Flexible Airlane Generation to Maximize Flow Under Hard and Soft Constraints

Shang Yang, Joseph S. B. Mitchell,

Stony Brook University, Stony Brook, NY 11794

Jimmy Krozel,

Metron Aviation, 45300 Catalina Court, Dulles, VA 20166

Valentin Polishchuk,

Helsinki Institute for Information Technology, Helsinki, FI-00014

Joondong Kim, and Jingyu Zou

Stony Brook University, Stony Brook, NY 11794

ABSTRACT

We consider a multicriteria optimization problem of simultaneously routing several classes of aircraft through an airspace at a fixed flight level in the presence of various types of constraints. Hard constraints are formed by hazards through which no aircraft can safely fly (e.g., severe convection, turbulence, or icing). Soft constraints are formed by hazards through which some pilots or airlines decide to fly while others do not (e.g., moderate turbulence or icing). We compute flight paths for two aircraft classes: Class-1 aircraft avoid hard constraints but are willing to fly through soft constraints, and Class-2 aircraft avoid both hard and soft constraints. Our work assists in the design of future operational concepts in which jetway routing is retired and aircraft paths are allowed to adjust to the shapes and positions of constraints. We are interested in determining the capacity of an airspace and feasible routes across an airspace with hard and soft constraints, given as input the demand profile indicating how many Class-1 and Class-2 aircraft are scheduled to enter the airspace. We report on experiments both with real and with synthesized weather data.

INTRODUCTION

A fundamental problem in Air Traffic Management (ATM) is estimation of the capacity of an airspace. The capacity measures the capability of the airspace to accommodate a predicted traffic demand through it, with the demand specified as the number of aircraft, of each of a set of aircraft *classes*, that intend to use the airspace during a given time interval. Capacity is impacted by constraints that come from various sources, including forecasted hazardous weather and planned Special Use Airspace (SUA) constraints. Such constraints may arise from typical, daily weather conditions, e.g., convective weather or turbulence, as well as from less frequent conditions such as in-flight icing, volcanic ash in the atmosphere, or other phenomena (see the survey of [Krozel and Murphy, 2007]). The capacity is the number of aircraft of each class that can be routed through the airspace while avoiding all constraints.

When the demand for an airspace exceeds its capacity, a Traffic Flow Management (TFM) strategy, such as an Airspace Flow Program (AFP) [Krozel et al, 2006; Brennan, 2007], may be required to adjust the demand to remain at or below the estimated capacity. In current ATM operations, a Flow Constrained Area (FCA) is established in en route airspace, such that the flow of aircraft crossing into the FCA is controlled by the AFP. Today, aircraft arrive at the FCA using standard jet routes, and thus the capacity of the FCA is dependent on the number of jet routes that remain open (do not have route blockage from hazardous weather) as a function of time. In the future, it is possible that the routes crossing the FCA can be synthesized unconstrained by jetway routing, assuming that Area Navigation (RNAV) and Required Navigation Performance (RNP) will allow for flexible routing structures. Routes crossing the FCA can be computed simultaneously as the AFP is planned, so that the number of routes and location of the routes crossing the FCA can be optimized subject to the demand on the FCA and the forecasted locations and shapes of hazardous weather constraints. Furthermore, the AFP can be designed to position aircraft into the appropriate airlane upstream of the FCA constraint region, regulating the crossing time into the FCA using AFP planning and control strategies [Krozel et al, 2006; Brennan, 2007]. In this paper, an approach to flexible airlane generation to maximize flow across such a constrained airspace is described. In general, the airspace could be an FCA, sector, center, or a grid cell describing an airspace partitioning.

We study the problem of computing flow rates for capacity estimation of an airspace at a fixed flight level in the presence of two types of constraints: *hard* and *soft*. Hard constraints are portions of airspace through which no aircraft can fly safely; these include SUA as well as severe hazardous weather phenomena (convection, turbulence, icing, or volcanic ash). Soft constraints are portions of airspace through which some aircraft classes can safely fly, while other aircraft classes may either need to avoid or choose to avoid (e.g., due to pilot or air carrier preferences). Pilots and air carriers determine the class of airlane in which they are willing to participate; thus, some pilots (air carriers) may choose to participate in an airlane that passes through a soft constraint while other pilots (air carriers) may choose to avoid such a constraint and participate in another class of airlane. Soft constraints may arise, for instance, from certain weather hazards, such as moderate convection, turbulence, or icing. In this paper, we concentrate on the case of *two* aircraft classes and *two* types of constraints (hard and soft); nevertheless, most of our results extend directly to multiple classes of aircraft and multiple types of constraints. (In general, the problem can be analyzed in terms of a *weather impact interaction grid*, which specifies which types of constraints must be avoided by which classes of aircraft; see [Lindholm, et al, 2009].) We assume the aircraft fall into two classes: Class-1 aircraft avoid hard constraints but are willing to fly through soft constraints, while Class-2 aircraft avoid both hard and soft constraints.

In contrast with the majority of the models used in the previous research on capacity estimation, our problem formulation recognizes the need to model weather hazards in terms of hard and soft constraints and (at least) two classes of aircraft. To the best of our knowledge, we are the first to present an algorithmic technique for capacity estimation addressing several types of constraints for multiple classes of aircraft.

Our previous work [Krozel et al., 2007] showed that the airspace capacity depends on the ATM control laws being implemented. These laws may represent decentralized techniques, such as Free Flight [RTCA, 1995], or centralized controls, such as flow-based routing [Prete and Mitchell, 2004; Prete, 2007; Krozel et al., 2007]. In this paper, we investigate centralized flow-based routing for two classes of flows. Our model assumes that there is a horizontal separation requirement between centerlines of flows, specified by given air traffic lane width, w. In addition, aircraft flying along one route are separated from each other by a Miles-in-Trail (MIT) requirement. The width w is quite general; it may depend on the class of aircraft and will, in general, be dependent on the RNP of the aircraft utilizing an airlane. For current-day applications, lateral separation requirements and MIT requirements of 7, 10, 15, or 20 nmi may be appropriate; for future operations, smaller separation requirements (e.g., 5 nmi or 3 nmi) may be utilized.

The research in this paper is in support of both the Next Generation Air Transportation System (NextGen) and the Single European Sky ATM Research (SESAR) operational concepts. The presented solution is not dependent on existing jet routes or ATM practices; our model assumes that the flight paths can be designed to pass through the airspace wherever constraints allow for feasible Class-1 or Class-2 traffic flows. Our study may help NextGen policy decision makers to evaluate variations of service-by-equipage operational concepts and to determine the extent to which convection, turbulence, icing, as well as SUA may limit en route capacity in NextGen. The results establish theoretical upper bounds on capacity imposed by hard and soft constraints at a given flight level for two (or more) classes of aircraft. However, the paper does not address the more general three-dimensional (3D) routing problem with aircraft that are climbing or descending to avoid hard or soft constraints; we leave this problem for future research.

Our model is intended to support the design of new roles for controllers and pilots in NextGen and SESAR. With this in mind, we address capacity estimation in terms of the limitations on traffic utilization based on the geometry of the airspace and the constraints within it; we do not directly address here the limitations on traffic utilization imposed by workload considerations of the people that monitor the airspace. In particular, we are not addressing maximum aircraft count per sector (e.g., Monitor Alert Parameter (MAP) values).

Related Work

In [Krozel et al., 2007], we reported on capacity estimation techniques for airspaces with convective weather constraints (of a single type); related prior research surveyed in that paper includes [Schmidt, 1975; Song et al., 2006; DeLaura and Evans, 2006; Chan et al., 2007]. An experimental comparison of techniques for estimating the sector capacity given convective weather constraints in today's ATM system is presented in [Song et al., 2008].

In NextGen we expect that jet routes can be dynamically redefined to adjust flows of traffic around weather constraints, and that controller workload will not be a limiting factor. In such a setting, the maximum capacity of an airspace can be assessed using extensions of maximum flow theory in networks (see, e.g., [Ahuja et al., 1993]) to maximum flows in geometric domains. The standard network MaxFlow/MinCut Theorem extends to geometric domains [Strang, 1983]; its algorithmic properties have been studied originally in [Mitchell, 1990] and more recently in [Mitchell and Polishchuk, 2007; Arkin et al., 2010]. The MaxFlow/MinCut theory has been applied in ATM capacity estimation tasks, for determining the maximum throughput across an en route airspace with hard constraints given either by a traffic flow pattern [Song et al., 2008], or by a uniform distribution of flow monotonically traversing in a standard direction (e.g., East-to-West), or random, Free Flight conditions [Krozel et al., 2007]. The maximum capacity of terminal airspace may also be determined by transforming the problem into an effectively two-dimensional (2D) domain on the ascent or descent cone modeling the transition airspace [Krozel et al., 2008].

We know of no prior algorithmic results directly related to the multi-class geometric flow routing problem studied in this paper. At the same time, there is an abundance of work on computing multiple paths and flows (of a single class) in geometric domains. In [van der Berg and Overmars, 2005], a classification of existing approaches to multiple paths planning is suggested. In particular, in the prioritized planning, the paths are found iteratively, one by one, treating already routed paths as obstacles for each newly added path. This is the same strategy as employed by [Chiang et al, 1997; Prete and Mitchell, 2004; Prete, 2007], who developed routing algorithms for multiple aircraft based on iterative packing in space-time. Apart from the prioritized routing, other routing schemes include centralized path planning, roadmap-based methods, and decoupled path planning [van der Berg and Overmars, 2005]. In [Arkin et al., 2010], a pseudopolynomial-time dual-approximation algorithm was given to determine a maximum number of trajectories for velocity-bounded agents (disks) moving in a polygonal domain in which there are moving (polygonal) constraints. In ATM terms, the diameter of the disks corresponds to lane width or the horizontal separation standard.

Paper Organization

We first discuss the modeling of the problem. We then review the theory of capacity estimation. Next, we present our main contribution – algorithmic methods for capacity estimation in presence of hard and soft constraints, and applications of our methods to icing and turbulence constraints. We conclude with a discussion of our results and future work.

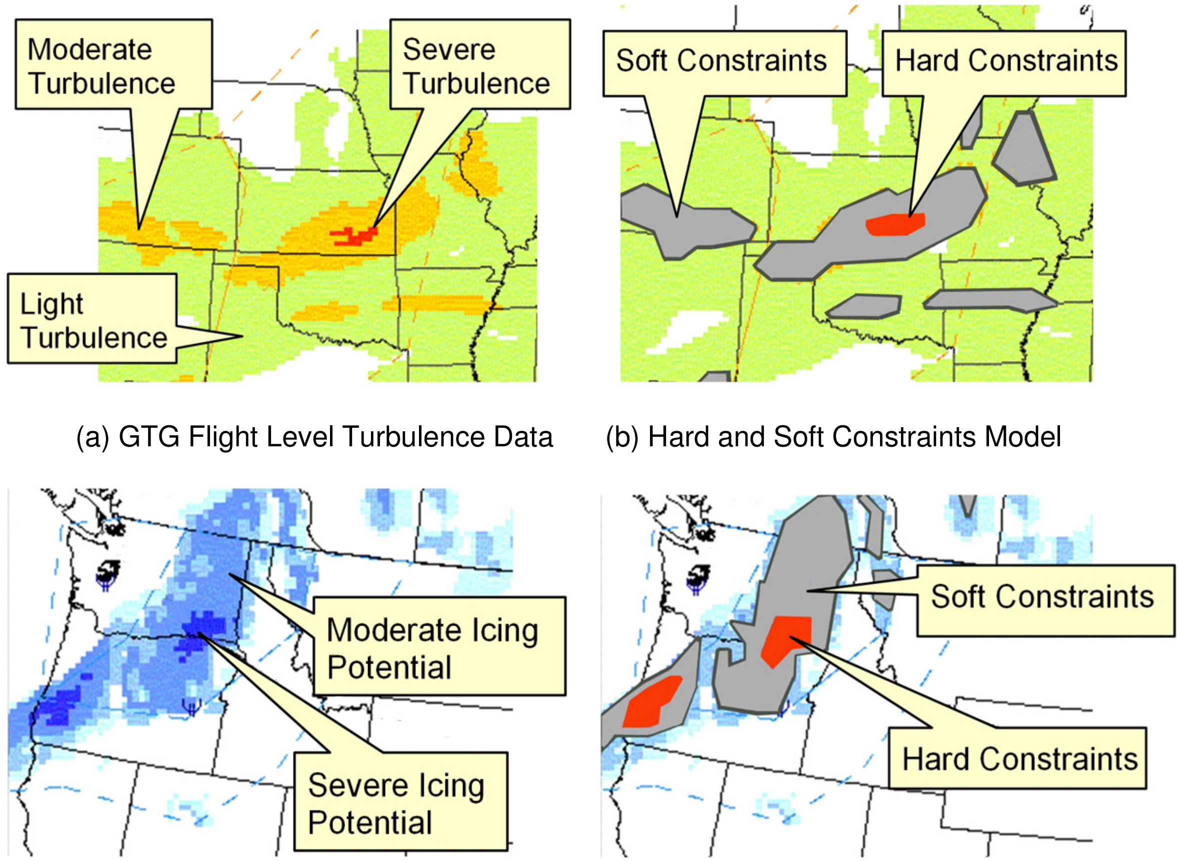
MODELING

To motivate our model, we present two examples of weather impacts on the National Airspace System (NAS) – turbulence and icing.

The first example is turbulence. [Krozel et al., 2011] studied the rules and regulations associated with turbulence, and have found that there are two significant levels of turbulence that determine hard and soft constraints. Moderate-or-Greater (MoG) turbulence tends to limit the capacity of en route airspace since passenger comfort and safety is a high priority for many airlines; thus, many airlines choose to avoid MoG turbulence. Still, some aircraft and airlines do fly through MoG turbulence; for instance, ferry flights, cargo flights, and some business jets. However, if Severe-or-Greater (SoG) turbulence is forecast or reported, it poses an immediate safety hazard, which closes airspace and, if encountered, may require diversion due to injuries and/or required aircraft inspections. SoG turbulence can cause aircraft structural damage/failure, loss of control, and injury or death to passengers. Thus, SoG turbulence areas of the NAS are not safe for flight, and therefore represent hard constraints.

The second example is in-flight icing. [Krishna and Krozel, 2009] studied the rules and regulations as well as pilot/aircraft/airline responses (e.g., cancellations, en route holding, altitude deviations, and diversions) related to significant in-flight icing events that result in SIGMETs (Significant Meteorological Information). SIGMET airspace regions severely restrict the flow of traffic through a region of airspace described by the horizontal polygon boundaries and the lower and upper altitude limits. The SIGMET generally describes a hard constraint region due to the severity of the icing potential within it, which is, generally, a SoG icing level. For MoG icing levels, aircraft and airlines may enter the airspace; however, the decision generally depends on the aircraft and equipment. If aircraft are equipped to address icing conditions, then they may flight through MoG icing. However, other aircraft, e.g., General Aviation (GA) aircraft, may not be equipped to fly through icing, in which case even MoG icing regions should be avoided. In-flight icing is thus determined by MoG or SoG icing severity levels and the pilot, airline policy or aircraft capability.

Figure 1 illustrates the modeling of turbulence and icing constraints. **Figure 1(a,b)** shows the conversion of a turbulence forecast (Graphical Turbulence Guidance (GTG)) into hard and soft constraints. **Figure 1(c,d)** illustrates the conversion of an in-flight icing forecast (Current Icing Potential (CIP)) into hard and soft constraints.



(a) GTG Flight Level Turbulence Data (b) Hard and Soft Constraints Model

(c) CIP Flight Level Icing Data (d) Hard and Soft Constraints Model

Figure 1. Mapping forecasts into hard and soft constraints.

Airspace Model

The traffic flow at a fixed flight level is modeled as a 2D problem. The airspace is modeled by a polygonal domain *P*, representing a NAS sector, center, or Flow Evaluation Area (FEA). The traffic enters *P* through a specified *source* edge, and exits through a *sink* edge. Envisioning West-to-East traffic, assume that the source and the sink are the “left” and the ‘right” edges of *P* (**Figure 2**). Furthermore, we do not consider flows of aircraft originating or terminating within airspace *P*.

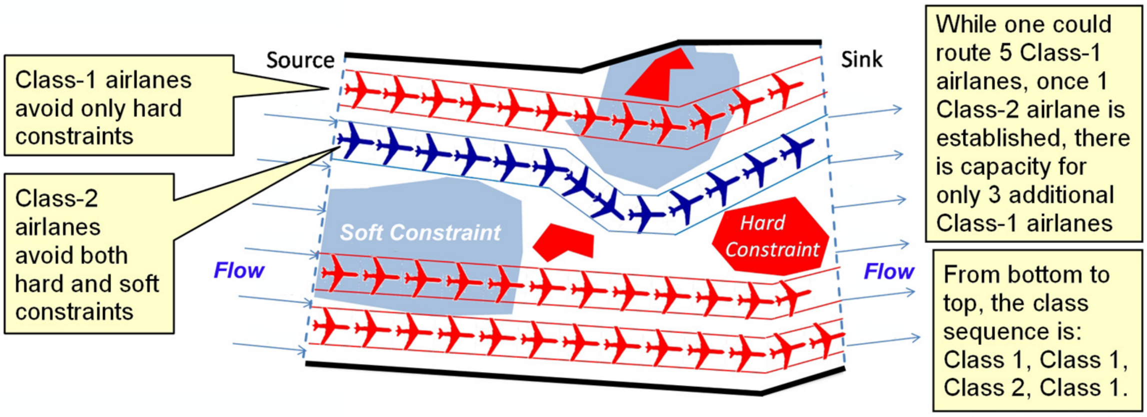


Figure 2. Airlane packing for two classes of aircraft among hard and soft constraints.

For throughput calculations, aircraft within the airspace are assumed to have a constant speed along each flow. While our model allows speeds to be different on different flows, our experiments assumed all aircraft had the same speed (we used 420 nmi/h). Additionally, the model allows for any specified horizontal separation requirement (which determines the airlane width, w); for our experiments, we assumed a horizontal separation requirement of 5 nmi for en route airspace. One may also specify an additional safety margin, δ with respect to hazardous weather constraints; airlanes are required to be at least horizontal distance δ from a constraint that it must avoid (we used δ=0). For simplicity, we do not account for the earth's curvature.

Hard and Soft Constraints

The severity of hazardous weather may be quantified by an intensity threshold in the National Weather Service (NWS) scale and a clearance level over the echo top height – the Convective Weather Avoidance Model (CWAM) [Song et al., 2006; DeLaura and Evans, 2006; Song et al., 2007; Chan et al., 2007; Song et al., 2008]. Regions of high convective weather intensity, such as CWAM Weather Avoidance Fields (WAFs) greater than a given threshold (e.g., 0.7), define airspace constraints. However, [Kuhn, 2008] suggests that there is no single threshold that defines a constraint region for all aircraft; instead, there is a need for more than one threshold to reflect different aircraft/pilot behavior with respect to convective weather constraints. This gives rise to the modeling of hard and soft constraint thresholds for TFM planning in the presence of convective weather: Regions with intensity above a threshold, τ_{hard}, define *hard* constraints, while those regions with intensity above a threshold, τ_{soft}, but below τ_{hard} ($\tau_{soft} < \tau_{hard}$), define *soft* constraints. Similarly, there are different levels of severity for turbulence and for icing, and these, together with pilot/airline preferences and aircraft type, determine regions defining hard or soft constraints. For example, as illustrated in **Figure 1**, the classification of turbulence or icing as SoG may define hard constraints, and the classification as MoG may define soft constraints.

Our analysis is with respect to short time intervals over which the forecast accuracy justifies the assumption that weather cells form static constraint regions. Yet, we discuss issues related to forecast uncertainty, and apply our methods to yield a stochastic throughput analysis. We leave for future work the extension of our methods to dynamic forecasts, for which we want to account explicitly for the time-varying nature of the weather constraints.

Objective

Our goal is to determine maximum throughput for a demand that consists of a mixture of Class-1 and Class-2 aircraft that are to cross airspace P from source to sink. Class-1 aircraft avoid hard constraints but are willing to fly through soft constraints, while Class-2 aircraft avoid both hard and soft constraints. Our formal goal is to establish if it is possible to route a specified number, I, of Class-1 airlanes and a specified number, J, of Class-2 airlanes. Each airlane is a thick path, whose centerline represents a route and whose width w represents the accuracy with which aircraft are expected to be able to navigate the route. Airlanes must be pairwise-disjoint and not overlapping the set of constraints relevant to the traffic utilizing the airlane (hard constraints for Class-1, and hard and soft constraints for Class-2). Thus, our problem is that of *packing* within the airspace a set of airlanes of two classes: I airlanes of Class 1 and J airlanes of Class 2, with each set of airlanes satisfying its corresponding set of constraints. Maximizing the throughput is a bicriteria optimization problem: one may want to maximize the number of Class-1 airlanes subject to a lower bound on the number of Class-2 airlanes or to maximize the number of Class-2 airlanes subject to a lower bound on the number of Class-1 airlanes. We provide algorithms that serve as centralized strategies for this optimization problem.

MAXIMUM FLOW RATE THEORY

We now review the theoretical solutions to the problem of computing a maximum flow rate from source to sink through an airspace.

Flows in Discrete Networks

Recall some basic notions and results about flows in graphs. A *directed graph*, $G=(N,A)$, consists of a set N of *nodes* and a set A of (directed) *arcs* (or *edges*) connecting pairs of nodes. A *flow network* is a directed graph G in which each edge e has a *capacity*, $c(e)$, and two nodes of N are designated as the *source* and the *sink*. All other nodes of N are *internal nodes*. A *flow* in G is an assignment of a *flow value*, $f(e) \leq c(e)$, to each edge e in A, such that for each internal node v the flow through v is *conserved*: the sum of the flows on edges going into v is equal to the sum of the flows on edges going out of v. The *value of the flow* in G is defined to be the total flow out of the source node, which equals (by flow conservation at internal nodes) the total flow into the sink node. In the *maxflow* problem, one seeks a flow with maximum value. A *cut* is a partition of the nodes N into two sets, X and Y, such that the source is in X and the sink is in Y. An edge $e=(u,v)$ *crosses* the cut if $u \in X$ and $v \in Y$. The *capacity* of the cut is the sum of the capacities of the crossing edges. The capacity measures the maximum amount of net flow possible from the source to the sink. A fundamental result in network flows is that *maxflow* equals *mincut*: the maximum flow possible is equal to the capacity of a minimum-capacity cut. This "*maxflow/mincut*" theorem is a manifestation of "duality" in linear programming and optimization theory; see, e.g., [Ahuja et al., 1993]. Efficient (polynomial-time) algorithms are known for computing maximum flows and minimum cuts in networks.

The notion of flow in a discrete network can be extended to a continuous geometric domain P [Strang, 1983; Mitchell, 1990]. Similar to the source and sink nodes in discrete networks, two edges on the boundary of the outer polygon of P are designated as the source and sink. Now, the source and sink edges split the outer boundary into two parts: top T and bottom B. More precisely, the outer boundary of P is partitioned into four portions, appearing in the clockwise order: bottom, source, top, and sink. Polygonal domain P contains *constraints*; the constraints are pairwise-disjoint simple polygons (**Figure 3**).

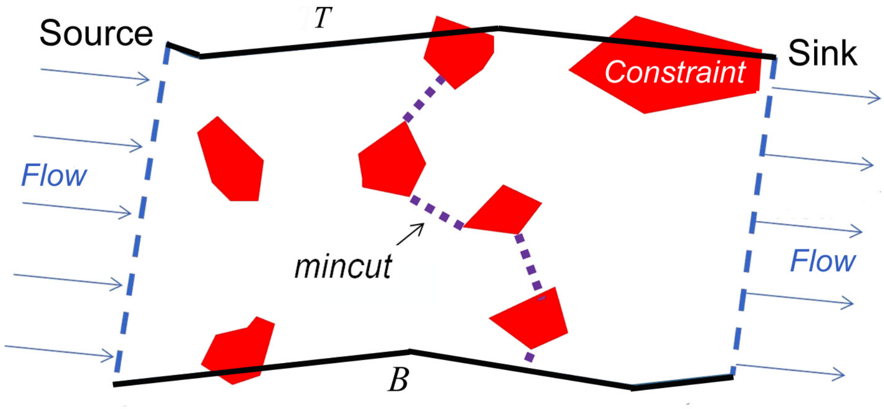


Figure 3. The Min-Cut defined by the source, sink, hazards, and sector geometry.

As with networks, a *cut* in *P* is a partition of the domain into two parts such that the source edge is in one part, and the sink edge is in the other; the *capacity* of the cut is the length of the common boundary of the parts. With a slight abuse of terminology, we usually refer to this common boundary as the *cut*. A *mincut* is a cut of minimum length. A mincut is therefore a sequence of line segments, connecting *B* to *T*, "hopping" from constraint to constraint in such a way as to minimize the total distance travelled within *P*. See **Figure 3.** [Strang, 1983] shows that, as in discrete networks, the maxflow/mincut theorem also holds for geometric domains, where flow refers to a divergence-free vector field in *P*, and the flow value is defined as the path integral, along the source, of the inner product of the flow field with an inward-pointing unit normal vector. (The divergence-free property of the flow field is the analogue of the flow conservation at internal nodes of a network.) [Mitchell, 1990] develops algorithms to compute a mincut efficiently in 2D polygonal domains using geometric shortest path techniques (the "continuous Dijkstra" paradigm); his algorithm also produces a flow field, which can be envisioned as a continuum of *flowlines* from source to sink. The mincut is the "bottleneck" to fluid flow from source to sink; its length quantifies the maximum achievable flow rate.

In order to apply the theory of continuous flows to our ATM model, we consider a variant of the theory, in which flowlines are grouped into discrete bundles of *thick paths* (airlanes of width *w*). The maxflow/mincut theory is extended to compute the maximum number of *thick paths* across the domain, from source to sink [Arkin et al., 2010]. Formally, a *thick path* is the Minkowski sum of a (thin) path, from a point on the source to a point on the sink, and a disk of diameter *w* centered at the origin. (E.g., if the thin path is polygonal, the thick path is the union of a set of rectangles of width w – one per edge of the centerline path, and a set of circles of diameter w – one per turn point of the centerline path). See **Figure 4**. The discrete maxflow problem in P is to compute a maximum number of pairwise-disjoint thick paths within P from source to sink. The maxflow/mincut theory extends to this "discrete" variant of the continuous maxflow problem [Arkin et al., 2010].

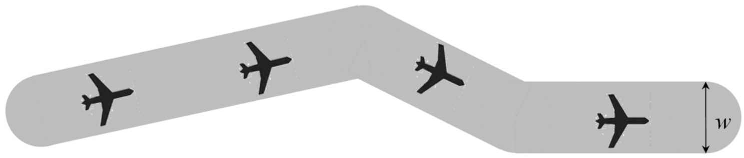


Figure 4. A thick path defines an airlane of width *w*.

Multi-Class Throughput Problem Formulation

We now describe a new problem, which generalizes the discrete maxflow in geometric domains to multi-class aircraft routing.

As before, the input to our problem is a polygonal domain P with a source and a sink edges on the boundary, and a set of polygonal constraints. The new aspect of the problem is that the constraints are of *K types* and the sought thick paths are of *M classes*. The width of a class-m path is w_m, $m=1,2,3,...,M$. A path of class m must avoid the constraints of types $O_m \subseteq \{1,\ldots,K\}$, but can pass through the constraints of the other types. (The specification of the sets O_m for each m gives the *weather impact interaction grid*, as discussed in [Lindholm, et al, 2009].) The capacity estimation problem is: Given integers $n_1, n_2, n_3, ..., n_M$, decide if there exist n_1 class-1 paths, n_2 class-2 paths,..., and n_M class-M paths from source to sink through P, with no overlap among the (thick) paths, and each class-m path avoiding constraints of types in the set O_m.

While the model just described applies to a wide range of possible types of constraints and aircraft classes and interactions between them, our focus for the remainder of the paper is on the case of just two types of constraints (i.e., $K=2$) and two classes of aircraft (i.e., $M=2$). Specifically, in our airlane routing problem, the constraints are either *hard* (type 1) or *soft* (type 2). The two classes of aircraft/paths, Class-1 and Class-2, have the sets of obstacles specified by $O_1=\{1\}$ and $O_2=\{1,2\}$, indicating that Class-1 aircraft avoid hard constraints (but can travel through soft constraints), while Class-2 aircraft avoid both hard and soft constraints.

Class Sequences

For any set of airlanes through P linking the source to the sink, the airlanes are ordered from bottom B to top T, with the ith airlane being the one with (i-1) airlanes below it. The specification of the ordered list of path classes is a *class sequence*, $C=(c_1,c_2, \ldots)$, where each c_i is one of the M classes of airlanes. Here, $M=2$, so each class sequence is a sequence of 1's and 2's, which we often denote without the parentheses and commas (e.g., 11222122 instead of (1,1,2,2,2,1,2,2). For example, **Figure 2** illustrates the class sequence C=1121.

Bottommost Paths

Paths (airlanes) $p_1, p_2, \ldots, p_m$ are called *bottommost paths* if p_1 runs "as close as possible" (in a sense made rigorous in [Arkin et al., 2010]), to the bottom B, and p_i runs as close as possible to p_{i-1}, for $i=2,3,..,m$.

Theoretical Results

[Kim et al., 2009] prove that if there exist m paths with class sequence C, then there exist m bottommost paths with the same class sequence C, and the paths can be found efficiently by a "bottommost filling of the domain" with the paths. Therefore, if the class sequence C is given, the capacity estimation problem can be solved efficiently by computing bottommost paths. However, [Kim et al., 2009] also show that, in general, the problem is NP-hard if the type sequence is not known, even if there are only two classes: it is NP-hard to decide if it is possible to route I lanes of Class 1 and J lanes of Class 2. An *alternation* in a class sequence C is a place where C changes from 1 to 2 or vice versa (note that the number of alternations, n_a, can be larger than the number of classes, e.g., the sequence 1211221212 has $n_a = 7$); when n_a is small, the problem can be solved exactly in polynomial time, with the exponent in the running time depending on n_a. Further, from the point of view of approximation algorithms, [Kim et al., 2009] show that for the problem in which there are two classes of paths and two types of constraints, one can obtain a provable approximation to the optimal number of Class-1 and Class-2 paths.

ALGORITHMIC SOLUTION APPROACHES

Next, we outline a routing algorithm that solves the problem: Given two integers, I and J, route I Class-1 paths and J thick Class-2 paths, each having width w, in a polygonal region P populated with hard and soft constraints. The solution for two classes does not run in worst-case polynomial time (after all, it is known to be an NP-hard problem, as already noted); instead, we devise an algorithm that has running time dependent on the number of alternations in the class sequence, a number that is expected to be small in practice (and, in fact, one may expect from a practical point of view that one is never interested in solutions having a huge alternation number, n_a). For problems with more than two classes/types of thick paths and constraints, with possibly different thicknesses, our methods extend to yield a similar solution, again with dependence on the number of alternations in the class sequences the algorithm enumerates.

Our algorithm works by scrolling though class sequences. For each class sequence C we use an efficient way to either compute the paths with the class sequence given by C (in which case the algorithm terminates with a success), or to determine that no collection of paths with sequence C exists (in which case we proceed to the next sequence). The total number of different sequences equals to $(I+J)!/(I!J!)$ which is the number of ways to choose I places for 1 and J places for 2 in a sequence of 1's and 2's of total length $I+J$; here $N!$ denotes *N-factorial* – the product of all integers from 1 to N. This number grows fast as a function of I and J; hence, we use several heuristics (described below) to guide the scrolling and prune large fraction of infeasible sequences.

For any particular given class sequence C, we find the paths by a bottommost fill, the details for which are presented next.

Bottommost Path Filling Algorithm

Rather than working with the continuum of all possible thick paths, we take a practical approach to generating bottommost paths that are restricted to a discrete set of possibilities, by searching for paths within a regular square grid V inside P. We add to V a group of source and sink nodes, which are discrete points uniformly distributed along the source and sink edges. See **Figure 5**.

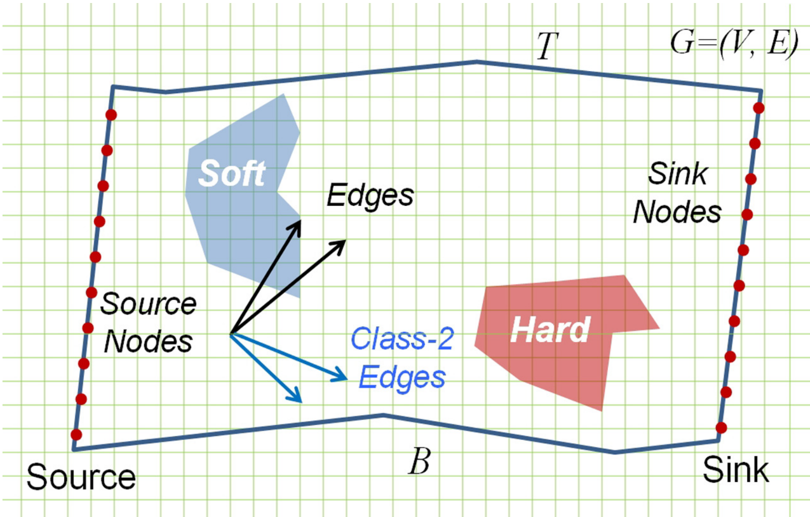


Figure 5. Graph *G* used in a bottommost path filling algorithm.

A directed graph $G=(V,E)$ is formed. The edge set E contains directed edges (p,q) connecting points $p=(p_x, p_y)$ and $q=(q_x, q_y)$ in V whenever q is to the right of p, and is "close" to p; specifically, we require that $1 \leq q_x - p_x \leq D$, using $D=5$ (a default parameter). We keep only those segments pq as edges in E, for which no point of pq lies within distance $w/2$ of any hard constraint. Moreover, within the set E, we mark an edge pq as a *Class-2* edge if, in addition, no point of pq lies within distance $w/2$ of any soft constraint. This models the fact that all edges are feasible for Class-1 aircraft, while only Class-2 edges are feasible for Class-2 aircraft.

In our bottommost path filling algorithm, we compute bottommost paths from source nodes to sink nodes within graph G, according to the path class. Since our directed edges are all oriented from left to right (recall that $1 \leq q_x - p_x$ for edge (p,q)), the computed paths are necessarily x-monotone; this enforces that the paths make progress in the direction from source to sink (left to right), without doubling back. One bottommost path is routed by a Depth-First Search (DFS) [Cormen et al., 2009] within G. Specifically, the path originates at the bottommost feasible source node, and progresses monotonically to the right, giving preference to the bottommost (most clockwise) feasible edges leading out of a node. The DFS succeeds in finding a path if it reaches a sink node; otherwise, the search from the source node fails, and we move to routing from the next available source node. **Figure 6** shows an execution of the algorithm for the sequence "121". At the highlighted nodes in the search, there is branching, as the depth-first search has to retrace its steps and choose a different branch, after having hitting a "dead end".

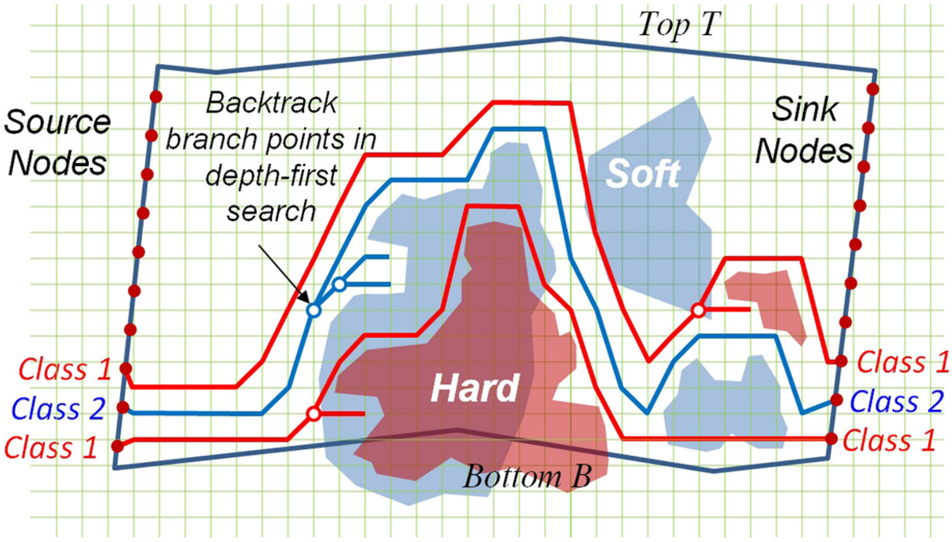


Figure 6. The bottommost fill algorithm for sequence "121".

At the completion of the bottommost fill, we either have a sequence of I Class-1 and J Class-2 paths as desired, or we know that the current choice of class sequence is infeasible, so the algorithm moves on to the next class sequence.

Sequence Pruning. Our algorithm maintains a *black list* of infeasible subsequences. Sequences that begin with a black-listed subsequence are immediately ruled out as infeasible. For instance, if sequence "1212212121" fails after testing only "1212", then "1212" is added to the black list so that all permutations starting with "1212" are immediately recognized as infeasible (without having to perform a bottommost fill search).

Leftover Space Testing. After routing a bottommost path, our algorithm computes two mincuts in the leftover space P_l above the routed paths in P: the mincut m_1 in P_l with only hard constraints and the mincut m_2 in P_l with both hard and soft constraints. We use the capacities m_1 and m_2 to judge if the leftover space can accommodate the remaining Class-1 paths and Class-2 paths. For instance, if we have successfully routed already $i<I$ Class-1 paths and $j<J$ Class-2 paths, then we know that the remaining $(I-i)+(J-j)$ paths (of either class) cannot be routed if $m_1<(I-i)+(J-j)$; this is because m_1 is the maximum number of thick paths that can be routed through P_l. Similarly, we know that the remaining $(J-j)$ Class-2 paths yet to be routed cannot exist in P_l if $m_2<J-j$; this is because m_2 is the maximum number of thick paths that can be routed through the remaining airspace while avoiding both hard and soft constraints.

Strategic Enumeration of Class Sequences. The order in which class sequences are explored makes a considerable difference in the speed with which our algorithm finds a routable class sequence, if one exists. (Recall that the algorithm concludes once a first feasible class sequence is discovered, for which a bottommost fill succeeds.) Instead of enumerating class sequences in arbitrary order or in a natural lexicographic order, our algorithm uses the order of increasing number of class alternations. For example, if we are searching for $I=5$ Class-1 paths and $J=5$ Class-2 paths, then lexicographic ordering would explore (1111122222, 1111212222, …), while enumeration by increasing number of class alternations would explore (1111122222, 2222211111, …). [Kim et al., 2009] experimentally proved that this enumeration strategy helps find a feasible class sequence much faster than some competing alternatives.

Postprocessing: Taughtening Bottommost Paths

The bottommost paths are generally unreasonably long and are unrealistic for ATM. Hence after the bottommost paths are routed, we "pull them taut" using an iterative local shortening heuristic. Specifically, we take the last-but-one topmost path, p_{I+J-1}, and declare it as an obstacle. Then, in the space between the top T and the obstacle p_{I+J-1}, we compute the shortest path, p^*_{I+J}, in G, from the source to the sink, that is of the same class and width as p_{I+J}. It is obvious that p^*_{I+J} is a feasible path because it does not penetrate any constraints that it has to avoid. We replace p_{I+J} with p^*_{I+J}. Next, we declare p^*_{I+J} and p_{I+J-2} as obstacles, and replace p_{I+J-1} with the shortest same-class path p^*_{I+J-1} between p^*_{I+J} and p_{I+J-2}. Continuing this way, we obtain a set of $I+J$ short paths, $p^*_{I+J},\ldots,p^*_{1}$. Our experimental results indicate that this taughtening heuristic is very effective in producing flyable paths that are monotonic in the direction from source to sink (see **Figures 7(d,f)** and **9(c)**).

EXPERIMENTS

We applied our solutions from the previous section to two sets of data. The first is the real-world weather forecast maps based on GTG and CIP forecasts over the NAS at 3:00 pm on Jan. 24, 2007 at 38000 ft. The second set is based on synthesized weather data that simulates real weather data and allows us to investigate the effectiveness of our algorithms on a broader class of constraints than is readily available in selected real datasets.

Real Weather Data

In our experiments with real weather data, the FEA is a rectangular domain extracted from a GTG map; the FEA covers a portion of Midwest of the US. Aircraft fly through the FEA from West to East; i.e., the source is the west side of the rectangle and the sink is the right side. (For our analysis it would not matter if the roles of source and sink are reversed.) The distance from the northern to the southern boundary of the FEA is 313 nmi. We assume that both Class-1 and Class-2 airlanes have width 8 nmi, with an additional separation of 8 nmi between airlanes; thus, the total distance from centerline to centerline of two adjacent airlanes is 16 nmi.

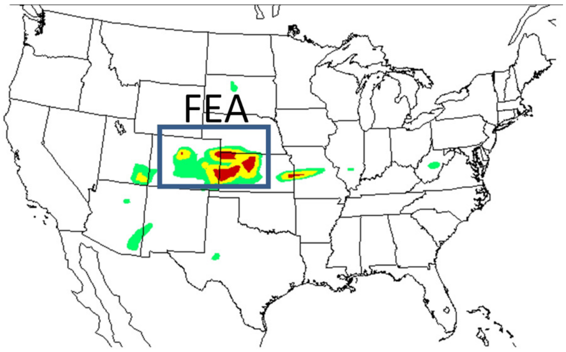
We conducted three experiments: (1) Given two integers I and J, as well as a designated class sequence C of I 1's and J 2's, determine if I Class-1 and J Class-2 airlanes can be routed from source to sink according to C. (2) Given integers I and J, but no class sequence, determine if I Class-1 and J Class-2 airlanes can be routed from source to sink. (3) Given prescribed entry and exit points on the source and sink edge*s*, as well as integers I and J, determine if there exist I Class-1 and J Class-2 airlanes between the entry and the exit points. Results of these experiments are shown in **Figure 7**.

Experimental Results

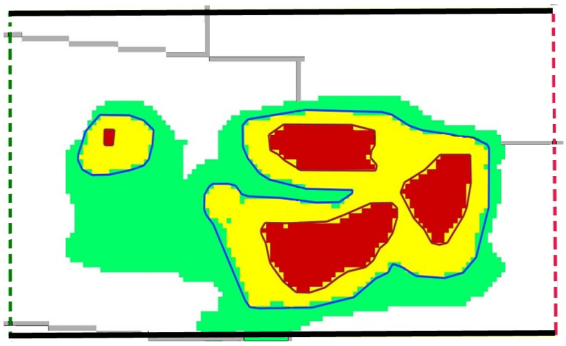
We tested over 2000 test cases, including both real weather data and synthesized weather data cases. Results show that our capacity estimation algorithm is practical and efficient. The algorithm's implementation is fast because it tests far fewer than the maximum possible number of sequences. For almost all test cases where the given lanes are routable, the algorithm tests only very few permutations (less than 10), even for very large scenarios, e.g. $I+J>100$. The test for each permutation is very fast: For a rectangular region partitioned into a base grid of 125-by-63 cells, in which there are 5 obstacles of average size (area) 176 cells, it takes 11.61ms, on average, to test a specific class sequence (using a 2.6G Intel CPU). For smaller regions that have sparse obstacles, e.g. with a base grid of 40-by-40 cells (a resolution that likely suffices for an airspace sector), it takes on average 3.32ms to test one class sequence. Therefore, for routable test cases, the algorithm usually reports results very quickly (less than 4 seconds even for very large instances).

For fail-to-route test cases, if there are only sparse obstacles, the algorithm usually returns "unroutable" very quickly (within 5 seconds), thanks to leftover space testing and sequence pruning. But there are extremely slow fail-to-route cases for which the algorithm must test up to 20% of the maximum number, $(I+J)!/(I!J!)$, of class sequences. (This, of course, is expected, since we know the problem is NP-hard.) Fortunately, for cases in which $I+J<20$, the algorithm was always able to report the result in less than 120 seconds, in all of the experiments.

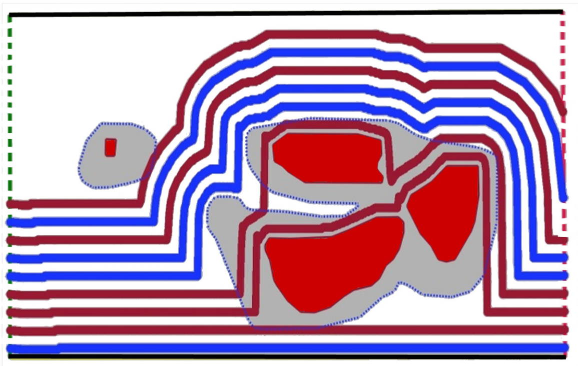
The path tautening phase of the algorithm was found, on average, to take, per path being optimized, 2.16 s for the smaller instances (based on a 40-by-40 grid of cells), and 7.97 s for the large instances (based on a 125-by-63 grid of cells). Examples of paths after tautening are shown in **Figures 7(d,f) and 9(c)**; the resulting paths appear to be reasonable for ATM applications.



(a) GTG map

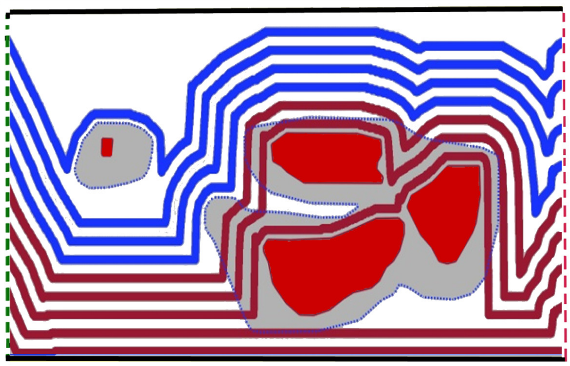


(b) FEA and constraints

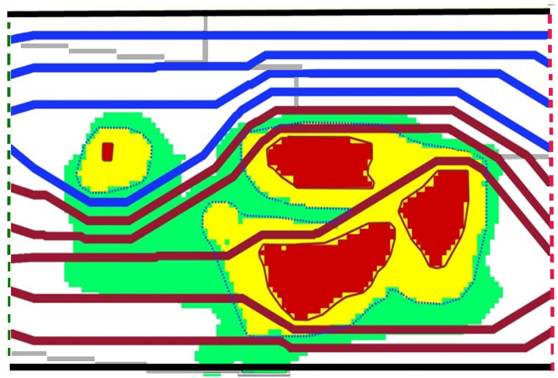


(c) Bottommost paths

(d) Pulled taut paths



(e) Bottommost paths with given endpoints



(f) Paths from (e) pulled taut

Figure 7. Experiments using real weather, with I=5, J=4.

Probabilistic Weather Maps

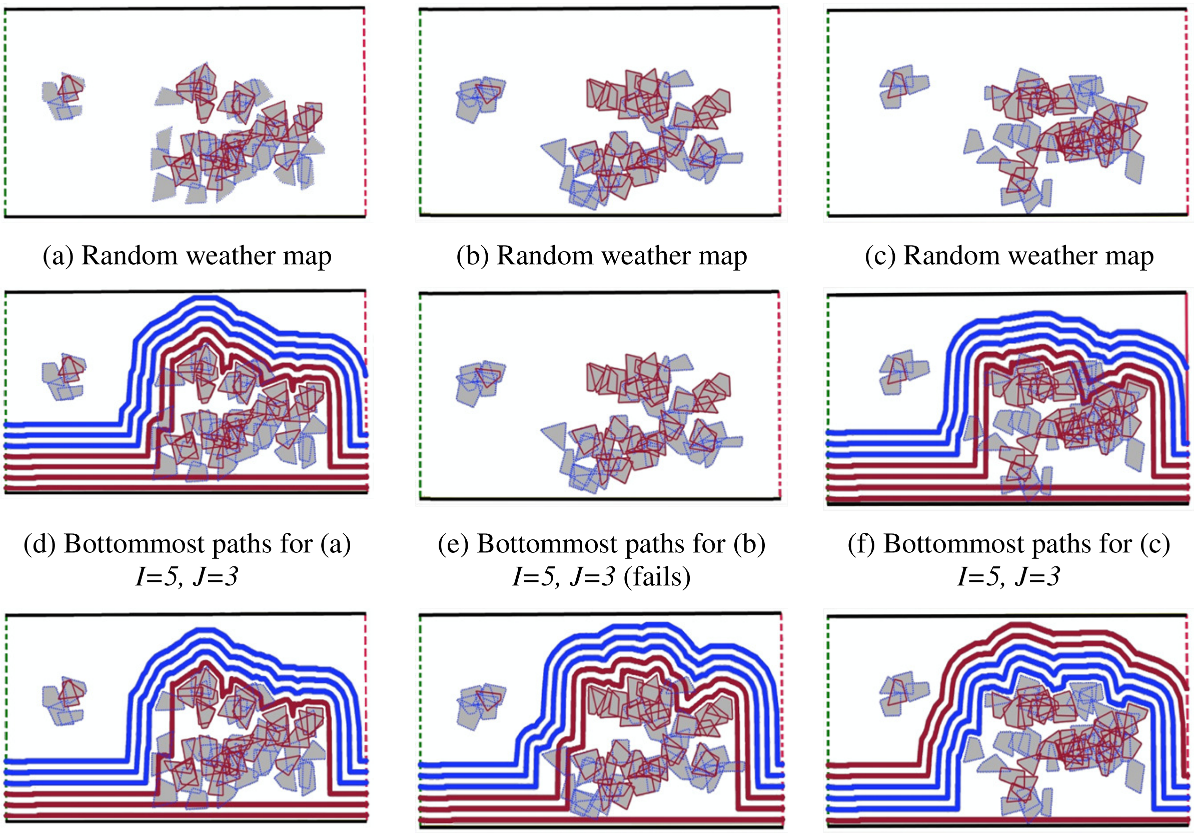
So far we assumed that the position and size of the constraints in the airspace were known precisely in advance, with no source of randomness. While this assumption may be valid over short time horizons (when forecast data is highly accurate), it is more realistic to expect that weather data is described by a *stochastic* forecast, perhaps represented as a weighted ensemble set of forecasts (with weights corresponding to probabilities or "beliefs"). The ensembles may, e.g., be generated by numerical prediction models with randomly perturbed initial conditions, by multiple models applied to the same initial conditions, or by a data assimilation process. In [Mitchell et al., 2006], maximum throughput and capacity estimation is studied for stochastic weather models based on an ensemble of forecasts. Specifically, for each forecast, the mincut is computed and the probability distribution of the mincut value is calculated explicitly, based on the probabilities associated with members of the ensemble. More sophisticated stochastic weather models, e.g., based on probabilistic weather maps and ensembles of probabilistic weather maps, can also be considered, together with an explicit modeling of the spatial correlation between nearby points.

These methods can form the basis of a TFM strategy for mixed equipage aircraft classes involved in an AFP [Krozel, et al, 2006]. Specifically, for a given probabilistic weather map associated with a look-ahead time (e.g., 1-h, 2-h, or 3-h forecasts), predicting levels of turbulence and/or icing, we can compute the probability of being able to route I airlanes for Class-1 traffic and J airlanes for Class-2 traffic. Then, based on how high these probabilities are, a threshold-based policy can be used to decide whether to continue flights on their predicted flow across a FEA, or to reroute the flights around the FEA, or to perform an altitude change (based on computing the probabilities of routability at other altitudes using our capacity algorithm). That is, we convert a probabilistic forecast to an ensemble of forecasts, each with an associated probability. Then, we run our deterministic algorithm for each member of the ensemble, and compile the data to determine the probability of successful routing over a full set of ensemble forecasts.

In our experiments, given one forecast for the region of interest P, which yields a set of hard/soft constraints (based on MoG or SoG levels of turbulence or icing), we generate an ensemble of forecasts to represent a stochastic forecast model. The generation is based on randomly selecting a seed point q inside P. If q is within a constraint (hard or soft), we place, with probability p (a user-defined parameter, close to 1) a random polygonal constraint centered at q. (In our experiments, we used random quadrilaterals, generated by selecting four points at random in the four quadrants centered at q; other random shapes are possible too, e.g., random disks, polygons, etc.) If the seed point q falls outside all constraints then random polygonal constraint is placed at q with probability 1-p. This way we obtain a set of forecasts each looking similar to the nominal input forecast.

The test results of the probabilistic weather maps are shown in **Figure 8**. Based on the same weather map that was used in **Figure 7**, we generate 1000 random instances of a weather forecast and test the probability that I Class-1 and J Class-2 airlanes can be routed through the FEA by executing the algorithm on each generated instance (member of a synthetic ensemble).

Additional results for capacity estimation based on synthesized weather data are shown in **Figure 9**. The algorithm succeeds upon discovering the routable class sequence "1111122222211112", which is found quickly, since it has only 3 alternations.



(a) Random weather map

(b) Random weather map

(c) Random weather map

(d) Bottommost paths for (a) I=5, J=3

(e) Bottommost paths for (b) I=5, J=3 (fails)

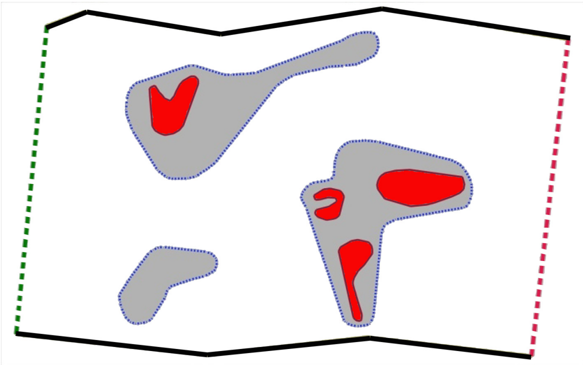
(f) Bottommost paths for (c) I=5, J=3

(g) Bottommost paths for (a) I=3, J=3

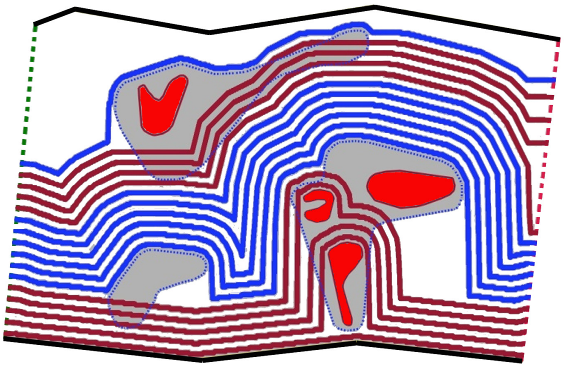
(h) Bottommost paths for (b) I=3, J=3

(i) Bottommost paths for (c) I=3, J=3

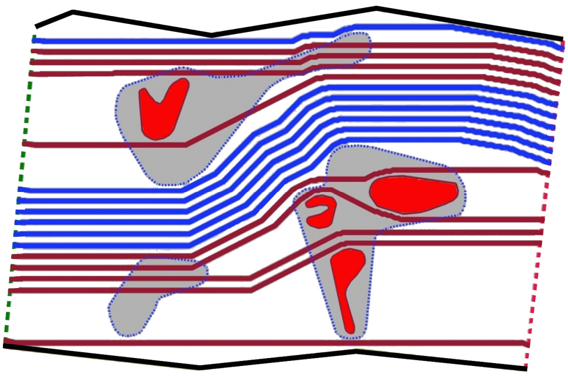
Figure 8. Paths computed using probabilistic weather maps.



(a) Airspace model



(b) Bottommost paths



(c) Paths pulled taut

Figure 9. Synthesized weather data, with *I*=9, *J*=7.

CONCLUSIONS

We present a mathematical model and an algorithmic method for capacity estimation in airspaces with hard and soft constraints for two classes of aircraft. We show how this algorithm can be used to generate flexible airlanes that cross over a region of airspace with hard and soft constraints, given the preferences of pilots (air carriers) for the weather hazards that they are willing to fly over. We discuss related theoretical results and propose a practical algorithm for throughput computation and routing of two-class airlanes among hard and soft constraints. Further, we demonstrate results from these algorithms and report on experience in applying them to perform capacity estimation using deterministic and probabilistic weather maps based on real weather data as well as on synthetic data. The algorithms presented here are well grounded in theory and are shown experimentally to be practical and efficient. For future applications such as crossing flow constrained areas (FCAs) in the National Airspace System (NAS), these algorithms are suitable for the estimation of the capacity of the airspace (such as an FCA) and the design of flexible routing structures crossing such an airspace.

FUTURE RESEARCH

Future research includes:

1. Better approximation algorithms for capacity estimation in the presence of hard and soft constraints. Can we obtain an approximation algorithm for maximizing the number, I, of Class-1 airlanes, given a lower bound on the number, J, of Class-2 airlanes? (Known approximation algorithms [Kim et al., 2009] relax the optimality of the number of airlanes of both classes.)
2. Allowing non-monotone airlanes. (Our implementation is based on an algorithm that searches over sets of x-monotone airlanes.)
3. 3D airspaces, allowing multiple altitudes and climb/descent profiles. In the case of turbulence, for instance, the preferred avoidance maneuver is a vertical maneuver, rather than a horizontal re-routing maneuver [Krozel et al, 2011].
4. Multiple sources/sinks on the boundary of the airspace and to crossing patterns of traffic, in which the demand includes flows of aircraft that must cross. This introduces *scheduling* into the problem, and the search becomes one of computing a maximum number of thickened "tubes" in space-time, with tubes of different classes, corresponding to the aircraft capabilities with respect to hard and soft constraints. Our current techniques do not extend straightforwardly to this scenario. A possible future approach may be to apply the techniques of this paper, for noncrossing patterns, in each of the major flow directions; then, a scheduling algorithm can be used to select trajectories within each "layer" of the solution, in order to avoid conflicts.

5. Three or more types of constraints ($K \geq 3$) and three or more aircraft classes ($M \geq 3$). This will allow us to compute capacities for multiple classes of aircraft whose constraints are determined by a general *weather impact interaction grid* [Lindholm, et al, 2009].
6. Modeling limitations that come from controller workload for monitoring the airspace.
7. Dynamic aspects of the problem, including: moving weather cells, changing traffic composition, and organizing classes of traffic prior to entering into a FCA. Our methods assume that the traffic mix is known (or at least estimated) over a given time horizon. In order to address general changes in traffic mix over a planning time horizon, it is necessary to solve the problem in the space-time domain, e.g., using techniques of [Arkin et al., 2010]; these techniques have not yet been generalized to the mixed equipage domain, though, so this remains a topic for future research. If weather cells move, particularly if they move "as a whole", then it would be advantageous to extend our work to design "flexible" airlanes, as has been studied recently by [Krozel et al., 2010] in the case of a single class of aircraft.
8. Introduction of a cost function. While our post-processing method of pulling paths taut is a heuristic intended to produce short paths, it does not come with theoretical guarantees on the optimality of the set of paths. The problem of computing an "optimal" set of (thick) paths in a domain is known as the geometric minimum cost flow problem, which has been studied by [Mitchell and Polishchuk, 2007] for single class flows. Future work will examine the extension of that theory to hard/soft constraints and multiclass (multicommodity) flows.

ACKNOWLEDGEMENTS

This research was funded by NASA Ames Research Center under contract NNA07BB36C for the NextGen ATM – Airspace Project – Translation of Weather Information to Traffic Flow Management Impact. The authors appreciate the frequent inputs from our contract monitor, Mr. William Chan from NASA Ames Research Center. Finally, we appreciate the financial support of the sponsor of the research, NASA NextGen Project Manager, Dr. Paramal Kopardekar. J. Mitchell is partially supported by the National Science Foundation (CCF-0729019, CCF-1018388). V. Polishchuk is supported by a personal grant from the Academy of Finland.

LIST OF ACRONYMS AND SYMBOLS

A	set of directed arcs in a directed graph

AFP	Airspace Flow Program
B	the bottom of a polygonal domain
AFP	Airspace Flow Program
ATM	Air Traffic Management
$c(e)$	capacity of an edge e in a directed graph
C	class sequence
CIP	Current Icing Potential
CWAM	Convective Weather Avoidance Model
DFS	Depth-First Search
e	arc in a directed graph
E	edges of the graph used for path searching in the bottommost filling algorithm
$f(e)$	flow value on an arc
FEA	Flow Evaluation Area
FCA	Flow Constrained Area
$G(V,E)$	graph used for path searching in the bottommost filling algorithm
GA	General Aviation
GTG	Graphical Turbulence Guidance
I	number of Class-1 paths (airlanes)
J	number of Class-2 paths (airlanes)
K	number of different types of constraints
M	number of different classes of aircraft/airlanes
maxflow	a flow of maximum value
mincut	a cut of minimum capacity, or the size of such a minimum-capacity cut
MAP	Monitor Alert Parameter
MIT	Miles-in-Trail
MoG	Moderate-or-Greater
N	set of nodes in a directed graph
NAS	National Airspace System
NextGen	Next Generation Air Transportation System
NWS	National Weather Service
O_m	set of constraints that a type-m thick path avoids

P	region of interest which is a polygonal domain
p_i	ith thick path
P_l	the space in the leftover space testing
RNAV	Area Navigation
RNP	Required Navigation Performance
SESAR	Single European Sky ATM Research Joint Undertaking
SIGMETS	Significant Meteorological Information
SoG	Severe-or-Greater
SUA	Special Use Airspace
τ_{hard}, τ_{soft}	thresholds for defining the hard and soft constraints
T	the top of a polygonal domain
TFM	Traffic Flow Management
V	vertices of the graph used for path searching in the bottommost filling algorithm
w	width of a thick path
WAF	Weather Avoidance Field
X, Y	division of N into two sets, X and Y: source node is in X, sink is in Y

REFERENCES

Arkin, E.M., Mitchell, J.S.B. and Polishchuk, V., (2010), "Maximum Thick Paths in Static and Dynamic Environments," *Computational Geometry Theory and Applications,* 43(3).

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993), *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Brennan, M., (2007), "Airspace Flow Programs – A Fast Path to Deployment," *Journal of Air Traffic Control*, 49(1).

Chan, W., Refai, M., and DeLaura, R., (2007), "Validation of a Model to Predict Pilot Penetrations of Convective Weather", *AIAA Aviation, Technology, Integration and Operations Conf.*, Belfast, Northern Ireland, Sept.

Chiang, Y.-J., Klosowski, J.T., Lee, C., and Mitchell, J.S.B. (1997), "Geometric Algorithms for Conflict Detection/Resolution in ATM", *IEEE Conf. on Decision and Control*, San Diego, CA.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2009), *Introduction to Algorithms*, MIT Press, Cambridge, MA.

DeLaura, R. and Evans, J. (2006) “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.

Kim, J., Mitchell, J. S. B., Polishchuk, V., Yang, S., Zou, J., (2009), “Routing Multi-Class Traffic Flows in the Plane”, Technical Report, Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY.

Krishna, S., and Krozel, J., (2009) “Impact Analysis for In-Flight Icing Hazards,” *AIAA Guidance, Navigation, and Control Conf.,* Chicago, IL, Aug.

Krozel, J., Jakobovits, R., and Penny, S., (2006), “An Algorithmic Approach for Airspace Flow Programs,” *Air Traffic Control Quarterly*, 14(3).

Krozel, J., Klimenko, V., and Sharman, R., (2011) “Analysis of Clear-Air Turbulence Avoidance Maneuvers,” *Air Traffic Control Quarterly*, To Appear.

Krozel, J., and Murphy, J.T., Jr., (2007), “Weather Hazard Requirements for NGATS Aircraft,” *Integrated Communications, Navigation, and Surveillance Conf.*, Herndon, VA, May.

Krozel, J., Mitchell, J.S.B., Paakko, A., and Polishchuk, V. (2010), “Throughput/Capacity Tradeoffs for Routing Traffic in the Presence of Dynamic Weather,” *Proceedings of the 4th International Conference on Research in Air Transportation*, Budapest, Hungary, June.

Krozel, J., Mitchell, J.S.B., Polishchuk, V., and Prete, J., (2007), “Maximum Flow Rates for Capacity Estimation in Level Flight with Convective Weather Constraints,” *Air Traffic Control Quarterly*, 15(3).

Krozel, J., Prete, J., Mitchell, J.S.B., Kim, J., and Zou, J., (2008), “Capacity Estimation for Super-Dense Operations,” *AIAA Guidance, Navigation, and Control Conf.*, Honolulu, HI, Aug.

Kuhn, K., (2008) “Analysis of Thunderstorm Effects on Aggregated Aircraft Trajectories,” *Journal of Aerospace Computing, Information and Communication*, Vol. 5, April.

Lindholm, T., Krozel, J., and Mitchell, J.S.B., (2009), “Concept of Operations for Addressing Multiple Types of En Route Hazardous Weather Constraints in NextGen,” *AIAA Aviation Technology, Integration, and Operations Conf.*, Hilton Head, SC, Sept.

Mitchell, J. S. B. (1990), "On Maximum Flows in Polyhedral Domains," *Journal of Computer and System Sciences*, Vol. 40, pp. 88-123.

Mitchell, J.S.B. and Polishchuk,V., (2007). "Thick non-crossing paths and minimum-cost flows in polygonal domains," *Proceedings of the 23rd Annual ACM Symposium on Computational Geometry*, Gyeongju, South Korea, June.

Prete, J., and Mitchell, J.S.B., (2004), "Safe Routing of Multiple Aircraft Flows in the Presence of Time-Varying Weather Data," *AIAA Guidance, Navigation, and Control Conf.*, Providence, RI, Aug.

Prete, J., (2007), *Aircraft Routing in the Presence of Hazardous Weather,* Ph.D. Thesis, Computer Science, Stony Brook University, Stony Brook, NY.

RTCA (1995), *Report of the RTCA Board of Directors' Select Committee on Free Flight,* RTCA, Inc., Washington, DC, Jan.

Schmidt, D. K. (1975), "On Modeling ATC Work Load and Sector Capacity," *Journal of Aircraft*, 13(7).

Song, L., Wanke, C., and Greenbaum, D. (2006), "Predicting Sector Capacity for TFM Decision Support," *6th AIAA Technology, Integration, and Operations Conf.*, Wichita, KS, Sept.

Song, L., Wanke, C. and Greenbaum, D., (2007), "Predicting Sector Capacity under Severe Weather Impact for Traffic Flow Management," *AIAA Aviation Technology, Integration, and Operations Conf.*, Belfast, Northern Ireland, Sept.

Song., L., Wanke, C., Greenbaum, D., Zobell, S, and Jackson, C., (2008), "Methodologies for Estimating the Impact of Severe Weather on Airspace Capacity," *26th Intern. Congress of the Aeronautical Sciences,* Anchorage, AK, Sept.

Strang, G., (1983), "Maximal Flow through a Domain," *Mathematical Programming*, Vol. 26, pp. 123–143.

Van den Berg, J.P. and Overmars, M.H., (2005). "Prioritized Motion Planning for Multiple Robots," *IEEE/RSJ International Conf. on Intelligent Robots and Systems,* Edmonton, Canada, Aug.

BIOGRAPHIES

Shang Yang is a Research Assistant at Stony Brook University, working towards a Ph.D. in Computer Science. His primary area of specialization is algorithms and computational geometry, particularly in applications to aircraft routing and ATM.

Joseph S. B. Mitchell received a BS (1981, Physics and Applied Mathematics), and 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 ATM, sensor networks, computer graphics, visualization, manufacturing, and geographic information systems.

Jimmy Krozel is a Principal Analyst 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 80 technical publications, and is the winner of two AIAA best paper awards. His research interests include computational geometry, visualization, intelligent path prediction, intent inference, and weather impacts on ATM.

Valentin Polishchuk received a diploma (1996, Applied Physics and Math) from Moscow Institute of Physics and Technology and an MS (2002, Operations Research) and Ph.D. (Applied Mathematics and Statistics, 2007) from Stony Brook University. Polishchuk worked as a researcher in the Environmental Modeling Lab of the Nuclear Safety Institute, Moscow (1998-2000), and as a research assistant in the Applied Mathematics and Statistics Department at Stony Brook, working in the areas of optimization, algorithms, and computational geometry with applications in ATM. Since 2007, he is a postdoctoral researcher with the Helsinki Institute for Information Technology.

Joondong Kim is a Software Engineer at Google. At the time of this study, Dr. Kim was a Research Assistant with the Department of Applied Mathematics and Statistic, Stony Brook University. He holds a Ph.D. in Applied Mathematics and Statistics from Stony Brook University. His primary area of specialization is optimization, algorithms and computational geometry, particularly in applications to vehicle routing and ATM.

Jingyu Zou is a Software Engineer at Google. At the time of this study, Dr. Zou was a Research Assistant with the Department of Applied Mathematics and Statistic, Stony Brook University. He holds a Ph.D. in Applied Mathematics and Statistics from Stony Brook University. His primary area of specialization is optimization, algorithms and computational geometry, particularly in applications to vehicle routing and ATM.