5th USA/Europe ATM 2003 R&D Seminar, Budapest, Hungary, 23-27 June 2003.

Real-Time Conflict-Free Trajectory Optimization

Matt R. Jardin, Ph.D., NASA Ames Research Center, Moffett Field, California

Abstract

The increasing amount of air traffic along the existing jet route structure has led to inefficiencies and chronic congestion in enroute airspace. Analysis suggests that enroute capacity may be increased by a factor of five, and that direct operating costs might be reduced by about 4.5% (about $500 Million per year) if aircraft were permitted to fly unconstrained optimal wind (free flight) routes. The multi-aircraft optimization problem is characterized by high system complexity. This is primarily because of the large number of inter-aircraft separation constraints. Several approximate solution approaches have been tried in prior research, but none has yet achieved practical conflict free trajectory optimization. The main challenges are in the areas of computationally efficient wind-optimal routing, aircraft conflict detection, and optimal conflict resolution. Contributions are presented in each of these areas. High-fidelity simulation results are presented to demonstrate that near-optimal route optimization can be achieved for current air traffic densities in under a minute on typical computer hardware.

Introduction

The objective of this research is to develop a method to optimize and deconflict enroute trajectories of all aircraft on a continental scale in real-time (about one minute or less). The method is based on the real-time strategic optimization of aircraft trajectories. The term “strategic” indicates the fact that conflict-free optimal trajectories are computed on time scales of thirty minutes or more into the future. In between optimization cycles, 4-dimensional (4D) control would presumably be used to cause aircraft to follow the optimal trajectories, though 4D control is only recommended, not required. This is in contrast to tactical concepts where aircraft would resolve conflicts as they arose, on time scales of about ten or fifteen minutes, without considering the longer-range consequences on trajectory optimization.

This same basic 4D concept has been studied in various forms throughout the past three decades [1-6], but the ability to compute optimal conflict free trajectories for thousands of aircraft in real time has remained an elusive goal. Existing algorithms are either too slow, or they produce oversimplified solutions.

An analysis of enroute traffic data in the United States has shown that a reduction of about 4.5% in total flight time (or fuel use) would be achievable if aircraft were allowed to fly optimal wind routes. This is a reduction of about 500 hours per day in flight time, which translates to nearly $1 million per day ($360 million annually). In addition to direct cost savings, freeing aircraft from the confines of the structured routing system may enhance safety by spreading aircraft over more airspace and thereby lessening the chance for collisions.

In the next section, a brief synopsis of aircraft trajectory optimization is presented to show why the multi-aircraft problem is so complex and to justify the search for approximate solution methods. This is followed by a section to introduce the concept for achieving multi-aircraft trajectory optimization in real-time. Details of the key algorithms comprising this concept are then presented, followed by a section on analysis and simulation results.

Background: Optimal Air Traffic Control

The strategic trajectory optimization problem for a single aircraft may be stated as a cost function minimization problem with dynamic constraints and constraints on initial and final aircraft states

$$J_i = \int_0^{t_{fi}} L_i(\bar{x}_i, t)\,dt \qquad \begin{cases} \bar{x}_i(0) - \bar{x}_{0i} = 0 \\ \bar{x}_i(t_{fi}) - \bar{x}_{fi} = 0 \\ \dot{\bar{x}}_i = f(\bar{x}_i, t) \end{cases} \tag{1}$$

In eq. (1), $\bar{x}_i$ is the state vector for aircraft i, J_i is the integrated trajectory cost, and the objective function, $L_i(\bar{x}, t)$, is usually defined as the time rate of change of the direct operating cost (DOC), a linear combination of fuel and time costs for commercial aircraft operation.

The single-aircraft optimization problem in eq. (1) is usually decoupled into separate vertical and horizontal trajectory optimization problems. The vertical problem is recast as a convex optimization problem in an energy state form and solved for the optimal speed and altitude profile vs. path distance for the case of zero winds [7,8]. The primary result is that optimal long-range vertical profiles for commercial jet transport aircraft consist of optimal ascent and descent segments connected by a long cruise-climb or step-climb segment.

Optimal horizontal routes are not as easy to compute because the variations in the wind field lead to a non-convex nonlinear optimization problem with potentially many regions of local minima.

The multi-aircraft air traffic control cost function may be written as the sum of N single-aircraft cost functions as follows

$$J_{\text{tot}} = \sum_{i=1}^{N} J_i \qquad (2)$$

with inequality constraints on aircraft separation

$$\Delta d_{ij} \geq D_{\min} \qquad \begin{cases} 1 \leq i \leq N \\ 1 \leq j \leq N \\ i \neq j \end{cases} \qquad (3)$$

where Δd_{ij} is the distance between the i^{th} and j^{th} aircraft, and $D_{\min}$ is the minimum allowable separation (e.g. 5 nautical miles).

Because of these constraints, it can be shown that one must examine the local optimum solutions within N_o possible aircraft orderings, where N_o is given by

$$N_o = 2^{((N(N-1))/2)} \qquad (4)$$

For 500 aircraft (the maximum number of aircraft operating at the busiest flight level over the United States at any instant in time) eq. (4) says that nearly 10^{37500} possible aircraft orderings would need to be examined to guarantee the optimal solution would be found. Most of the possible orders are not feasible, but they all must be examined if an optimal solution is to be guaranteed. The complexity of this problem justifies an approximate solution approach. Many such approaches have been developed over the past three decades.

Some of the earliest research on 4D air traffic control concepts was initiated by Erzberger beginning in the 1970s [1]. Similar work was carried out by Boeing researchers to develop a complete 4D air traffic control system where aircraft would be given conflict-free 4D flight plans, and would then be required to use accurate 4D control to follow their assigned flight plans [2]. Under the European PHARE program, researchers examined concepts based on the use of 4D tubes, which were essentially 4D flight plans with error buffer regions [5]. Work has continued on various approaches to global air traffic control optimization both in the United States and in Europe [9-16].

In the current work, a strategic 4D approach is again taken, but attention is now given to computing efficient, realistic, conflict-free 4D trajectories for thousands of aircraft in real-time.

System Concept

The air traffic control optimization problem is characterized by high system complexity and is thought to be in the NP-hard class of problems [9,17]. This necessitates the use of approximate optimization solution approaches. The concept proposed here is to iteratively compute optimal trajectories for each aircraft in the system while holding previously-planned trajectories fixed. The initial trajectory for each aircraft is the unconstrained optimal wind trajectory, which is iteratively modified until all conflicts are resolved. This approach comes from the assumption that the enroute airspace is, and will continue to be, sparsely occupied. The sparse airspace assumption is investigated in the next subsection, followed by a more detailed description of the system concept and a semi-empirical analysis of the expected computational effort required by the iterative solution approach.

The Sparse Airspace Assumption

By modeling the number of conflicts that any one aircraft would encounter in a population of N aircraft as a binomial random variable, X, it can be shown that the expected total number of conflicts is approximated by

$$N_c \equiv \sum_{i=1}^{N} \frac{E[X]}{2} = \frac{N(N-1)p}{2} \qquad (5)$$

The probability, p, that an aircraft will encounter a conflict with any other aircraft may be determined by adjusting p to fit simulated conflict count results fed by real aircraft schedule data. Using data from such a study [18], the value of p for the currently structured airspace environment is estimated to be 9×10^{-6} so that the total number of expected conflicts for a population of 3000 aircraft is only about 40 (fig. 1). This corresponds to 1 expected conflict for every 75 aircraft at the current-day maximum aircraft density in a structured routing environment. The expected number of conflicts is even lower in a free routing environment because more airspace is utilized. The maximum number of aircraft is only expected to increase by a factor of 2 over the next 20 years so that the airspace will continue to be relatively sparsely occupied.

By the sparse airspace assumption, one would expect that the global cost function of eq. (2) would be flat near the

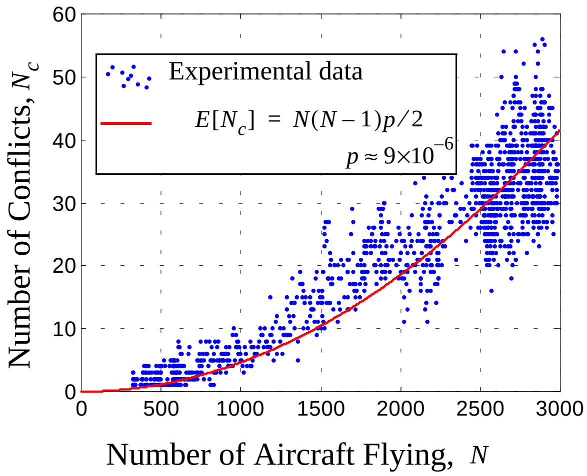

Figure 1. Conflict probability at current traffic levels.

optimal solution. In other words, there should be many solutions near the optimal solution with the same minimum cost. If this assumption is true, then an iterative approach should produce near-optimal solutions with relatively little computational effort. Since there are likely to be many feasible solutions near the optimal solution, this implies that one should not expend any great effort to find the absolute optimal solution.

High-Level Concept Description

The guiding principle is to keep the computations as simple as possible so that near-optimum performance might be achieved in real-time. One such simplification is to decouple the horizontal and vertical trajectory optimization functions as has been the practice for many years. Aircraft operators would initiate the optimization process by computing and requesting flight plans for a chosen altitude and airspeed as determined by the vertical profile optimization methods discussed previously. The following discussion is focused on the computation of conflict free optimal horizontal routes for the requested vertical profiles. The system is developed here for the 2D (horizontal plane) case, with an extrapolation of results to the 3D case made later in this paper.

Another simplification that is justified by the sparse airspace assumption is that trajectories may be sequentially optimized and deconflicted without modifying previously optimized trajectories. In general, this will not lead to the true optimal solution, but it can be shown to lead to nearly optimal solutions with high probability.

Proceeding sequentially through a list of all active aircraft (an active aircraft is one that is either currently in flight, or one that is scheduled to depart within some chosen window of time in the future), the first step for each aircraft is to compute the optimal wind route while ignoring any potential conflicts (fig. 2). The resulting

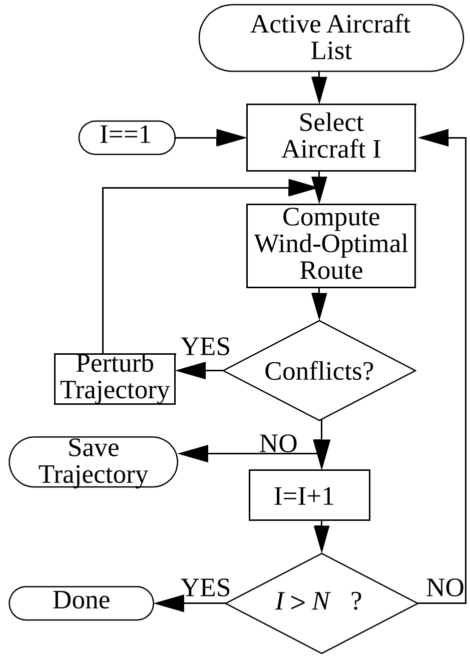

Figure 2. Optimization Algorithm Flowchart

optimal wind route is checked for conflicts with all previously deconflicted trajectories. If any conflicts are detected, a perturbed optimal wind route is computed and checked again for conflicts. The iterations continue until a conflict-free route is obtained, at which time computations begin for the next aircraft in the list. This process continues until optimal conflict-free trajectories have been computed for all aircraft. The first-scheduled-first-served method has been demonstrated to be generally equitable to all aircraft, though its performance relative to other ordering techniques has not been evaluated in this research.

Computational Performance Analysis

Estimates of the computational performance of the optimization algorithm are required to determine the time to complete all trajectory computations as a function of the number of aircraft. The number of conflicts is random, so the approach is to choose an appropriate model based on the physical nature of aircraft conflicts and then to choose model parameters that best fit empirical data. This is similar to what was done in the sparse airspace analysis. The model is developed for the horizontal plane (2D) and the results are extrapolated to 3D by appropriate scaling.

Assuming the expected number of computations for each aircraft is independent, the expected value of the total number of computations for all N aircraft is given by

$$E\left[\sum_{i=1}^{N} \xi_i\right] = \left[\sum_{i=1}^{N} E[R_i]\right] \cdot (E[\xi_{wo}] + E[\xi_{cd}]) \qquad (6)$$

where $E[\ ]$ is the expectation operator, ξ_i is the total number of computations for aircraft i, R_i is the number of conflict resolution iterations for aircraft i, ξ_{wo} is the number of computations required to compute a single optimal wind route, and ξ_{cd} is the number of computations required to check a single route for conflicts.

The expected number of conflict iterations, $E[R_i]$, will now be modeled based on physical considerations. The hypothesis is that for a sequential conflict resolution strategy, it is equally likely at each iteration that another conflict may be encountered. This suggests the use of the Geometric Random Variable (GRV).

If R_i is modeled as a GRV representing the number of iterations required to resolve all conflicts for the i^{th} aircraft, where each resolution iteration is considered to be an independent Bernoulli trial with probability P_i of being conflict free, then the probability mass function (pmf) for R_i is given by

$$p_{ik} = P_i(1-P_i)^{(k-1)} \begin{cases} i = 1, 2, \ldots, N \\ k = 1, 2, \ldots \end{cases} \qquad (7)$$

where p_{ik} is the probability of resolving a conflict in k iterations for the i^{th} aircraft. Typical values of P_i are close to unity so that the probability of finding a conflict-free solution during the first iteration is high, and the probability that a conflict-free trajectory won't be found until a later iteration decreases rapidly.

The conflict-free probability, P_i, is expected to decrease as the number of aircraft increases. In the interest of developing a simple model with a small number of parameters, a linear form for P_i is chosen

$$P_i = \frac{(C_0+1)}{C_1} - \frac{1}{C_1} i \qquad (8)$$

where C_0 and C_1 are parameters that are to be determined to best fit observed data.

Summing the expected number of conflict iterations over all N aircraft, simplifying, and making several function approximations leads to the following expression for the expected number of iterations as a function of the number of aircraft

$$Y_N \equiv \sum_{i=1}^{N} E[R_i] \approx C_1 \ln\left(\frac{C_0}{C_0 - N}\right) \qquad (9)$$

The parameters C_0 and C_1 may be chosen to best fit the observed number of resolution iterations as a function of the number of aircraft. The algorithm of fig. 2 may be run on a set of real aircraft schedule data, and the number of total resolution iterations may be recorded. An example result is depicted in fig. 3. The data for the simulation

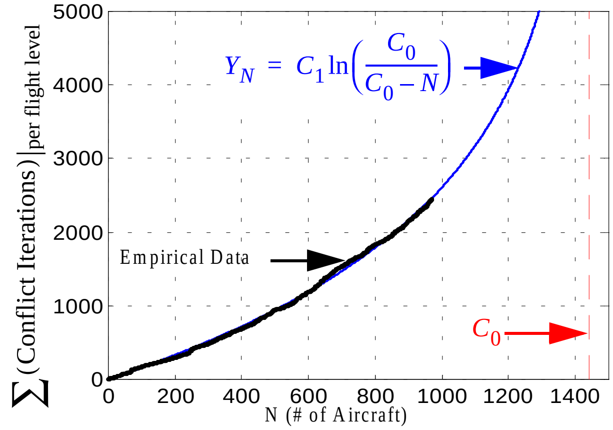

Figure 3. A semi-empirical model of the number of conflict iterations vs. the number of aircraft (2D).

were taken from the Enhanced Traffic Management System (ETMS) data feed for all aircraft in the continental United States domain at flight levels 330 and 350 on a particular day. The origin, destination, and scheduled departure times were extracted from the data and used to drive the simulation according to the algorithm in fig. 2.

The shape of the Y_N vs. N curve from the model matches observations well. Notice also that eq. (9) predicts a discontinuity when N equals C_0 so that C_0 may be considered as an estimate of the maximum airspace capacity for the specific iterative resolution algorithm used. This makes intuitive sense because there is a finite amount of airspace available. The capacity model of eq. (9) predicts just over 1,400 aircraft per flight level, which is much more reasonable. For reference, the maximum number of aircraft found at any flight level today is about 450, so the model predicts that capacity may be increased by a factor of 3.

Note that eq. (9) was developed without assuming that the model would be applied to the horizontal plane (2D) or to 3D airspace. Therefore, eq. (9) also applies to the 3D case. To extrapolate the empirically measured values of C_0 and C_1 to the 3D case, the 2D values must simply be multiplied by N_{FL}, the number of discrete flight levels in the 3D volume of space under consideration. The extrapolated 3D parameter values are given by

$$C_0\big|_{3D} = C_0\big|_{2D} \cdot N_{FL} \tag{10}$$

$$C_1\big|_{3D} = C_1\big|_{2D} \cdot N_{FL} \tag{11}$$

The extrapolated number of conflicts for the 3D case is easily shown to be

$$Y_N\big|_{3D} = Y_N\big|_{2D} \cdot N_{FL} \tag{12}$$

Component Algorithms

As shown in the previous section, the computations for optimal wind routing and for conflict detection are both multiplied by the number of resolution iterations. Since the number of iterations cannot be controlled, the way to minimize computational effort is to reduce the number of computations for optimal wind routing and conflict detection. The algorithms applied to problems in these areas will now be described.

Neighboring Optimal Wind Routing

The optimal wind routing problem is nonconvex, so practical solution approaches have usually been based upon randomized algorithms or discrete dynamic programming algorithms, with much of the work being proprietary to airlines or flight planning companies. Both of these types of approaches can be computationally intensive. For the current approach, an efficient optimal wind routing technique is required. It is also desired that the optimal wind routing algorithm may also be used to compute perturbed optimal wind routes for conflict resolution.

An algorithm called Neighboring Optimal Wind Routing (NOWR) has been developed for the efficient computation of optimal wind routes [19,21]. Based on neighboring optimal control techniques, NOWR is efficient to implement because it is a simple time-varying linear feedback algorithm where the feedback gains are computed such that the total flight time is minimized to second order (fig. 4). Perturbations in winds along the desired route are fed back to modify the aircraft heading to produce a minimum time route. The wind and wind shear are required at N_w discrete measurement points

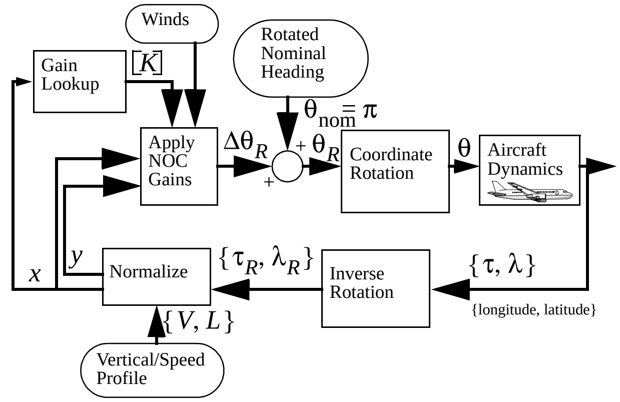

Figure 4. Neighboring Optimal Wind Routing.

along the nominal great-circle route between origin and destination. The winds may be obtained from the Rapid Update Cycle (RUC) or similar large-scale gridded wind product.

The NOWR feedback law to compute the minimum-time aircraft heading, θ_{opt}, is applied in a rotated and normalized coordinate frame where the nominal great-circle heading becomes $180°$ (π radians). The minimum-time heading in the rotated frame is given by

$$\theta_{opt} = \theta_{nom} + K_y \cdot y + \sum_{i=1}^{N_w} (K_{ui} \cdot u_{yi} + K_{vi} \cdot v_{bi}) \tag{13}$$

where the nominal heading, θ_{nom}, is π radians, K_y is the feedback gain for perturbations in the cross-track direction, K_{ui} is the feedback gain for perturbations in along-track wind shear in the cross-track (y) direction at wind grid point i, and K_{vi} is the feedback gain for perturbations in the cross-track wind at wind grid point i. In principle, the number of wind grid points, N_w, is arbitrary, but one should not use more points than are necessary to accurately represent the winds because each additional point adds two new feedback gains to the implementation. In practice, between 10 and 15 points are adequate to represent the winds across the continental United States for optimal wind routing calculations.

The NOWR feedback law of eq. (13) is applied in a rotated and normalized coordinate frame so that the same gains may be applied to flights between any two points on the globe at any airspeed. The optimal wind routes computed with the NOWR algorithm are quite smooth and often exhibit interesting detail (fig. 5). Details of the NOWR algorithm, including the analytical expressions for the feedback gains, may be found in ref. [21].

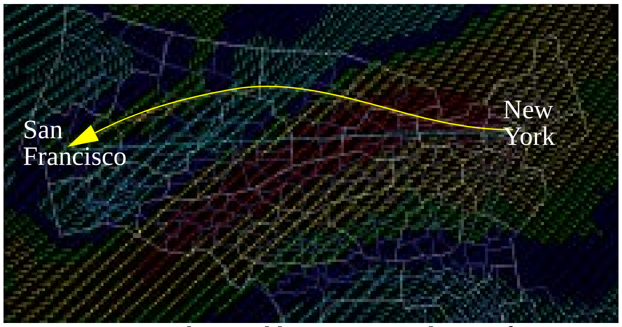

Figure 5. The neighboring optimal route from New York to San Francisco with RUC data from 14 Feb. 2000.

Using wind data from the RUC for six different weather days, computational statistics were computed for the NOWR algorithm. Neighboring optimal wind routes were computed for 42 different long-distance routes across the United States between some of the most common city pairs (e.g. New York to Los Angeles). The computation metric is the Floating Point Operation, or FLOP, from the MATLAB software package [20]. The FLOP is an approximate measure of the number of floating point operations used to execute a segment of code and is more useful for relative comparisons of different algorithms rather than for absolute computational effort measurements. For a set of typical cross-country routes, the mean computational effort for the NOWR algorithm is about 75,000 FLOPs, with a standard deviation of about 7,000 FLOPs.

For comparison, a study of the optimization performance and computational effort of discrete dynamic programming relative to NOWR was conducted [21] (fig.

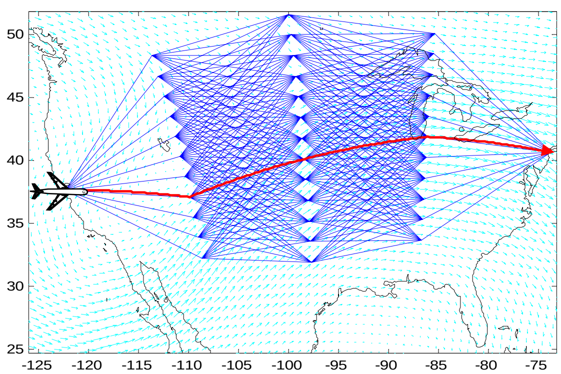

Figure 6. Discrete Dynamic Programming

6). The resolution of the dynamic programming grid was varied and the resulting minimum time trajectories and FLOPs were recorded. The grid resolution that produced the same optimization performance (on average) as the NOWR algorithm required more than 5 times the FLOPs of the NOWR algorithm. Qualitatively, these dynamic programming solutions were coarse (fig. 6), and the method did not easily lend itself to efficient perturbation techniques for conflict resolution.

The NOWR algorithm has been ported to the C computer language and installed within the Future ATM Concepts Evaluation Tool (FACET) for further simulation studies [22]. Tests on a 450 MHz Sun Ultra 60 workstation have shown that the NOWR algorithm can compute optimal wind routes in 40 milliseconds on average. As will be discussed later, this is fast enough to achieve real-time global optimization.

Conflict Detection: Deterministic Conflict Grid

The application of concepts from computer science to conflict detection has led to algorithms that have performance proportional to N [23]. Heuristic algorithms have also been developed and shown to be adequate for the purposes of air traffic control decision support tools [24]. Other algorithms have been developed based on the idea of *clustering*, or dividing up the conflict detection problem into a set of smaller ones according to which aircraft are in the same general neighborhood [12].

While these approaches have shown significant improvement over the N^2 performance of a brute force conflict search, they are still far too slow for use in a real-time optimization system. In addition, these algorithms were developed for the purpose of detecting conflicts among all aircraft in a set at the same time. In the case of the current optimization approach, each trajectory is sequentially optimized and deconflicted so that it is not necessary to recheck all aircraft for conflicts at each iteration.

For these reasons, a simple conflict detection algorithm called the Conflict Grid (CG) has been developed. The basic idea is to store optimal aircraft trajectories in a 3-dimensional grid space (two horizontal spatial dimensions and time) as they are computed by setting the values of the corresponding grid cells to one (binary "on", or "true") (fig. 7). The grid cell dimensions are set according to the allowable aircraft separation limits. If a grid cell is found to be already occupied, then it is immediately known that the current trajectory will be in conflict, and conflict resolution perturbations may be computed. Note that regions of bad weather and special use airspace may easily be incorporated into the conflict grid by setting the corresponding grid cell values for any of these areas of restricted airspace. The computational benefit of the CG approach derives from the elimination of pairwise distance computations.

The latitude and longitude grid dimensions are set large enough to include the airspace above the entire United

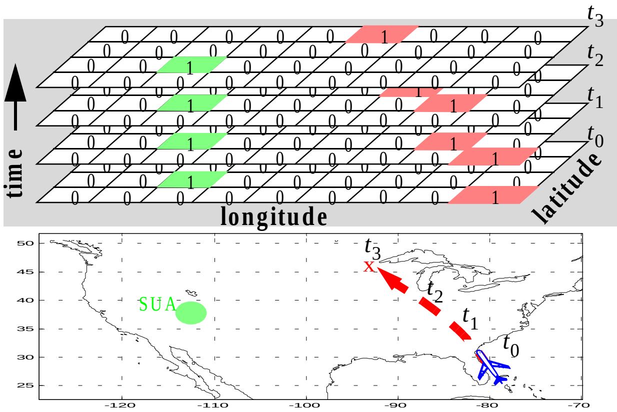

Figure 7. The Conflict Grid method.

States, plus an extra buffer to account for trajectories that might extend slightly beyond the U.S. airspace. The time dimension of the grid is set to span the appropriate amount of conflict look-ahead time. The maximum flight time for an aircraft across the continental United States is less than 7 hours, so this can be used as the maximum bound on the range of the CG time window. With these definitions, the CG may be mathematically represented by a 3-dimensional matrix as follows

$$CG = CG(i, j, k) \qquad \begin{cases} 0 \le i \le \left\lceil \dfrac{(\tau_{max} - \tau_{min})}{\Delta\tau} \right\rceil \\ 0 \le j \le \left\lceil \dfrac{(\lambda_{max} - \lambda_{min})}{\Delta\lambda} \right\rceil \\ 0 \le k \le \left\lceil \dfrac{(t_{max} - t_{min})}{\Delta t} \right\rceil \end{cases} \qquad (14)$$

$$\Delta\tau \equiv \frac{K_x \cdot \Delta x}{\cos(\max(\lambda))} \qquad \Delta t \equiv \frac{(t_{max} - t_{min})}{\left(\left\lceil \dfrac{(t_{max} - t_{min}) \cdot V_{max}}{\Delta x} \right\rceil + 1\right)}$$

$$\Delta\lambda \equiv K_x \cdot \Delta x$$

where τ and λ are longitude and latitude, t is time, Δx is the spatial grid dimension, K_x is a constant to convert from distance units to angular units, V_{max} is the maximum anticipated ground speed, and $\lceil a \rceil$ is the ceiling of a, defined as the smallest integer greater than a.

Since each grid cell only needs to store binary values, each cell only requires one bit of memory. The Continental United States extends approximately 2500 nautical miles from east to west, and 1500 nautical miles from north to south. For a 7-hour conflict look-ahead time, a time grid resolution of 30 seconds, and a grid spacing of 5 nautical miles, the memory required for the Conflict Grid is less than 16 Megabytes.

The procedure for conflict detection is now described. At the beginning of the conflict detection loop, the CG is cleared so that the value of each grid cell is set to zero. The next step is to store any weather constraints or special use airspace constraints in the CG by setting the corresponding constrained airspace grid cell values to 1. The next step is to loop through the list of all active aircraft to compute a predicted trajectory for each aircraft sequentially. The trajectory is generally computed and stored as a set of vectors of three spatial coordinates vs. time. These vectors are then interpolated to the discrete time values of the conflict grid. As the values are interpolated, the corresponding values of the conflict grid are checked. If the grid cell values are zero, then that means there are no prior aircraft occupying that cell, and that the airspace of that cell is not restricted by bad weather or other regulation. In that case, the value of that grid cell is set to 1 to signify that it is occupied by the current aircraft. If any of the trajectory points for the current aircraft are found to be in conflict at any of the grid cells, then the trajectory must be modified and checked again. The process continues until all aircraft in the active list have been given conflict free trajectories.

The CG algorithm as discussed to this point has some subtleties that must be addressed. If only the actual occupied grid cells are marked as such, it may happen that aircraft in neighboring grid cells are in conflict with one another. Another situation that may occur is that the discretized time steps of a trajectory may overstep a grid cell so that a real conflict is not identified. These subtleties are easily addressed by simple modifications of the CG method (e.g. grid cell buffering), but the basic algorithm remains the same [21].

Referring to eq. (6), the number of FLOPS required by the CG method is typically less than 100, which is two orders of magnitude less than the FLOPS required for the NOWR computations. Therefore, the CG method requires a negligible amount of computation compared to NOWR, and is essentially computationally free.

Stochastic Conflict Grid

A slight modification to the deterministic conflict grid can enhance the utility of the conflict detection algorithm. Instead of storing a binary value in each grid cell, one may compute and store the probability that *at least one* active constraint exists in that grid cell. An active constraint is any entity that requires exclusive use of the airspace, such as an aircraft, a weather storm cell, special-use airspace, or even an aircraft trailing wake vortex.

The cumulative probability that at least one constraint will be active, $\bar{P}_i$, is given by

$$\bar{P}_i = 1 - (1 - P_i)(1 - \bar{P}_{i-1}) \qquad (15)$$

where P_i is the probability that constraint i is active, and $\bar{P}_{i-1}$ is the cumulative probability that any constraint up to, and including, constraint $i-1$ will be active.

This generalization of the conflict grid concept may be achieved without adding a great deal of computational effort.

Conflict Resolution

Once a conflict has been detected via the CG method, possible conflict resolutions may be computed in parallel. In the stochastic conflict grid method, conflict resolution maneuvers would be initiated when conflict probabilities surpassed a threshold value. When a conflict is detected, it is not known with which aircraft there is a conflict, nor is the particular geometry known. The approach is to perturb the current aircraft trajectory in both of the possible horizontal directions until a resolution is found. If resolutions in these two directions are computed in parallel, one may keep track of the lowest cost maneuver that resolves the conflict so that it may be chosen as the solution. In this work, only heading changes in the horizontal plane have been examined, but it is also possible to look for speed perturbations or altitude perturbations. The primary use of horizontal conflict resolution for enroute aircraft is justified based upon fuel efficiency and passenger comfort and other considerations.

With the NOWR algorithm, there is a natural mechanism in place to generate smooth perturbation trajectories that are close approximations to optimal wind resolution trajectories. By introducing the concept of the *pseudo-shear*, a control is put in place to cause perturbations in the optimal heading command. The concept is to identify the nearest wind grid point to the conflict location and then to modify the wind shear at that location by adding a pseudo-shear (fig. 8). One may, in parallel, examine several possible pseudo-shear values, including both positive and negative values, until a conflict-free trajectory results. The one with the minimum cost is then chosen as the solution.

The nearest wind grid point is approximated by using the time at which the conflict is detected. The time is divided by the total trajectory time to determine the proportion of the trajectory that has already been travelled. Since the wind grid points are evenly spaced along the nominal trajectory, the closest point can be approximated using the trajectory time ratio. For N_w wind grid points, the

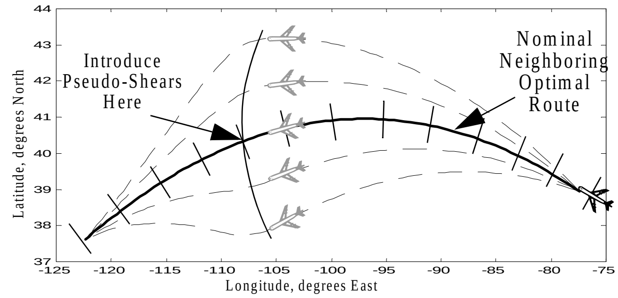

Figure 8. NOWR for conflict resolution

following expression gives the index of the grid point nearest to a conflict that is to occur at t_{conflict} along a trajectory that is predicted to take t_{total} time

$$i_w = N_w - \text{round}\left[(N_w - 1)\left(\frac{t_{\text{conflict}}}{t_{\text{total}}}\right)\right] \qquad (16)$$

The approximation in Eq. (16) has been demonstrated to work well. In practice, one may choose to limit the value of i_w so that wind shears are not introduced at either the first or the last wind control point where they have little effect.

The amount of pseudo-shear to use while iterating to find a conflict free trajectory is determined via parametric studies. If too small of a value is used, then excessive iterations might result. If too large of a value is chosen, then the first resolution maneuver that is found may be too large, and would be inefficient.

Simulation Results

The algorithms presented in this paper have been coded into software to run simulation studies. There are many free parameters in the algorithms presented, such as the dimensions of the conflict grid, the amount of pseudo-shear to use for perturbing trajectories, the amount of conflict look-ahead time to use, etc. These free parameters may be varied in simulation to determine their effect on the optimization performance, and to help choose appropriate values to use in practice. A brief glimpse of the results of two such studies is presented here. Additional results may be found in [21].

The first study examines the effect of reducing the required inter-aircraft separation on the effective airspace capacity. This was done by setting the CG dimensions to several different values and then simulating the complete optimization algorithm on real aircraft schedule data. The number of conflict iterations was recorded vs. the number of active aircraft and the airspace capacity metric, C_0, was computed for each CG dimension. A plot of the results shows how the maximum airspace capacity is expected to increase as the required separation decreases (fig. 9).

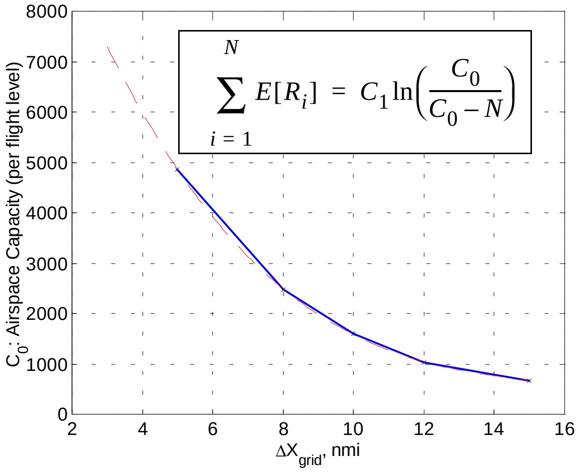

Figure 9. Airspace capacity vs. minimum aircraft separation

The next study examines the required computation rate to achieve real-time performance. By assuming that the optimal conflict free trajectories for all aircraft must be computed in the amount of time it takes a typical aircraft to transit a conflict grid cell, the following expression may be derived for the required computation rate of a neighboring optimal wind route

$$\rho_{NOWR} \approx \left[\left(\frac{\Delta X_{grid}}{V_{max}}\right)(1-\eta_c)\right]^{-1} \cdot C_1 \ln\left(\frac{C_0}{C_0 - N}\right) \quad (17)$$

where ρ_{NOWR} is the required NOWR computation rate (expressed as optimal routes per unit time), $(\Delta X_{grid}/V_{max})$ is the maximum allowable computation time, η_c is the ratio of communication time (time to communicate trajectories to the aircraft) to computation time, and the $\ln(\ )$ term is the expected number of computation iterations for N aircraft.

An interesting aspect of this problem is that as ΔX_{grid} increases, the amount of time available for computation increases. But when ΔX_{grid} increases, so does the number of conflicts because more airspace is required per aircraft. This suggests that some optimum grid spacing exists from a computational standpoint.

The results show that this is, indeed, the case. A plot of ρ_{NOWR} vs. ΔX_{grid} for three different values of η_c shows that there is a minimum required computation rate at about 12 nmi grid spacing (fig. 10). On the same plot, the computation rate achievable on an average Sun Ultra workstation (25 optimal routes per second) is shown to be

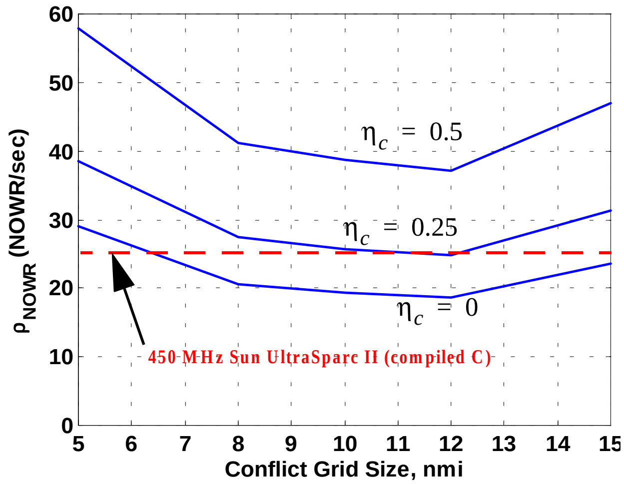

Figure 10. Recomputation rate vs. CG size

above the rate required for real-time operation for $\eta_c = 0.25$.

Conclusion

The main accomplishment of this work is that a complete set of conflict free optimal wind routes can now be computed for double the current-day single flight level air traffic density in less than one minute on an average (450MHz) workstation. This is an order-of-magnitude improvement over current state-of-the-art algorithms. This has been achieved through the development of an efficient optimal wind routing algorithm, and through improvements in the computational efficiency of strategic conflict detection and resolution.

References

[1] Erzberger, H., Pecsvaradi, T., “4-D Guidance System Design with Application to STOL Air Traffic Control,” 13th Joint Automatic Control Conference, Stanford, CA, Aug. 16-18, 1972.

[2] Erwin, R. L., “‘Strategic’ Time-Based ATC,” *Astronautics & Aeronautics*, V. 16, No. 11, Nov., 1978, pp. 56-61.

[3] Benoît, A., Swierstra, S., “A Cost-Efficient Control Procedure for the Benefit of All Airspace Users,” ACARD-CP-360, Guidance & Control Panel, 39th Symposium, Izmir, Turkey, 16-19 Oct., 1984.

[4] Erzberger, H., Tobias, L., “A Time-Based Concept for Terminal-Area Traffic Management,” AGARD-CP-410, Brussels, Belgium, 1986., pp. (52-1)-(52-14).

[5] Wilson, I. A. B., “PHARE: Definition and use of Tubes,” DOC 96-70-18, Eurocontrol, Rue de la Fusée 96, B-1130, Brussels, Belgium, 15 July, 1996 [Available at `http://www.eurocontrol.int/phare/`].

[6] Champougny, T., Duchene, A., et. al, “SOP: A Decision-aid Tool for Global Air Traffic Management System Optimisation,” *Air Traffic Control Quarterly*, Vol. 10(2), 2002, pp. 131-145.

[7] Erzberger, H., Lee, H., “Constrained Optimum Trajectories with Specified Range,” *AIAA Journal of Guidance and Control*, Vol. 3, Jan-Feb, 1980, pp. 78-85.

[8] Lidén, S., “Practical Considerations in Optimal Flight Management Computations,” *AIAA Journal of Guidance*, Vol. 9, No. 4, Jul-Aug, 1986, pp. 427-432.

[9] Durand, N., Alliot, J.-M., et al, “Optimal Resolution of En Route Conflicts,” *Air Traffic Control Quarterly*, Vol. 3, No. 3, 1995, pp. 139-161.

[10] Dean, G., Fron, X., et al, “ARC 2000: An Investigation into the Feasibility of Automatic Conflict Detection and Resolution,” *Air Traffic Control Quarterly*, Vol. 3, No. 4, 1995, pp. 229-259.

[11] Durand, N., Alliot, J.-M., et al, “Optimal Resolution of En Route Conflicts,” 1st USA/Europe Air Traffic Management Research and Development Seminar, Saclay, France, June 17-20, 1997.

[12] Chiang, Y.-J., Klosowski, J. T., et al, “Geometric Algorithms for Conflict Detection/Resolution in Air Traffic Management,” Proc. 36th IEEE Conference on Decision and Control, 1997.

[13] Oussedik, S., Delahaye, D., and Schoenauer, M., “Air Traffic Management by Stochastic Optimization,” 2nd USA/Europe Air Traffic Management Research and Development Seminar, Orlando, FL, Dec. 1-4, 1998.

[14] Ghosh, R., Tomlin, C., “Maneuver Design for Multiple Aircraft Conflict Resolution,” Proc. of the American Control Conference, Chicago, IL, June, 2000.

[15] Frazzoli, E., Mao, Z.-H., Oh, J.-H., Feron, E., “Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming,” *AIAA Journal of Guidance*, Control, and Dynamics, Vol. 24, No. 1, Jan.-Feb., 2001, pp. 79-86.

[16] Hu, X., Wu, S.-F., and Jiang, J., “GA Based On-Line Real-Time Optimization of Commercial Aircraft’s Flight Path for a Free Flight Strategy,” AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, August 6-9, 2001.

[17] Reif, J., Sharir, M., “Motion Planning in the Presence of Moving Obstacles,” *Journal of the Association for Computing Machinery*, Vol. 41, No. 4, July 1994, pp. 764-790.

[18] Bilimoria, K.D., Lee, H.Q., “Properties of Air Traffic Conflicts for Free and Structured Routing,” AIAA 2001-4051, AIAA Guidance, Navigation and Control Conference, 6-9 August, 2001, Montreal, Canada.

[19] Jardin, M. R., Bryson, A. E. Jr., “Neighboring Optimal Aircraft Guidance in Winds,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, Jul-Aug 2001, pp. 710-715.

[20] MATLAB Version 5.3, The Mathworks, Inc., Jan. 21, 1999.

[21] Jardin, M. R., “Toward Real-Time En Route Air Traffic Control Optimization,” Ph.D. Dissertation, Stanford University Dept. of Aeronautics & Astronautics, April 2003.

[22] Bilimoria, K., Sridhar, B., et al., “FACET: Future ATM Concepts Evaluation Tool,” *Air Traffic Control Quarterly*, Vol. 9, No. 1, 2001.

[23] Sridhar, B., Chatterji, G. B., “Computationally Efficient Conflict Detection Methods for Air Traffic Management,” American Control Conference, Albuquerque, NM, June 4-6, 1997.

[24] Isaacson, D. R., and Erzberger, H., “Design of a Conflict Detection Algorithm for the Center/TRACON Automation System,” 16th Digital Avionics Systems Conference, Irvine, CA, Oct. 26-30, 1997.

Key Words

aircraft trajectory optimization, optimal wind routing, air traffic control, neighboring optimal control, real-time optimization, air traffic conflict resolution.

Biography

Dr. Jardin has worked at the NASA Ames Research Center in Mountain View, California in air traffic control optimization and automation since 1993. While at NASA, he earned his Ph.D. from the Dept. of Aeronautics and Astronautics at Stanford University in April 2003. Prior to joining NASA, Dr. Jardin earned his Bachelor’s (1990) and Master’s (1992) degrees in Aeronautics and Astronautics from the University of Washington in Seattle, Washington. His research interests are in trajectory optimization and the application thereof.