 3 (41 *dBZ*) or higher; on the radar scope, this means to avoid yellow, orange, and red cells. Their research, based on actual weather and aircraft track data, indicates that pilots will penetrate NWS Level 1 and 2 weather but will take on deviations for NWS Level 3 or higher; the pilot surveys of [Weidner et al., 1998] confirm this preference.

Terminal weather forecasts will generally be available every 6-12 minutes. Modifications to the weather data will depend on the quality of forecast. We make the simplifying assumption in this paper that the weather data is *fixed* over the planning horizon. Since there are about 4 weather forecast updates over the course of the planning horizon for the distances over which we ran our experiments, this is not an unreasonable approximation for our first analysis.

Because of the potential hazards they pose, and due to the uncertainty in weather measurement and prediction, when applying our routing algorithm to hazardous weather avoidance, we require synthesized routes to stay at least a certain distance from weather cells of NWS Level 3 or higher. We do this by processing the weather data with a simple expansion (wildfire) algorithm to add safety margins to yellow and red weather data in the forecast so that planned paths do not fly too close to these regions. We expand by a specified amount $\delta_3>0$, in 4, 8, or 16 neighboring directions, depending on user preference. Yellow and red color components are first identified using the NWS Levels 3-6. The NWS Level 3 data are grown by δ_3 in all directions, where δ_3 is determined by the measurement uncertainty

(a function of the forecast prediction accuracy) and desired safety factors. NWS Level 4-6 data are grown by δ_4 in all directions, where $\delta_3 \leq \delta_4$, depending again on measurement uncertainty and desired safety factors (refer to Figure 2).

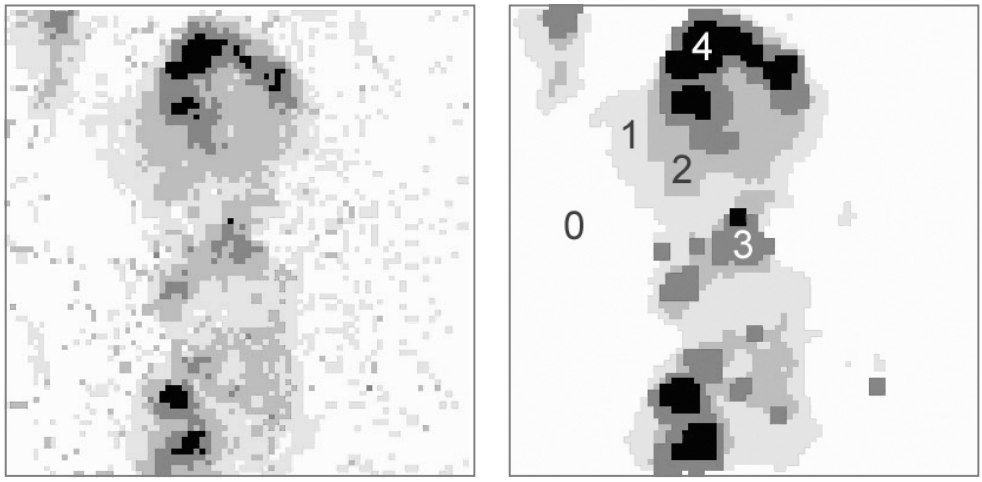
The weather processing part of the algorithm separates colors, simplifies the complexity, adds safety margins to hazardous weather (yellow, orange, red cells), and sums up the parts to form a new weather map layer. Finally, the weights w_i, as shown in Table 1 are assigned to each of these colors in the weather map layer. Further information on the weather processing algorithm is given in [Krozel et al., 1999].

OPTIMAL CONSTRAINED ROUTING ALGORITHM

The Main Algorithm

Our main routing algorithm is based on searching an appropriate discrete graph for shortest *k*-link paths using dynamic programming based on a variant of the Bellman-Ford shortest path algorithm and a post-processing step to locally optimize routes. See, e.g., [Cormen et al., 2001] or [Lawler, 1976] for background on dynamic programming, Bellman equations, the Bellman-Ford algorithm, and shortest paths in networks.

We let V denote a set of nodes $v \in V$ corresponding to regular (rectangular) grid points, at (x_v, y_v), that cover the region of the map M; the grid resolution is user-specified, but most often in our experiments the default is that the grid resolution is about half that of the weather grid data. We assume that s and t are in the set V; if they are not grid points, we augment V to include them.



(a) Input weather data image. (b) Output weather data image.

Figure 2: Weather data processing (NWS Level labeled, 8-neighbor expansion; $\delta_3 = 1$, $\delta_4 = 1$).

The nodes *V* represent a discretization of locations within the region of the map *M*. In addition to discretizing locations, we also discretize headings; in particular, we partition the set, $[0,2\pi)$, of all headings into *H* equal-size intervals of headings ("cones"), each of angle $2\pi/H$, and we let *C*(*h*) denote the cone corresponding to headings $[2\pi(h-1)/H, 2\pi h/H)$, for heading index $h \in \{1,2,...,H\}$. Associated with the special node *s* (resp., *t*) is the cone C_s (resp., C_t) of possible initial (resp., final) headings; the cones C_s and C_t represent boundary conditions on the route from *s* to *t*. We consider *state* $\Sigma = (v,h,k)$, for $v \in V$, $h \in \{1,2,...,H\}$, and $k \in \{1,2,...,k\}$, to be the situation that the aircraft is at location *v*, with a heading in cone *C*(*h*), having reached this situation by flying along a polygonal route from *s* to *v* that has at most *k* links. We denote the *state space*, $S = V \times \{1,\ldots,H\} \times \{1,\ldots,K\}$.

The discretization of locations and headings also leads to a natural discretization of the set of "actions" (transitions from one state to the next) in the optimal control problem. Specifically, for a given grid node *v* and heading index *h*, suppose that (*v*,*h*,*k*) is optimally reachable from *predecessor* $p(v,h,k) = (u,h',k-1)$; thus, the directed segment *uv* lies within the heading cone *C*(*h*), and *uv* represents the last link in an optimal *k*-link path from *s* to *v* that arrives at *v* with a heading in cone *C*(*h*). The set of possible actions associated with (*v*,*h*,*k*) is discretized according to turn angle at *v* and the length of the next link, *vw*, in an optimal (*k*+1)-link path. Since the route arrives at *v* from node *u*, heading in direction *uv*, and there is a maximum turn angle of $\theta_{\max}$, we discretize the turn angle at *v* to be in the set $\{-\theta_{\max} + j \cdot \Delta\theta : j = 1,2,...,m\}$, where $\Delta\theta = 2\theta_{\max}/(m+1)$, and *m* is a user-specified (odd) integer parameter. (Note that *j* = (*m*-1)/2 corresponds to no turn at *v*.) Since each link of the route is to be of length at least L_{min}, we also discretize the next link length to be in the set $\{L_{\min} + i \cdot \Delta L : i = 1,2,...,l\}$, where ΔL and *l* are user-specified parameters. The result of this polar coordinate discretization at *v* is a set of points that serve as candidates for the next turn after *v* along a route. In order to keep the set of candidates on the discrete grid *V*, we round each of them to the grid. Refer to Figure 3, where the candidates are shown as hollow dots, and the corresponding grid boxes to which points are rounded (to the center of a grid box) are shown shaded. We refer to the corresponding subset of (rounded) grid nodes $w \in V$ as the *action grid*, *A*(*v*,*h*), associated with the pair (*v*,*h*).

We let *f*(Σ) = *f*(*v*,*h*,*k*) denote the weight of a minimum-weight feasible path, π^* (*v*,*h*,*k*), from *s* to *v*, with initial heading in the cone C_s, with the link incident to *v* having heading in cone *C*(*h*), with the route having at most *k* links, and with each link defined by choices from the corresponding action grid.

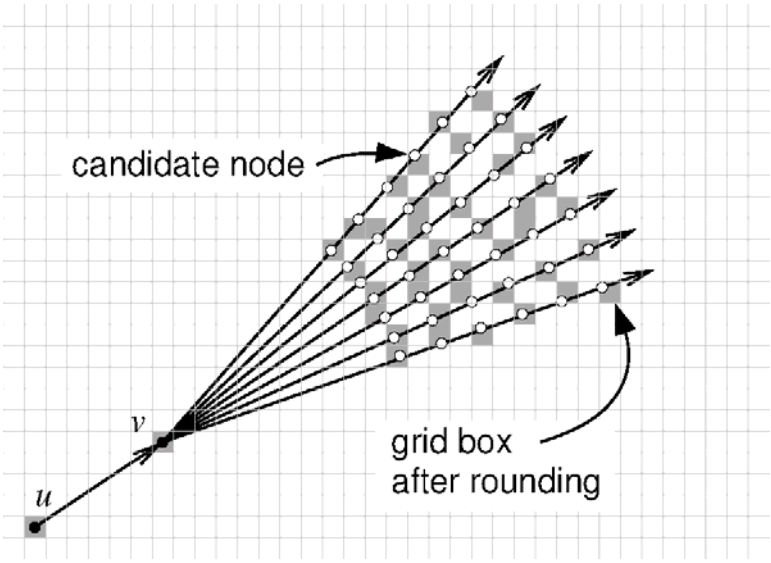


Figure 3. The action grid (m=5, l=6) over which the algorithm searches for the best link to follow node v in an optimal route.

We let $p(\Sigma) = p(v,h,k)$ denote the *predecessor* of $\Sigma = (v,h,k)$ in path $\pi^*(v,h,k)$. Thus, we can write $\pi^*(v,h,k) = (\Sigma_0, \Sigma_1, ..., \Sigma_k)$, where $\Sigma_k = (v,h,k)$, $\Sigma_0 = (s,h_0,0)$, and for each $i = 0,1,...,k-1$, $\Sigma_i = (v_i,h_i,i) = p(\Sigma_{i+1})$, with directed line segment $v_{i-1}v_i$ having a heading within the cone $C(h_i)$.

The algorithm has K stages, corresponding to link counts k = 0,1,2,…,K. Initially (k=*0*), we have $f(s,h,0)$ = 0 for each h in the allowed initial headings (i.e., those h for which $C(h) \cap C_s \neq \varnothing$); for other choices of h, $f(s,h,0) = +\infty$. For all other nodes $v \in V$, $v \neq s$, we set $f(v,h,0) = +\infty$, for all h.

At stage k, there is a subset $R(k) \subset V$, the *k-reachable set*, of grid nodes that have been reached so far with a path of at most k links, at some heading; initially, $R(0) = \{s\}$. For each $v \in R(k)$, and for each heading h for which $f(v,h,k) < \infty$, we determine the corresponding action grid $A(v,h)$, and for each $w \in A(v,h)$ we update the corresponding weight function values and predecessor pointers. There are two cases, depending on whether or not w corresponds to making a turn at v. Assume first that w does correspond to a turn at v (i.e., $j \neq (m-1)/2$). Then, let h' be the heading index corresponding to the directed segment vw, and let $c(v,w)$ be the cost of the link vw determined by the weight function (e.g., the weather intensity map). If $f(v,h,k) + c(v,w) < f(w,h',k+1)$, we update the value $f(w,h',k+1) \leftarrow f(v,h,k) + c(v,w)$, and set the predecessor pointer

$p(w,h',k+1) \leftarrow (v,h,k)$; otherwise, we leave $f(w,h',k+1)$ and $p(w,h',k+1)$ unchanged. In the case that *w* corresponds to there being no turn at *v* (i.e., $j = (m\text{-}1)/2$ and $h' = h$), only a lengthening of the segment *uv*, we do a similar update, but do not count *v* as a turn point: If $f(v,h,k) + c(v,w) < f(w,h,k)$, we update the value $f(w,h,k) \leftarrow f(v,h,k) + c(v,w)$, and set the predecessor pointer $p(w,h,k) \leftarrow (u,h,k)$, where *u* is the predecessor of *v*; otherwise, we leave $f(w,h,k)$ and $p(w,h,k)$ unchanged. At the conclusion of this algorithm, the optimal weights satisfy the dynamic programming recursion (Bellman equations):

$$f(w,h',k+1) = \min_{(v,h):(w,h')\in A(v,h)} \{c(v,w) + f(v,h,k)\}. \quad (3)$$

Figure 4 illustrates the application of the algorithm to weather avoidance, showing that we obtain a set of routes, one for each $k = 0,1,2,\ldots, K$, so that a suite of options can be presented to the user.

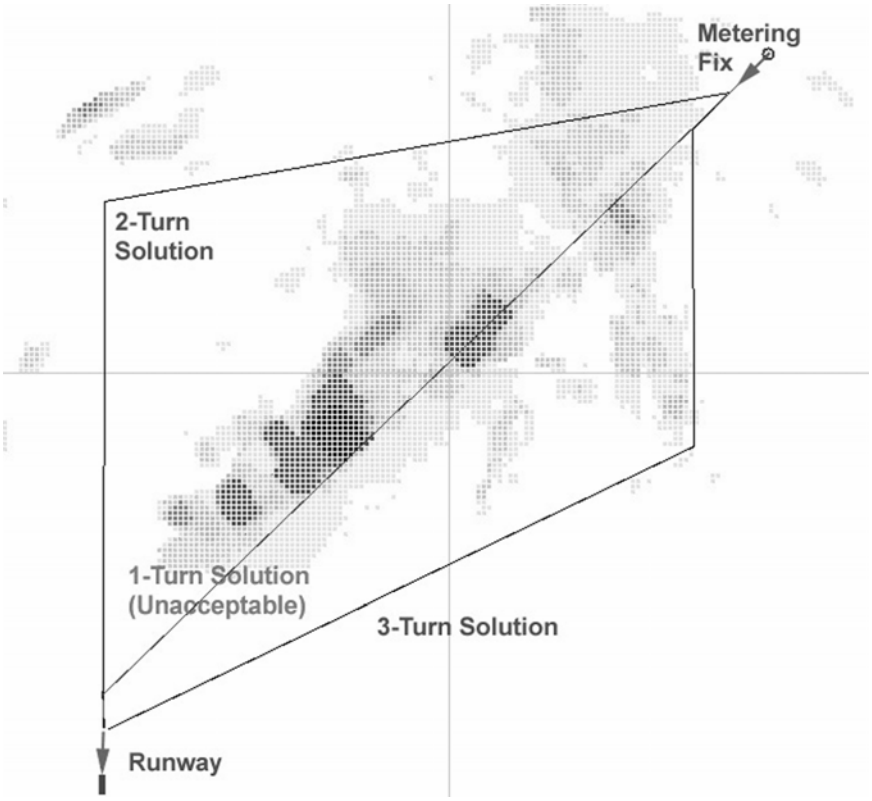
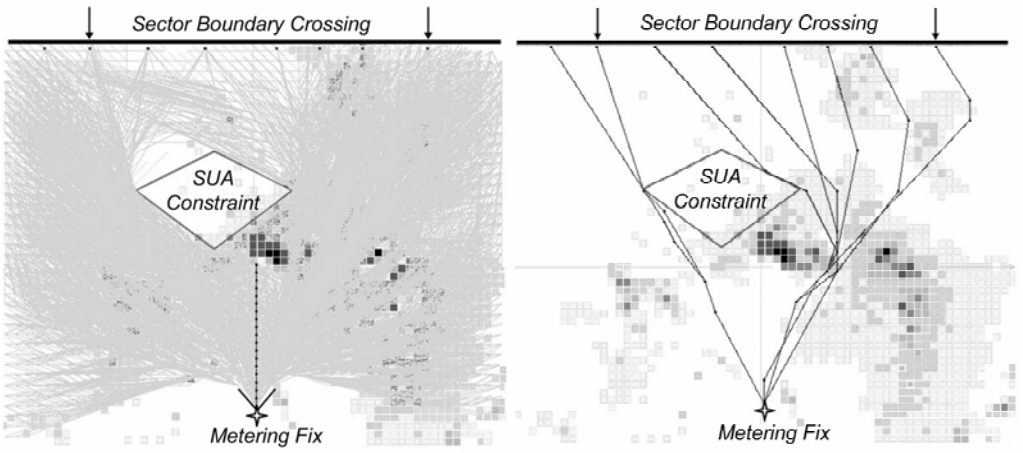


Figure 4. The shortest 1-turn (unacceptable), constrained 2-turn, and constrained 3-turn paths.

The result of the algorithm is a tabulation of the optimal function values $f(v,h,k)$ and the predecessor pointers $p(v,h,k)$, for each grid node v and each choice of h and of k. The pointers $p(v,h,k)$ define an *optimal path map* (OPM) over the domain, allowing one to trace an optimal path to any state (v,h,k) by following the pointers back to the source, s. This feature of the algorithm is illustrated in Figure 5.

The asymptotic running time of the algorithm is $O(kE)$, where E is the number of links connecting pairs of states. Since there are H $|V|$ nodes, each with degree at most ml (the number of elements in the action grid), we see that $E \le H\,|V|\,ml$. Since we usually consider the discretization parameters H, m, and l to be small constants, the dependence on k and $|V|$ is $O(k|V|)$. In practical terms, the algorithm is quite efficient; the time required to compute an optimal route is about 0.1-1 s on a low-end personal computer, depending on some of the user parameter choices.

One means of accelerating the algorithm that we employ is to do a form of pruning of the search space, as follows. Consider a state $\Sigma = (v,h,k)$, with corresponding last link given by the directed segment uv. If uv does not lie within the cone C_t of final headings, and if the smallest angle between uv and C_t is greater than $(K-k)\theta_{\max}$, then we can prune the state (v,h,k), not following any neighboring states reachable from it, since it is impossible to reach the destination t with a heading in C_t, even if every remaining turn is at the maximum possible angle, $\theta_{\max}$.



(a) Search Space. (b) Optimal Path Map (OPM).

Figure 5: Search space used to generate the OPM with respect to a source s at a metering fix of an airport. In this example, in addition to costs induced by the weather intensity map, a SUA constraint with infinite cost is imposed.

Post-processing

The OPM allows one to trace an optimal *k*-link path from s to any grid node v, with any discrete heading index h. While this route is optimal among the class of routes defined by V and the discretization of the headings and the action grid, it is not necessarily an exact global optimal route in the continuum of possible routes; as the discretization becomes arbitrarily fine in the limit, it approaches the exact optimum. (A numerical analysis of the rate of convergence is beyond the scope of this paper.)

In order to optimize further the output of our discrete search, we have added a post-processing step that locally optimizes the grid-based route, π_G, bringing it closer to a global optimum, π^*. The post-processing step makes a (small) specified number of passes along the route π_G, locally optimizing each turn point in succession along the route. The local optimization of a turn point v is based on considering perturbations of v to points equally spaced on a small circle centered at v, selecting the best such perturbation in terms of how much the weighted path length is decreased. (In our experiments, we place 36 evenly distributed points about the circle.) If no improvement is found, the radius of the perturbation is divided by 2, and it is done again, continuing some small specified number of times, until the radius of perturbation is negligibly small. Experimentally, we have found that this post-processing step allows for noticeable improvements in the quality of computed routes, allowing us to use a courser search grid V while not sacrificing the quality of solutions.

Related Work

Optimal path algorithms in computational geometry have been studied extensively; see, e.g., the surveys [Mitchell, 2000] and [Mitchell, 2004].

[Mitchell and Papadimitriou, 1991] introduced the algorithmic study of the *weighted region problem* (WRP) of computing minimum-cost paths in weighted polygonal subdivisions. They gave the first (1+ε)-approximation algorithm to compute optimal paths that runs in polynomial time, polynomial in *n* (the number of vertices of the subdivision) and logarithmic in $1/\varepsilon$ (where $\varepsilon > 0$ is any given small number). Their algorithm exploits the local optimality property that shortest paths bend according to Snell's law of refraction; this is also exploited in the algorithms of [Rowe and Richbourg, 1990] and [Richbourg et al., 1987]. Recent alternative solution methods for the WRP have been devised based on discretizing links of the subdivision, placing "Steiner points" judiciously, and interconnecting them to form a discretization graph, *G*(*V*,*E*), which can then be searched for a minimum-cost path. Implementations of these algorithms were the basis of the first

experimental studies of the WRP; see, e.g., [Lanthier et al., 1997] and [Mata and Mitchell, 1997]. Subsequent theoretical investigations of the WRP have focused on careful placement of Steiner points either on edges ([Aleksandrov et al., 1998], [Aleksandrov et al., 2000], [Reif and Sun, 2000], [Sun and Reif, 2003]) or on face bisectors [Aleksandrov et al., 2005], with additional techniques to speed the computation of shortest paths in the resulting discretization graph ([Reif and Sun, 2000], [Sun and Reif, 2003], [Aleksandrov et al., 2005]). These WRP algorithms do not, however, address the turn constraints that we impose in our algorithm: The routes they compute may have a very large number of turns and may include arbitrarily sharp turn angles and very short links.

Algorithms to approximate paths using at most k links in weighted subdivisions, with guaranteed approximation bounds, are given in [Daescu et al., 2005]. They propose two different algorithms for approximating k-link shortest paths based on the computation of *exact* or *approximate* solutions to the 1-link problem [Chen et al., 2001], [Chen et al., 2003,Daescu, 2002] combined with a search of a discrete graph interconnecting Steiner points placed on edges of the subdivision (as in [Lanthier et al., 2001], [Aleksandrov et al., 2005], [Reif and Sun, 2000], [Sun and Reif, 2003], with important modifications in the vicinity of vertices). Their two solutions produce approximation paths whose weight is guaranteed to be within a $(1+\varepsilon)$-factor from the optimal weight of a k-link solution, while having at most $5k-2$ links or $14k$ links, respectively. Their algorithms do not bound discrete curvature of paths.

The problem of computing shortest paths having at most k links in *unweighted* settings (e.g., within simple polygons or among a set of obstacles) has been studied by [Piatko, 1993], [Arkin et al., 1991], and [Mitchell et al., 1992]. While their methods allow an explicit bound on the number of turns, they do not address discrete curvature constraints and do not apply in general cost settings.

EXPERIMENTS

We conducted a series of experiments applying our algorithm to compute aircraft routes in the vicinity of DFW airport. We have experimented with several scenarios, using different choices of starting and destination locations (s and t), various weather data, and various choices of parameters in the optimization. We have demonstrated the use of the algorithm in computing routes linking metering fixes to entry points of short and longside arrival patterns, using the heading constraint cones (C_s and C_t) to constrain the routes to align with arrival/departure legs and to conform with crossing constraints at fixes.

In the experiments, we used weights $w_0=1$, $w_1=3$, $w_2=7$, $w_3=13$, $w_4=21$, and $w_5=w_6=\infty$ (infinity). These were determined to be good choices based

on inspection of many routes. The weights are a user-specified parameter and are easily modified. In order to illustrate the effects of changes in the selection of weights assigned to different NWS levels, we show the results of four different choices of weights in Figure 6.

For the results shown here, which summarize typical experimental runs, we used two different sets of weather data, as shown in Figures 7, 8; this data shows storm cells in the immediate vicinity of the nominal paths from the metering fix to the runway. In addition, we illustrate examples in which constraints are imposed on arrivals to avoid the departure airspace. Figures 7 and 8 present the results of computing a set of trajectories corresponding to optimal routes of the OPM that connect points at or near the arrival fix to the arrival pattern, in line with the runway.

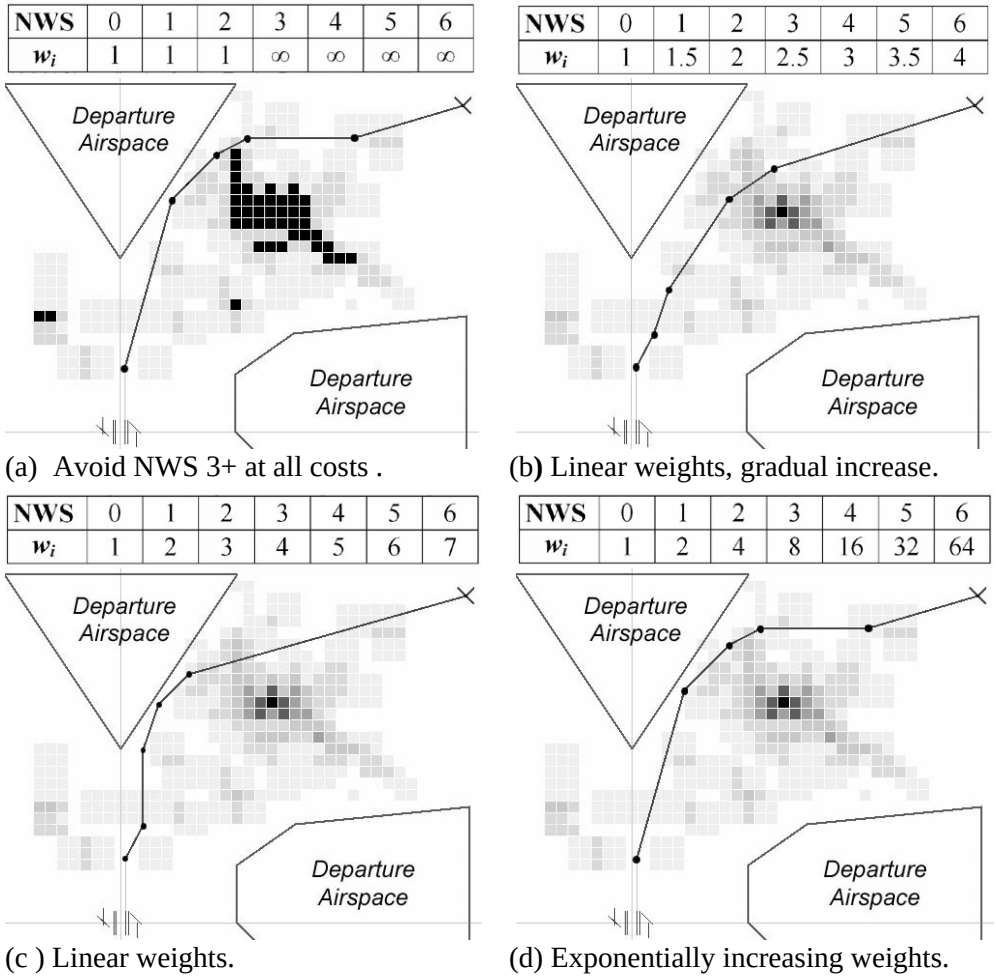


Figure 6. Sensitivity of the solution to different choices of weights assigned to NWS levels (K=5 in all cases).

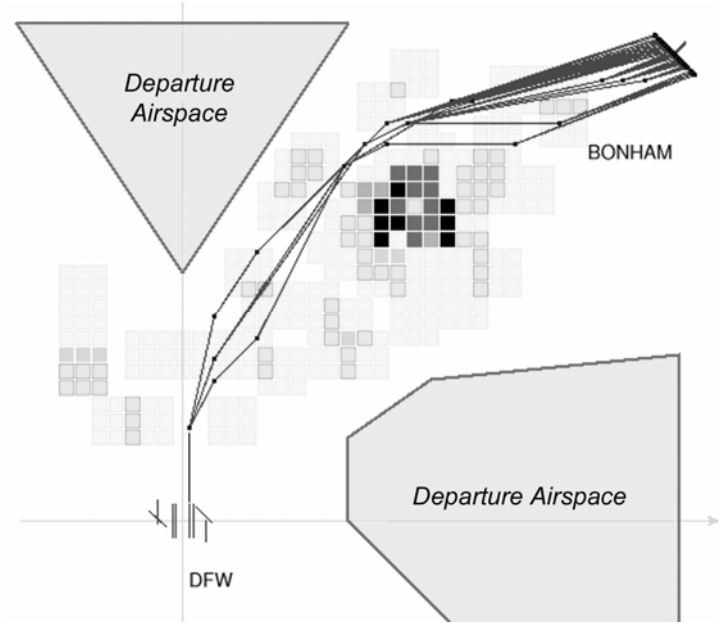


Figure 7. First weather dataset (DFW).

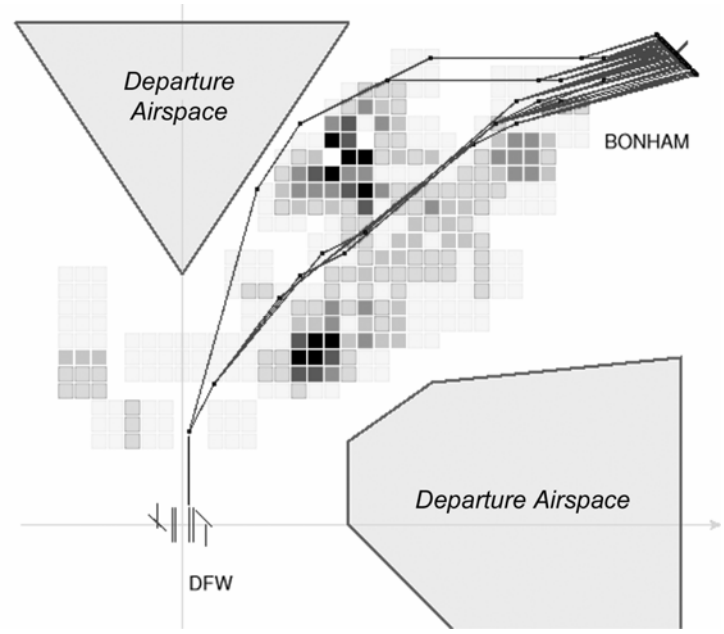


Figure 8. Second weather dataset (DFW).

In these experiments, we generated 100 random location/heading pairs along the metering fix (Bonham). (The positions and headings were generated independently of each other, using a truncated Gaussian.) The metering fix is modeled as a segment of length 3.7 kilometers, and the turn angle is constrained to be at most 10° (rather than 30°) between the heading at the point of crossing the metering fix and the next link (leg) in the flight path.

In our experiments, we used a maximum turn angle of $\theta_{max} = 30°$ and a choice of $L_{min} = 5$ kilometers. The nodes V are taken from a 65 x 65 grid, and the headings are discretized into cones of 15°, with $H = 24$. We used $m = l = 5$, so that the turns are selected among the choices of going straight, turning by $\pm 15°$ or turning by $\pm 30°$, and the action grid had 25 possibilities.

The results of these experiments are shown in Figures 9, 10. For all runs, the algorithm was able to find a feasible path having at most K links for all 100 generated pairs of location/headings at the metering fix. The mean (μ), standard deviation (σ), and minimum and maximum values, over the set of paths were computed. As expected, the larger the value of K, the lower the path cost. Note, however, that the path length may go up as K increases, since the additional links (legs) are used to decrease the weather-induced path *cost* (which is the objective function), rather than to reduce the path length. Note too that there tends to be more variation in the path cost than there is in the path lengths.

CONCLUSIONS

This paper investigates the problem of computing optimal turn-constrained routes in the presence of weather hazards. We propose a general purpose constrained routing algorithm for turn-constrained routes in the presence of weather-induced costs. The algorithm computes a set of constrained optimal paths (an Optimal Path Map). We illustrate the application of the algorithm to computing weather avoidance routes in the vicinity of an airport, where than be many constraints on the routes, both from the geometry of the airspace and from the weather hazards. The routing algorithm is general enough to model practical constraints including aircraft dynamics limits (turn angle constraints), pilot and controller workload considerations (number of turns), and constraints from arrival vs departure corridors. While our algorithm has been described for the case of two-dimensional route planning, without taking into account altitude, the method extends directly to three dimensions, at a corresponding computational cost in increasing the size of the discrete search space.

FUTURE WORK

In future work, the following issues need to be addressed in order that our algorithm be fully applicable in practice:

- The algorithm needs to be modified for planning routes in the presence of a stochastic weather model, with predicted storm tracks having an appropriate probability distribution.

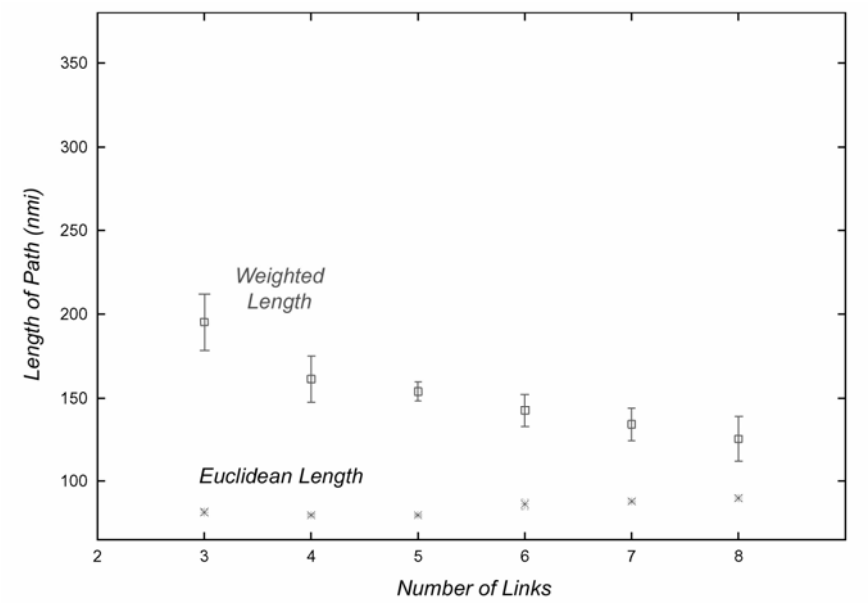


Figure 9. Weather dataset 1: Means and standard deviations of Euclidean length and of weighted length, as a function of the allowed number *K* of links.

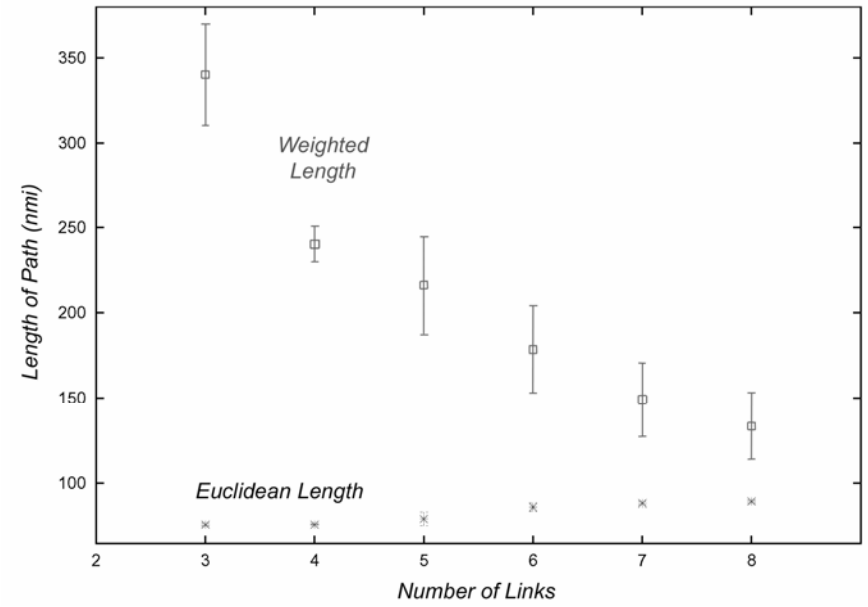


Figure 10. Weather dataset 2: Means and standard deviations of Euclidean length and of weighted length, as a function of the allowed number *K* of links.

- The implementation needs to be modified to take into account the full 4-D space-time nature of the problem, including the the horizontal, altitude, and time dimensions. While our algorithm readily applies to the full 4-D space-time domain, the current implementation and visualization is in two spatial dimensions. Further investigation is needed also to address altitude profile choices.

- Future research is needed to determine how to meet a required time of arrival at a metering fix (or other location).
- Weather data distribution needs to be taken into consideration for an ATM system solution. Aircraft arriving at a merge point from different directions will get different radar views of the same storm, and this will be different from the ground-based view of the weather, for instance, from ITWS. In the future, weather data (and weather forecast information) may need to be data linked to the cockpit in order to create a common situational awareness of the weather problem between pilots and controllers.
- Voice and data link communications requirements need to be researched to identify a feasible solution to implementing this algorithmic approach to either airborne applications or ground-based ATM applications of the future. Initially, controllers may need to provide complex clearances using voice communications, but eventually, with a greater percentage of the fleet equipped with data link technology, a solution may progress towards a fully automated system.

ACKNOWLEDGMENTS

This research was funded by NASA Ames Research Center under the following contracts NAS2-97055, NAS2-98023, and NAS2-02075. J. Mitchell has been partially supported by grants from the National Science Foundation (CCR-0098172, ACI-0328930, CCF-0431030, CCF-0528209), the U.S.-Israel Binational Science Foundation (grant No. 2000160), Metron Aviation, and NASA Ames (NAG2-1620).

LIST OF ACRONYMS AND SYMBOLS

ADS-B	Automatic Dependent Surveillance – Broadcast
AGL	Above Ground Level
ATL	Atlanta Hartsfield International Airport
ATM	Air Traffic Management
CTAS	Center TRACON Automation System
DFW	Dallas Ft. Worth International Airport
ETA	Estimated Time of Arrival
FAR	Federal Aviation Regulation
FMS	Flight Management System
ITWS	Integrated Terminal Weather System
MSL	Mean Sea Level
NWS	National Weather Service
OPM	Optimal Path Map
ORD	Chicago O'Hare International Airport

SUA	Special Use Airspace
TCP	Trajectory Change Point
VIL	Vertically Integrated Liquid
WRP	Weighted Region Problem
$A(v,h)$	action grid for node v, heading h
$c(v,w)$	cost of link vw
$C(h)$	heading cone for heading h
C_s, C_t	cones of initial, final headings
δ	expansion amount in weather processing
$f(\Sigma) = f(v,h,k)$	weight of a minimum-weight path to state (v,h,k)
h	index of a heading interval
H	number of intervals in discretization of headings
k	number of links in a path so far
K	maximum number of links in a planned path
l	integer parameter specifying number of different link length options
$L_{\min}$	minimum length of a path link
m	integer (odd) parameter specifying number of different turn options
M	map of the region of interest
$p(v,h,k)$	predecessor of state (v,h,k)
$\pi^*(v,h,k)$	optimal path from start location to state (v,h,k)
s	start location
S	state space given by $V \times \{1,...,H\} \times \{1,...,K\}$
$\Sigma = (v,h,k)$	a state, corresponding to node v, heading interval h, and k links
t	destination location
θ_{max}	maximum turn angle
v	a node of the search graph, at location (x_v, y_v)
V	set of all nodes in the search graph
$w(p)$	weight function at a point $p \in M$
W_G	weather grid of intensity values

BIBLIOGRAPHY

Aleksandrov, L., Lanthier, M., Maheshwari, A., and Sack, J.-R. (1998), "An ε-Approximation Algorithm for Weighted Shortest Paths on Polyhedral Surfaces," *Proceedings 6th Scandinavian Workshop Algorithm Theory*, Vol. 1432 of *Lecture Notes on Computer Science*, pp. 11-22, Springer-Verlag, Stockholm.

Aleksandrov, L., Maheshwari, A., and Sack, J.-R. (2000), "Approximation Algorithms for geometric Geometric Shortest Path Problems," *Proceedings 32nd Annual ACM Symposium Theory Computing*, pp. 286-295, Portland, OR, May.

Aleksandrov, L., Maheshwari, A., and Sack, J.-R. (2005), "Determining Approximate Shortest Paths on Weighted Polyhedral Surfaces," *Journal of the ACM*, 52 (1), pp. 25-53.

Arkin, E. M., Mitchell, J. S. B., and Piatko, C. D. (1991), "Bicriteria Shortest Path Problems in the Plane," *Proceedings 3rd Canadian Conference Computational Geometry*, pp. 153-156, Vancouver, Canada, August.

Chen, D., Daescu, O., Hu, X., Wu, X., and Xu, J. (2001), "Determining an Optimal Penetration among Weighted Regions in Two and Three Dimensions," *Journal of*

Combinatorial Optimization, 5 (1), pp. 59-79.

Chen, D. Z., Hu, X., and Xu, J. (2003), "Optimal Beam Penetration in Two and Three Dimensions," *Journal of Combinatorial Optimization*, 7 (2), pp. 111-136.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001), *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Daescu, O. (2002), "Improved Optimal Weighted Links Algorithms," *Proceedings of ICCS, 2nd International Workshop on Computational Geometry and Applications*, pp. 65-74, Amsterdam, April.

Daescu, O., Mitchell, J. S. B., Ntafos, S., Palmer, J. D., and Yap, C. K. (2005), "*k*-link Shortest Paths in Weighted Subdivisions," *Proceedings 9th Workshop Algorithms Data Structures*, Vol. 3608 of *Lecture Notes on Computer Science*, pp. 325-337. Springer-Verlag, Waterloo, Canada, August.

Evans, J. E. and Ducot, E. R. (1994), "The Integrated Terminal Weather System (ITWS)," *The Lincoln Laboratory Journal*, 7 (2), pp. 449-474.

Federal Aviation Administration (2003), 2003 Aviation Capacity Enhancement Plan, Washingtion, DC.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1997), *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Co., Reading, MA.

Krozel, J., Lee, C., and Mitchell, J.S.B. (1999), "Estimating Time of Arrival in Heavy Weather Conditions," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, August.

Lanthier, M., Maheshwari, A., and Sack, J.-R. (1997), "Approximating Weighted Shortest Paths on Polyhedral Surfaces," *Proceedings 13th Annual ACM Symposium Computational Geometry*, pp. 274-283, Nice, France, June.

Lanthier, M., Maheshwari, A., and Sack, J.-R. (2001), "Approximating Shortest Paths on Weighted Polyhedral Surfaces," *Algorithmica*, 30 (4), pp. 527-562.

Lawler, E. L. (1976), *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York, NY.

Mata, C. and Mitchell, J. S. B. (1997), "A New Algorithm for Computing Shortest Paths in Weighted Planar Subdivisions," *Proceedings 13th Annual ACM Symposium Computational Geometry*, pp. 264-273, Nice, France, June.

Mitchell, J. S. B. (2000), "Geometric Shortest Paths and Network Optimization," In Sack, J.-R. and Urrutia, J., editors, *Handbook of Computational Geometry*, pp. 633-701, Elsevier Science Publishers B.V. North-Holland, Amsterdam.

Mitchell, J. S. B. (2004), "Shortest paths and Networks," In Goodman, J. E. and O'Rourke, J., editors, *Handbook of Discrete and Computational Geometry (2nd Edition)*, chapter 27, pp. 607-641. Chapman and Hall/CRC, Boca Raton, FL.

Mitchell, J. S. B. and Papadimitriou, C. H. (1991), "The Weighted Region Problem: Finding Shortest Paths through a Weighted Planar Subdivision," *Journal of the ACM*, 38, pp. 18-73.

Mitchell, J. S. B., Piatko, C., and Arkin, E. M. (1992), "Computing a Shortest *k*-link Path in a Polygon," *Proceedings 33rd Annual IEEE Symposium on Foundations Computer Science*, pp. 573-582, Pittsburgh, PA, October.

Piatko, C. D. (1993), *Geometric Bicriteria Optimal Path Problems*. Ph.D. Thesis, Cornell University, Ithaca, NY.

Reif, J. and Sun, Z. (2000), "An Efficient Approximation Algorithm for Weighted Region Shortest Path Problem," *Proceedings of the 4th Workshop on Algorithmic Foundations of Robotics*, pp. 16-18, Hanover, NH, March.

Rhoda, D. A. and Pawlak, M. L. (1999), "The Thunderstorm Penetration/Deviation Decision in the Terminal Area," *American Meteorological Society's 8th Conference on Aviation, Range, and Aerospace Meteorology*, pp. 308-312, Dallas, TX, January.

Richbourg, R. F., Rowe, N. C., Zyda, M. J., and McGhee, R. (1987), "Solving Global Two-dimensional Routing Problems using Snell's Law," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1631-1636, Raleigh, NC, March/April.

Rowe, N. C. and Richbourg, R. F. (1990), "An Efficient Snell's Law Method for Optimal-Path Planning across Multiple Two-dimensional, Irregular, Homogeneous-cost Regions," *International Journal of Robotics Research*, 9 (6), pp. 48-66.

RTCA Special Committee-186 (2002), *Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)*, RTCA/DO-242A, June.

Sun, Z. and Reif, J. H. (2003), "Adaptive and Compact Discretization for Weighted Region Optimal Path Finding," *Proceedings 14th Symposium on Fundamentals of Computation Theory*, Lecture Notes in Computer Science. Springer-Verlag, Malmö, Sweden.

Weidner, T., Hunter, C. G., and Guadiamos, C. (1998), *Severe Weather Avoidance Pilot Survey*, Technical Report TR-98148-4-01, Seagull Technology, Inc., Los Gatos, CA.

BIOGRAPHIES

Jimmy Krozel is a Senior Engineer in the Research and Analysis Department at Metron Aviation. Jimmy Krozel received an AS (1984, Computer Science), BS (1985, Aeronautical Engineering), MS (1988, Aeronautical Engineering), and Ph.D. (1992, Aeronautical Engineering) from Purdue University. Krozel was a Howard Hughes Doctoral Fellow (1987-1992) while at the Hughes Research Labs (1987-1992). Krozel is an Associate Fellow of the AIAA, has over 50 technical publications, and is the winner of two AIAA best paper awards. His research interests include computational geometry, computer graphics, visualization, air traffic management, air traffic control, intelligent path prediction, intent inference, and autonomous vehicles.

Changkil Lee conducted this research while a Ph.D. student in the Department of Applied Mathematics and Statistics at Stony Brook University. Dr. Lee's research is in the areas of algorithms, computational geometry, and optimization. He now works in the area of finance.

Joseph S. B. Mitchell received a BS (1981, Physics and Applied Mathematics), and an MS (1981, Mathematics) from Carnegie-Mellon University, and Ph.D. (1986, Operations Research) from Stanford University. Mitchell was with Hughes Research Labs (1981-86) and then on the faculty of Cornell University (1986-1991). He now serves as Professor of Applied Mathematics and Statistics and Research Professor of Computer Science at Stony Brook University. His primary research area is computational geometry, applied to problems in air traffic management, sensor networks, computer graphics, visualization, manufacturing, and geographic information systems.