

# Trajectory-based Air Traffic Management (TB-ATM) under Weather Uncertainty

Arnab Nilim\*      Laurent El Ghaoui†      Mark Hansen‡      Vu Duong §

## Summary

Explosive growth of commercial air travel poses great challenges to air traffic management and causes increasing delays. Much of the delay is weather induced. The traditional routing strategy, which consists of an aircraft under a predicted storm avoiding the bad weather zone completely, is too conservative. One way to reduce delays caused by enroute weather is to take a less conservative route, risking higher delay to attain a better expected delay, instead of avoiding the bad weather zone completely. In order to implement this strategy, one approach involves shifting the current airspace-based air traffic management system to a trajectory-based air traffic management (TB-ATM) system where there is only one controller responsible for each aircraft from gate to gate. We address the single aircraft problem using Markov decision processes (where the weather processes is modeled as a stationary Markov chain) and a dynamic programming algorithm. The approach provides a dynamic routing strategy for an aircraft that minimizes the expected delay, when the aircraft's nominal path may be obstructed by a bad weather. Our algorithm is implemented in different aircraft routing scenarios and the improvements in delays in comparison with the traditional methods are shown.

## Overview

The growing volume of air traffic is straining the limits of the FAA/Eurocontrol's air traffic infrastructure as well as key airports' runway capacity. According to the FAA, flight delays have increased by more than 58 percent since 1995, cancellations by 68 percent. The cost of delays to airlines and passengers are billions of US dollars per year. The air traffic flow management problem under deterministic environment is well addressed in [1], [2], [3], [12], [13], [14]. The major portion of this delay is due to bad weather which is stochastic in nature. There is an urgent need for automation tools which explicitly deal with the dynamics and the stochasticity of the storms and provide solutions that reduce the expected delay in the air traffic control system.

In spite of the tremendous technological advancement, the current air traffic management system uses the technologies that were developed at least a decade ago. The current airspace architecture is not designed properly to include all of these advanced technological features efficiently in the system. When an aircraft goes from one point to another point, it flies through a number of air-sectors. The size of an air sector depends on the number of aircraft in that region. The capacity of an air sector is around 40 aircraft at a time. There are two controllers in each air-sector to handle the traffic, a planning controller and an executing controller. The planning controller works at a strategic level and tries to minimize the number of conflicts. The executing controller works at a tactical level and ensures that there is no conflicts. There is an inherent uncertainty between these processes. The current system works fine if the traffic in the airspace is light. But the problems occur during peak hour situations. When there are many aircraft in a small region, the size of the air sector becomes proportionally smaller. In the time scale,

---

\*Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. Internet: [nilim@eecs.berkeley.edu](mailto:nilim@eecs.berkeley.edu)

†Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. Internet: [elghaoui@eecs.berkeley.edu](mailto:elghaoui@eecs.berkeley.edu)

‡Department of Transportation Engineering, University of California, Berkeley, Internet: [mhansen@euler.berkeley.edu](mailto:mhansen@euler.berkeley.edu)

§Euro Control Experimental Centre, BP 15 91222 Brtigny-sur-Orge, Cedex, France. Internet: [vu.duong@eurocontrol.fr](mailto:vu.duong@eurocontrol.fr)

the size of sector can go down to 5 minutes. If the storms are predicted to happen in these regions, the situation worsens even further.

Our approach involves shifting the Air traffic management system to trajectory based from the airspace based. There will be only one controller for each aircraft from gate to gate. Here, we will mainly focus on the en-route portion of the flight for a aircraft where the nominal path of the aircraft may be obstructed by a bad weather. Our work is carried through in three major phases,

1. In the first phase, we propose a dynamic routing strategy of a single aircraft that will minimize the fuel/time cost or maximize profit/safety for an individual aircraft, whose nominal route is predicted to be impeded by storms. We use a Markov decision process and a dynamic programming algorithm that provides an optimal routing strategy.
2. In the second phase, we propose an algorithm for dynamic routing whose solution is robust with respect to the estimation errors of the storm probabilities. Based on an exact linear programming (LP) formulation of the dynamic programming algorithm proposed in the first phase, we add a further generalization in the LP formulation: we assume that the transition probabilities are unknown but bounded within a convex set. Our algorithm optimizes the performance of the system, given there are errors in the estimation of probabilities of storms.
3. In the final phase, we propose a system level optimized solution. We use combinatorial auction techniques to allocate the resources (airspace) to each aircraft. We will also look at the complexity issues of our algorithm so that our solution can be used in real time.

In this paper, we mainly focus on the first phase of the work and present the Markov decision processes and dynamic programming algorithm for optimal routing strategy of an aircraft under weather uncertainty.

## The motivation of the problem

Consider a two dimensional flight plan of an aircraft. We are interested in finding the optimal path in the TMA/En-route portion, as the rest is taken care by the control tower. There are inherent uncertainties in the TMA/En-route portion of the flight. We could address the problem as in decentralized fashion, as a large-scale stochastic optimization problem.

Consider the following scenario.

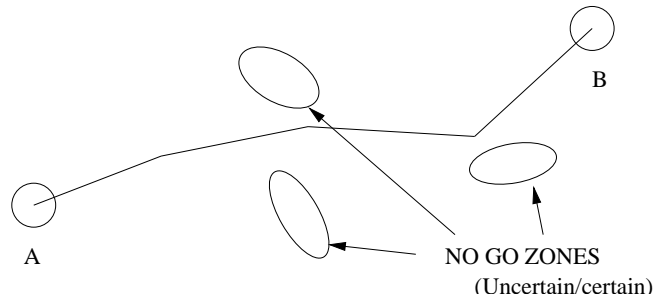


Figure 1: 2-D flight plan of an aircraft

An aircraft is flying from point A to point B. There are many obstacles that might interfere with the shortest possible route. These obstacles can be both stochastic and deterministic in nature. Examples of the stochastic “no zones” include storm zones, intersections of the flight plan of other aircraft, strong wind zones and the deterministic “no zones” include military and national security airspace. Given those constraints, the optimal route of the aircraft is the one which provides the minimal cost( time, fuel) while avoiding all of the obstacles.

In the optimization problem, it is easier to deal with the deterministic “no zones”, as we can assign them an infinite cost if aircraft were to penetrate through the zones. But the problem becomes more interesting if the “no zones” are stochastic. Various weather teams produce predictions that some zones in the airspace will be unusable in certain time and their predictions are dynamically updated with time. In the current practice, those stochastic zones are assumed to be unusable, and solution proceeds as if they are deterministic constraints. As those zones were just predicted to be of unusable with a certain probability, it often turns out that the zones were

perfectly usable. As the rerouting strategies do not use these resources, airspace resources are under-utilized, leading to congestion in the remaining airspace. In our proposed model, we will not exclude the zones which are predicted to be unusable (with some probability) at a certain time. Instead, we will assign some cost to those zones, a cost which will depend upon the probability of the prediction. Our solution will take into consideration the fact that there will more updates with the course of flight and recourse will be applied accordingly. We take a less conservative route in avoiding the bad weather zone where we take a risk in delay to attain a better expected cost instead of avoiding the bad weather zone deterministically. For this class of problems, we look for the “best policy”, not the “best path”. Determining the “best policy” is deciding where to go next given the currently available information. We consider the set of decisions facing an aircraft that starts moving towards the destination along a certain path, with the recourse option of choosing a new path whenever a new information is obtained.

### A simplified example to provide the intuition of the more general problem

In this section, we present a simplified example which lends to the insight of the more complex problem that we are trying to solve. Let’s consider a scenario where the aircraft is at the point O and it’s destination point is D. There is a prediction that there is a probability  $p$  that the region K, ( which is distance  $M$  away from O and with width  $L$  and obstructs the nominal route of the aircraft) might be unusable by the aircraft due to a storm.. When the aircraft comes  $s$  distance close to the region, it can be determined for sure whether it can go through it or not. The problem is to figure out the routing strategy of the aircraft in this situation.

The decision variable that we need to compute is  $\phi$ , which is the angle that the aircraft should proceed until it knows the condition of K for sure. When it reaches the point A, where it knows the condition of K with probability 1, a decision can be made. If K is usable, then it will follow the path AD and reached it’s destination. Otherwise, it will take the path AB and then BD. The expected length of this strategy is  $l_1 + p(m_1 + m_2) + l_2$ . Or given,  $M, N, s, p$ ,

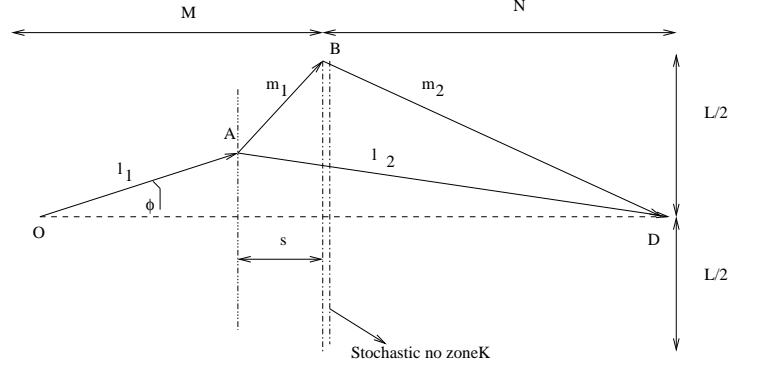


Figure 2: 2-D flight plan of an aircraft

Our objective function would be,

$$\min_{\phi} \left\{ \frac{M-s}{\cos \phi} + (1-p) \frac{N+s}{\cos \left( \frac{(N+s) \cos(\phi)}{\sqrt{(M+s)^2 \cos^2(\phi) + (N-s)^2 \sin^2(\phi)}} \right)} + p \left[ \sqrt{\left(\frac{L}{2}\right)^2 + d^2} + \sqrt{s^2 + \left(\left(\frac{L}{2}\right)^2 - (M-s) \tan(\phi)\right)} \right] \right\}$$

where the constraint set is,  $0 \leq \phi \leq 90^0$

This optimization problem can be solved easily and when  $p = 0$ , the value of  $\phi$  is 0, when  $p = 1$ , the value of  $\phi$  is  $\tan^{-1} \frac{L}{2M}$ . We can see that, with the traditional method, the path OBD is always chosen even if the value of  $p$  is small. In contrast, our strategy takes a chance to go towards the bad weather region and hence incurs a lower expected cost.

Though the toy problem described above gives us most of the insights about our problem, some important extensions are needed to address a practical situation.

1. The probability of a bad weather is not constant with time, instead it varies with time and dynamically updated. A probabilistic model is needed to simulate the weather change/prediction update with the progress of time. So instead of taking recourse at one stage of time, it has to be in multiple times, that leads us to form the problem in a dynamic programming setting
2. In the real life problem, there will be more than one block which will make the formulation more complicated because the solution of avoiding one block will effect the solution of avoiding the other

blocks.

### Weather uncertainty model

Various weather teams provide the probability of a storm at a particular place at a particular time. The weather information is updated in about every 15 minutes. The further away the prediction is from the event, the more unreliable it becomes and vice versa. We can discretize the time in a number of 15 minutes time intervals. From the weather science, we can have the probability “Pr” or “Qr” of a particular region such that Pr=probability (there will be a storm in that region in the next 15 minutes time interval/there is a storm in the current time in the region ), or Qr=probability (there will be no storm in that region in the next 15 minutes time interval/there is no storm in the current time in the region)

It is also realistic to assume that the aircraft has a perfect knowledge about the regions which are 15 minutes (15 times the velocity of the aircraft provides the distance) away from it. Whenever an aircraft is in 15 minutes away from the storm region, the pilot knows for sure whether there is a storm in that region or not. The probability of storm at a particular region is time varying and takes a value 0 or 1 when it reaches 15 minutes away from the region.

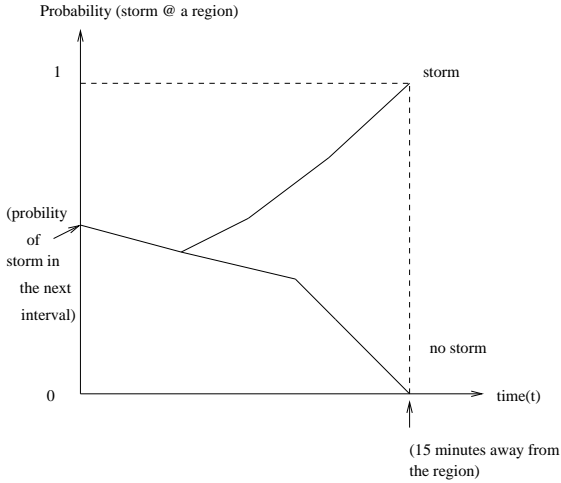


Figure 3: Variations of the probability of storm with time

We can discretize time as  $1, 2, \dots, n$  stages, where “1”

corresponds to the time 0 – 15 minutes from the current time, “2” corresponds to the time 15 – 30 minutes from the current time and so on. Let,  $T$  is the time required to go from the origin to destination in the worst possible routes (where the worst possible route is defined as the routing strategy where nominal path is followed assuming no storm and applying recourse just before the storm zone to avoid the storm). No policy will take more time than  $T$ . Total number of stages can be calculated by the following formula,

$$n = \frac{T - \text{mod}(\frac{T}{15})}{15} + 1 \tag{1}$$

Suppose, there are  $1, 2, \dots, m$  storms that are predicted to happen that might force the aircraft to deviate from its nominal path. We can define state “1” corresponding to the state as having storm in a region at a particular stage and state “0” corresponding to state as having no storm at a particular stage in that region. As we know the status of any storm in the time interval of 0 – 15 minutes (stage 1), we can assign 0(1) to every storm at the stage 1. Again we also know the conditional probability of having(not having) a storm in a region in the next 15 minutes time interval (stage 2), given there is a storm(no storm) in the current time (stage 1) in the region. If we assume that the probability of storm in a particular zone varies in a Markovian fashion, we can represent the transition model of each storm in the following manner,

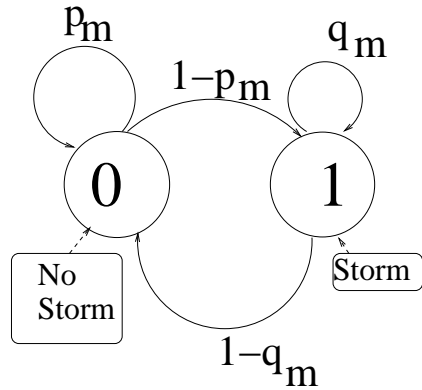


Figure 4: Transition probabilities for storm “m”

We can define,  $p_1$  is the probability of no storm in the next stage if there is no storm in the current stage in a particular region and  $q_1$  is the probability of having

a storm in the next stage if there is a storm in the current stage in that particular region..

Similarly, we can define  $p_2, q_2, p_3, q_3, p_4, q_4, \dots, p_m, q_m$ . If there are  $m$  storms, it will be a  $2^m$  state Markov chain. A  $2^m$  tuple vector can describe the storm situation completely, i.e., [1000..0] denotes that there is a storm in zone '1' and the rest of the predicted bad weather zones are storm free. We can define  $\mathbf{1} = [0000..0]$ ,  $\mathbf{2} = [1000..0]$ ,  $\mathbf{3} = [0100..0]$ , .....,  $\mathbf{M} = [1111..1]$  ( $2^m$ th state). The transition matrix  $P$  for the Markov chain is  $2^m \times 2^m$  matrix whose  $(i, j)$ th entry  $p_{ij}$  denotes the probability of  $j$ th state in the next stage, given the current state is  $i$ . The transition matrix can be defined as follows,

$$P = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{12^m} \\ \dots & \dots & \dots & \dots \\ p_{2^m 1} & p_{2^m 2} & \dots & p_{2^m 2^m} \end{pmatrix} \quad (2)$$

If the current state  $X_1 = i$  is given, the probability that the  $k$ th state  $X_k = j$  can be determined as follows,

$$p(X_k = j / X_1 = i) = P_{ij}^k \quad (3)$$

Which is the  $(i, j)$  the component of  $P^k$  matrix.

The  $P$  matrix is an input in our algorithm. We can form a  $P$  matrix in a storm situation where the storms are moving and are not probabilistically independent. But for the demonstration in this paper, we assume that there is no movement of storms, i.e., the zones where no storms are predicted to happen are assumed to remain as perfectly usable by the aircraft over all time and storms don't expand or contract over time (relaxation of this assumption will increase the size of the probability matrix). The following example will illustrate the weather model used in the demonstration. If there are 2 storms, there will be  $2^2$  or 4 states. The transition matrix  $P$  will be  $4 \times 4$ . The states will be,

- $\mathbf{1} = [11]$  (Both zones have storm)
- $\mathbf{2} = [10]$  (Storm in zone 1, no storm in zone 2)
- $\mathbf{3} = [01]$  (No storm in zone 1, storm in zone 2)
- $\mathbf{4} = [00]$  (No storm in either of the zones)

where  $P$  is the following matrix,

$$\begin{pmatrix} q_1 q_2 & q_1(1-q_2) & q_2(1-q_1) & (1-q_1)(1-q_2) \\ q_1(1-p_2) & q_1 p_2 & (1-q_1)(1-p_2) & (1-q_1)p_2 \\ (1-p_1) & (1-p_1)(1-q_2) & p_1 q_2 & p_1(1-q_2) \\ (1-p_1)(1-q_2) & (1-p_1)p_2 & p_1(1-p_2) & p_1 p_2 \end{pmatrix}$$

## Graphical representation of the airspace

Our problem is to find out the direction of the aircraft in a way that will take into account the fact that more information will be received in the course of the flight. The aircraft will stick to the direction till it receives another weather update. In the previous section, we have already discretized the time and divided the decision horizon in a number of stages. An aircraft will not receive any information until it goes to another stage. We simulate the variation of the probability of storm by using our Markovian model. The direction to be taken in each stage is a continuous problem, which is hard to solve. As it is a common practice in ATC process that aircraft follow some fixed way points, it is an acceptable formulation to discretize the airspace. Let's 'A' is the origin point and 'B' is the destination point. The region described by the rectangle LPNM contains all the storms and the origin and destination points (figure 6). LPNM defines the airspace that contains the efficient routing of the aircraft. We can discretize the airspace LPNM by a rectangular gridding system. We can think each grid point as a way point through which an aircraft can fly. When the aircraft is in the point 'A', the solution of the problem is the grid point in which the aircraft will reach in 15 minutes, before the next update. Once, the point is decided, the aircraft can just fly straight to the point and the vector direction of the aircraft can be determined. Reducing the distance between two grid points result in better solution, but it incurs more expensive calculation. Reducing the distance between two grid points to zero gives the continuous solution.

## Reducing the search space

If there are  $m$  storms, we will have  $N = 2^m$  states. For each of the state  $i = 1, 2, 3, \dots, N$ , we can easily determine the deterministic shortest path from the origin 'A' to the destination 'B'. Let's define  $SP_i$  as the shortest path corresponding to deterministic state  $i$ . If  $W_{i,j}$  is region formed by the  $SP_i$  and  $SP_j$  as the boundaries.  $R = \bigcup_{i,j} W_{i,j}$  is the region which contains all

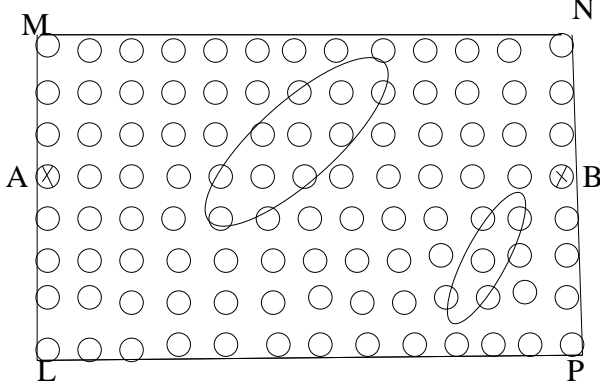


Figure 5: Rectangular grid of the airspace

the possible shortest path.

The search space can be reduced based on the following fact. For all  $i = 1, 2, \dots, N$ , let  $SP_i$  denotes the corresponding shortest paths. If  $R$  is the region formed by the convex combination of all the points on  $SP_i \forall i = 1, 2, \dots, N$ , we can run our algorithm in the region  $R$  only, which contains the stochastic shortest path with multiple recourse. Complexity issues of these kinds of problems are described in [9], [10], [11].

### Stochastic Dynamic Programming

The goal of a stochastic dynamic programming is to choose a policy that minimizes (maximizes) the expected value of cost (reward) earned over a given finite time span. If there are a finite  $n$  stages, the problem can be solved recursively [4]. The dynamic of the discrete time system can be defines as,

$$x_{k+1} = f_k(x_k, u_k, w_k), \forall k = 0, 1, 2, \dots, n-1 \quad (4)$$

Where,  $x_k$  is the state of the system at the time  $k$ ,  $u_k$  is the the control applied at the system at the time  $k$  and  $w_k$  is the randomness of the system at the time  $k$ . The control  $u_k$  is constrained to take values from a given nonempty subset  $u_k(x_k)$ . The random disturbance  $w_k$  is characterized by a probability measure. We can consider the class of control law that consist of a finite sequence of functions,  $\mu = [\mu_0, \mu_1, \dots, \mu_{n-1}]$ , where  $u_k(x_k) \in u_k(x_k) \forall x_k$ . Such control law will be termed as admissible.

Given an initial state  $x_0$ , the problem is to find an admissible control law  $\mu = [\mu_0, \mu_1, \dots, \mu_{n-1}]$  that minimizes cost function. If  $g_i(x_i)$  is the cost associated with

the stage  $x$  and there are  $i$  stages to go. Expected cost function can be defined in the following way [5] [6] [7],

$$J_\mu(x_0) = E_{(w_k \forall k=0,1,2,\dots,n-1)}(g_n(x_n)) + \sum_{k=0}^{n-1} g_k(x_k, \mu_k(x_k), w_k) \quad (5)$$

Subject to constraints,

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k) \forall k = 0, 1, 2, \dots, n-1 \quad (6)$$

For a system in which the uncertainty can be described in a Markovian manner, the control equations can be written in a more convenient way [8]. If there are  $N$  states in the system and  $n$  stages to go, a policy that delivers the maximal(minimal) value is called an optimal policy.  $v(i, n)$  is the value function where  $i$  is the current state of the system and  $n$  is the stages to go.  $v(i, n)$  is unique but there can be more than one policy that can provide this value. Dynamic programming is an inductive approach. An optimal policy for a process whose current state is  $i$  and  $n$  stages to go, must make use of optimal policies for the system with  $n-1$  step remaining. If after the initial decision and transition, the system is in state  $j$ , the original optimal policy now constitute an optimal policy for the system with the initial state  $j$  and  $(n-2)$  stages remaining. There are  $A_i$  alternatives out of state  $i$  on the first transition. If alternative  $k$  is predescribed, the expected gain on the initial transitions would be  $q_i^k$  and the probability of moving to state  $j$  from state  $i$  would be  $p_{ij}^k$ . If the system does in fact move to state  $j$ , from the principle of optimality, the total expected cost over the optimal policy over the last  $(n-1)$  stages would be  $v(j, n-1)$ . Hence the total expected cost is  $q_i^k + \sum_{j=1}^N p_{ij}^k v(j, n-1)$ . It follows that  $(n-1)$  satisfies the recursive equation,

$$v(i, n) = \min_{(1 \leq k \leq A_i)} [q_i^k + \sum_{j=1}^N p_{ij}^k v(j, n-1)] \quad (7)$$

### Markov Decision Process Algorithm for dynamic routing of aircraft under uncertainty

We are mainly concerned with the enroute part of the aircraft flight where the velocity remains almost constant (say  $V$ ). So we can consider the velocity to be constant. In this way, minimizing expected delay is same as minimizing expected distance to be traveled to go to the destination from the origin. Our objective function is the expected distance to be traveled. Decision variables are the nodes to go at the end of each stage time till it reaches the destination. We define  $\delta \approx 15V$  as distance from the storm from where the pilot knows for sure whether there is a storm or not. As previously described, the aircraft is at origin 'A' and the destination is 'B'. There are  $m$  regions, labeled  $k_1, \dots, k_m$ , which are predicted to have storm so that it might obstruct the nominal path of the aircraft. We get weather update in every 15 minutes. The transition matrix  $P$  is given. Our algorithm provides the routing strategy of the aircraft in order to obtain the minimum expected distance. The algorithm is described below.

**Step 1:** Calculate the the number of stages  $n$ , where  $n = \frac{T - \text{mod}(\frac{T}{15})}{15} + 1$  and  $T$  is the time required in the worst possible route.

**Step 2:** Discretize the airspace with a rectangular grid (of spacing  $\lll 15$  min).

**Step 3:** Prune the search space which is obtained by the convex combination of all the shortest path routes corresponding to different states.

**Step 4:** Determine the points that can be reached in the next 15 minutes. This can be approximately calculated if we draw an annular region with  $15 \times V \pm \epsilon$  as radii, with a predefined angle  $\theta$  and checking which grid points fall in the region. For the first stage, the angle can be obtained from the orientation of the storms. For the next stages, the angle  $\theta$  could be the maximum permissible turning angle of the aircraft.

Let,  $(x, y)$  is the coordinates of 'A' and  $(z_1, w_1)(z_2, w_2)(z_3, w_3), \dots, (z_a, w_a)$  are  $a$  such points that can be reached at the end of first stage.

**Step 5:** Assign appropriate costs. Costs in our algorithm should be such that they enforce the solution

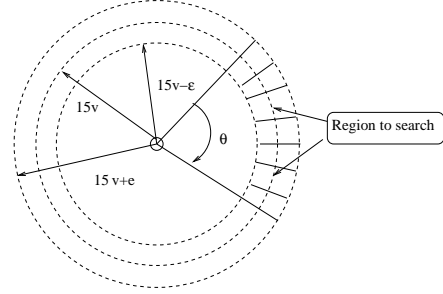


Figure 6: determining grid points that can be reached in the next stage

will take a path through the predicted storm zone if there is no storm and avoid that if there is a storm. Cost from going to a point from a point is a function of the state of storm and the Euclidean distance between the two points.

Define  $c(i, x, y, z_j, w_j)$  as cost to go from  $(x, y)$  to  $(z_j, w_j)$  if the storm state is  $i$ .

Starting from origin  $(x, y)$

*Check:*

For  $j = 1, 2, \dots, a$

If  $\{(z_j, w_j) \in k_1 \text{ or } k_2 \text{ or } \dots k_m\}$  or  $\{\text{the straight line } (\lambda(x, y) + (1 - \lambda)(z_j, w_j) \text{ and } 0 \leq \lambda \leq 1) \text{ connecting } (x, y) \text{ and the point } (z_j, w_j) \text{ cut any of the predicted storm zone}\}$  and  $\{\text{state of the intersected zone corresponds to a storm at that particular zone}\}$

$c(i, x, y, z_j, w_j) = \{\text{Very high cost}\}$

else

$c(i, x, y, z_j, w_j) = \|(x, y) - (z_j, w_j)\| \approx 15V$

endif

endfor

Proceeding this cost assignment till it reaches the destination point

**Step 6:** Defining the value function for our dynamic program,

$v(i, x, y, n) =$  Expected minimum distance to go if the aircraft is at the initial point  $(x, y)$  with the initial state  $i$  and it has atmost  $n$  stages to go to reach the destination point 'B'.

**Step 7:** Assigning the boundary value to the value functions that guarantee the desired solution.

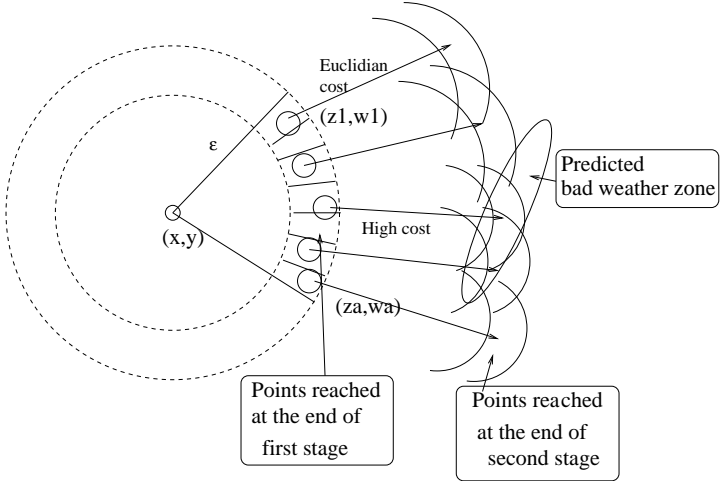


Figure 7: Cost assignment

*Case 1:* We have to make sure that we always get a complete path (path containing the origin and the destination) as a solution. If  $(p, q)$  are the coordinates of the destination point, the conditions below guarantee that the solution will provide a complete path.

For any state  $i$ ,  
if  $\{u=p\}$  and  $\{v=q\}$   
 $v(i, u, v, 0) = 0$   
else  
 $v(i, u, v, 0) = \infty$   
endif

*Case 2:* In the iteration process we need to put values of the value function for the points which are less than  $15V$  apart from the destination point.

For any  $n$ ,  
for any points  $(l, m)$  such that  $\|(l, m) - (p, q)\| \leq 15V$   
if  $\{$  There is no storm zone in the straight line  $(\lambda(l, m) + (1 - \lambda)(p, q))$  and  $0 \leq \lambda \leq 1$   $\}$   
 $v(i, l, m, n) = \|(l, m) - (p, q)\|$ , for any  $i$   
elseif  $\{$  for all  $i$  corresponds to the state with no storm at that zone  $\}$   
 $v(i, l, m, n) = \|(l, m) - (p, q)\|$   
else  
 $v(i, l, m, n) = \infty$   
endif

**Step 8:** Implement the recursive equations,

$$v(i, x, y, n) = \min_{(z_1, w_1), \dots, (z_a, w_a)} V \quad (8)$$

where,

$$V = \begin{pmatrix} c(i, x, y, z_1, w_1) + \sum_{j=1}^{2^m} p_{ij} v(j, z_1, w_1, n-1) \\ c(i, x, y, z_2, w_2) + \sum_{j=1}^{2^m} p_{ij} v(j, z_2, w_2, n-1) \\ \dots \\ c(i, x, y, z_a, w_a) + \sum_{j=1}^{2^m} p_{ij} v(j, z_a, w_a, n-1) \end{pmatrix}$$

Here we don't know the values of  $v(j, z_1, w_1, n-1)$ ,  $v(j, z_2, w_2, n-1)$ , ...,  $v(j, z_a, w_a, n-1)$ . To obtain those values, we need other recursive relations. Our calculation moves forward till we reach the boundary conditions and then we back track and calculate all of the value functions. From this value functions and we can trace the minimum expected distance path. The aircraft will keep continue proceeding according to the solution till a new update is obtained. After receiving a new weather update, the aircraft can run our model again to obtain a new updated route with the new input (position and vector direction of the aircraft at that time, the new updated weather). In this way, an aircraft will follow a trajectory which is updated in every 15 minutes. The aircraft avoids the bad weather zone if there is actually a storm, but takes a less circuitous route if there is no storm. As a result, the expected distance traveled is minimized.

## Simulation

We have implemented the algorithm in MATLAB which was executed on the 'ultra 5 sun microsystem machine', where we ran our program in some simplified situations to obtain the optimum dynamic routing of an aircraft under uncertainty. We have mainly considered two simplified scenarios. In the scenario 1, an aircraft's current position is at  $[0, 0]^T$  and the destination is at  $[360, 0]^T$ . (All the units are in n.mi). The velocity of the aircraft is 480 n.mi/hour. There is a prediction that a storm might obstruct it's nominal flight path. The storm zone is a rectangular space with the corner points at  $[192, -96]^T$ ,  $[200, -96]^T$ ,  $[200, 96]^T$  and  $[192, 96]^T$ . We are assuming that the weather information of the portion of the airspace that can be reached in 15 minutes is deterministic. The probability of storm propagates in a Markovian fashion with time. If there is a storm currently, the probability that the storm will stay there in the next 15 minutes is .80 (and consequently, there will not be any storm with a



probability .20). Moreover, if there is no storm currently, the probability that there will be a storm in the next 15 minutes is .25 (and consequently, there will not be any storm with a probability .75). The traditional method of handling this kind of situations is either of the following ways,

1. **Traditional Strategy 1 (TS1):** Flying in the direction that avoids the storm zone completely
2. **Traditional Strategy 2 (TS2):** Flying along the nominal path till it reaches sufficiently close to the zone such that the storm situation is known deterministically and taking a circuitous route if there is a storm or flying along the nominal path to reach to the destination if there is no storm.

If the aircraft follow TS1, the traveled distance will be 417.18 n.mi. If it follows TS2, the expected traveled distance will be 393.6 n.mi.. Our algorithm will provide the dynamic routing strategy in which the aircraft will fly with an angle  $14.9^\circ$  till it gets a further update. In the next update, it knows the storm situation with certainty and avoid the zone if there is a storm or fly directly to the destination in case of good weather. In this way, expected flight length will be 379.2 n.mi.. We introduce a performance metric “Improvement Measure (I.M.) of Strategy ‘A’ over ‘B’ ”, which is the percentage of the maximum possible improvement gained by using strategy ‘A’ instead of strategy ‘B’. The maximum possible improvement of any routing strategy is the difference between expected distance traveled in that strategy and the nominal distance (which is the Euclidean distance between the origin and the destination). As the minimum possible travel distance is 360 n.mi., the maximum possible improvement over TS1 is  $417.18 - 360 = 57.18$  n.mi.. The Improvement Measure (I.M.) of our strategy over TS1 is  $\frac{417.18 - 379.2}{417.18 - 360} \times 100$  or 66.42 percent. In addition, the possible improvement over TS2 is  $393.6 - 360 = 33.68$  units. The Improvement Measure (I.M.) of our strategy over TS2 is  $\frac{393.6 - 379.2}{393.6 - 360} \times 100$  or 42.76 percent.

In the scenario 2, the storm is farther away from the current position. The aircraft’s current position is at  $[0, 0]^T$ , the destination is at  $[480, 0]^T$  and the velocity is 480 n.mi/hour. The storm zone is a rectangular space with the corner points at  $[304, -120]^T$ ,  $[320, -120]^T$ ,  $[320, 120]^T$  and  $[304, 120]^T$ .

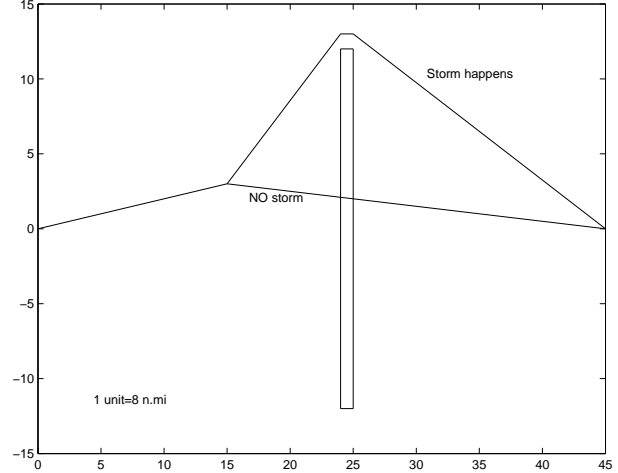


Figure 8: Dynamic Routing Strategy

The probability of storm propagates in the same manner as described above. In this scenario, just avoiding the bad zone (TS1), the traveled distance will be 548.82 n.mi.. In TS2, the expected traveled distance will be 542 n.mi.. The minimum possible travel distance is 480 n.mi.. The maximum possible improvement over TS1 is 62.82 n.mi.. In our strategy, the expected distance is 511.12 n.mi.. The Improvement Measure (I.M.) of our strategy over TS1 is 54.78 percent. Moreover, the possible improvement over TS2 is 30.88 n.mi.. The Improvement Measure (I.M.) of our strategy over TS2 is 49.81 percent. The result can be summarized in the following way,

	I.M. of our model over TS1	I.M. of our model over TS2
Scenario 1 (3-Stage problem)	66.42 %	42.76%
Scenario 2 (4-Stage problem)	54.78 %	49.81 %

Table 1: Improvement comparisons.

In general, as we use the updated information in our optimization process, our strategy should provide higher Improvement Measure (I.M.) over traditional strategies when the number of stages, size of the storms and the number of the storms increase.

## Conclusion

Our ongoing work in developing a tool to dynamically route a single aircraft under uncertain weather is much needed in the aviation community in order to reduce the delay of the ATC system. Two way advantages that can be derived with the implementation of our tool,

1. Less circuitous route for any aircraft which is subjected to bad weather.
2. Less overloading of aircraft in the neighboring sectors of the predicted storm zones.

The complexity of the computation depends on the origin-destination pair, size and location of the storms, level of discretization and the stages of information updates. The complexity of the algorithm is exponential with the number of storm. The probabilities that are generated by various weather prediction agencies are often incorrect. In future, we are proposing a robust dynamic routing strategy where a best solution can be obtained even when there are errors in the estimation of storm probabilities. We are also working to have a decentralized solution which can be formulated as a combinatorial auction process. We will investigate how an auction process, where each airline company bids competitively for the airspace, could solve the problem.

## References

- [1] Dimitris Bertsimas, Sarah patterson. The air Traffic flow management problem with enroute capacities. *Operations Research*, 46:406-422,1998.
- [2] J. Goodhart. Ph.D thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley, 1999
- [3] M. Bielli, G. Calicchio, B. Nicoletti and S. Riccardelli. The air traffic flow control problem as an application of network theory. *Computation and Operations Research*, 9:265-278,1982.
- [4] Stuart Dreyfus *The art and theory of dynamic programming* , Academic Press, New York, 1977
- [5] M. Vidyasagar. *A theory of learning and generalization: with applications to neural networks and control systems*. Springer- Verlag, London, 1997.
- [6] D. P. Bertsekas. *Dynamic Programming and Stochastic Control*, Academic Press, New York ,1976.
- [7] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, 1987.
- [8] Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley Inter-Science, New York, 1994.
- [9] G. H. Polychronopoulos and J. N. Tsitsiklis. Stochastic Shortest Path Problems with Recourse, *Networks*, 27:133-143, 1996.
- [10] H.N. Psaraftis and J.N. Tsitsiklis, Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs, *Operations Research*, 41:91-101 1,1993
- [11] D.P. Bertsekas and J.N. Tsitsiklis, An Analysis of Stochastic Shortest Path Problems, *Mathematics of Operations Research*, 16:580-595, 1991.
- [12] C. Burlingame, A. Boyd, K. Lindsey. Traffic flow management modeling with the Time Assignment Model, MP 93W0000036, *The MITRE CORPORATION*, Virginia, January, 1994
- [13] B. Lucio. Multilevel Approach to ATC problems: On-line Strategy Control of Flights. *International Journal of Systems Science*, 21:1515-1527, 1990
- [14] H. Marcia. The Potential of Network Flow Models for Managing Air Traffic, , MP 94W0000058, *The MITRE CORPORATION*, Virginia, June, 1995

## Author Biographies

Arnab Nilim is a Ph.D. student in the Department of Electrical Engineering and Computer Sciences (EECS) at the University of California Berkeley. At Berkeley, he has conducted numerous studies in aviation-related problems and published research papers in a variety of conferences (AIAA, CDC, INFORMS).

Laurent El Ghaoui is an Associate Professor in the Department of Electrical Engineering and Computer Sci-

ences(EECS) at the University of California Berkeley.

Mark Hansen is an Associate Professor in the Department of Transportation Engineering at the University of California Berkeley.

Vu Duong is the Head of project at Euro Control, Experimental Centre.

### **Acknowledgment**

The authors would like to thank Jianhai Hu, Anuj Puri, Professor Stuart Dreyfus, Professor Pravin Variya and Professor Christos Papadimitriou for their valuable suggestions in the course of this research.

Research supported in part by NEXTOR under grant NGCDM 2-4909 and Euro Control.